aws

Guida per l'utente

Amazon CloudWatch

Amazon CloudWatch: Guida per l'utente

Table of Contents

Che cos'è Amazon CloudWatch?

Amazon CloudWatch monitora le tue risorse Amazon Web Services (AWS) e le applicazioni su cui esegui AWS in tempo reale. Puoi utilizzarlo CloudWatch per raccogliere e tracciare i parametri, che sono variabili che puoi misurare per le tue risorse e applicazioni.

La CloudWatch home page mostra automaticamente le metriche relative a ogni AWS servizio che utilizzi. Puoi inoltre creare pannelli di controllo personalizzati per visualizzare i parametri relativi alle applicazioni personalizzate e visualizzare raccolte personalizzate dei parametri scelti.

Puoi creare allarmi con parametri di controllo e inviare notifiche o apportare automaticamente modifiche alle risorse che stai monitorando quando viene superata una soglia. Ad esempio, puoi monitorare l'utilizzo della CPU, le letture e le scritture sul disco delle istanze Amazon EC2 e, quindi, utilizzare tali dati per determinare se debbano essere avviate ulteriori istanze per gestire il carico incrementato. Puoi inoltre utilizzare questi dati per fermare le istanze poco utilizzate per risparmiare denaro.

Con CloudWatch, ottieni una visibilità a livello di sistema sull'utilizzo delle risorse, sulle prestazioni delle applicazioni e sullo stato operativo.

Accedendo CloudWatch

È possibile accedere CloudWatch utilizzando uno dei seguenti metodi:

- CloudWatch Console Amazon: https://console.aws.amazon.com/cloudwatch/
- AWS CLI: per ulteriori informazioni, consulta Configurazione con the AWS Command Line Interface nella Guida per l'AWS Command Line Interface utente.
- CloudWatch API: per ulteriori informazioni, consulta Amazon CloudWatch API Reference.
- AWS SDK: per ulteriori informazioni, consulta Tools for Amazon Web Services.

Servizi correlati AWS

I seguenti servizi vengono utilizzati insieme ad Amazon CloudWatch:

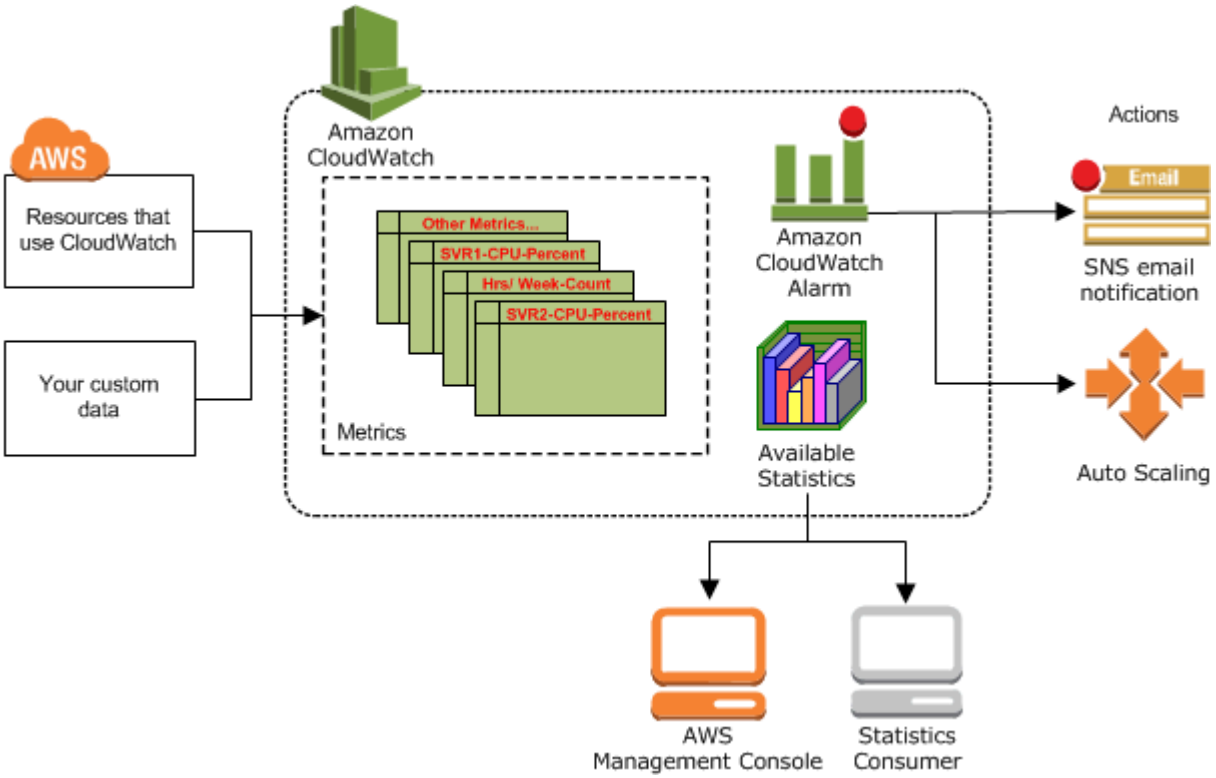
- Amazon Simple Notification Service (Amazon SNS) coordina e gestisce la consegna o l'invio di messaggi agli endpoint o ai clienti abbonati. Utilizzi Amazon SNS CloudWatch per inviare messaggi

quando viene raggiunta una soglia di allarme. Per ulteriori informazioni, consulta Impostazione delle notifiche Amazon SNS.

- Amazon EC2 Auto Scaling permette di avviare o terminare automaticamente istanze Amazon EC2 in base a policy definite dall'utente, controlli dello stato di integrità e pianificazioni. Puoi usare un CloudWatch allarme con Amazon EC2 Auto Scaling per ridimensionare le tue istanze EC2 in base alla domanda. Per ulteriori informazioni, consulta Dimensionamento dinamico nella Guida per l'utente di Amazon EC2 Auto Scaling.
- AWS CloudTrailti consente di monitorare le chiamate effettuate all' CloudWatch API Amazon per il tuo account, incluse le chiamate effettuate da AWS Management Console AWS CLI, e da altri servizi. Quando CloudTrail la registrazione è attivata, CloudWatch scrive i file di registro nel bucket Amazon S3 specificato durante la configurazione. CloudTrail Per ulteriori informazioni, consulta Registrazione delle chiamate CloudWatch API Amazon con AWS CloudTrail.
- AWS Identity and Access Management (IAM) è un servizio web che ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse per i tuoi utenti. Utilizza IAM per controllare chi può utilizzare le tue risorse AWS (autenticazione), quali risorse e in che modo (autorizzazione). Per ulteriori informazioni, consulta Gestione delle identità e degli accessi per Amazon CloudWatch.

Come CloudWatch funziona Amazon

Amazon CloudWatch è fondamentalmente un archivio di metriche. Un AWS servizio, come Amazon EC2, inserisce i parametri nel repository e tu recuperi le statistiche in base a tali parametri. Puoi anche recuperare le statistiche basate su parametri personalizzati, se questi sono stati messi in archivio.



Puoi utilizzare le metriche per calcolare le statistiche e quindi presentare i dati graficamente nella console. CloudWatch Per ulteriori informazioni sulle altre AWS risorse che generano e inviano metriche, consulta. CloudWatch AWS servizi che pubblicano CloudWatch metriche

Puoi configurare le operazioni degli allarmi in modo da arrestare, avviare o terminare un'istanza Amazon EC2 quando vengono soddisfatti determinati criteri. Inoltre, puoi creare allarmi che avviano automaticamente operazioni Amazon EC2 Auto Scaling e Amazon Simple Notification Service (Amazon SNS). Per ulteriori informazioni sulla creazione di CloudWatch allarmi, consulta. Allarmi

AWS Le risorse di cloud computing sono ospitate in strutture di data center ad alta disponibilità. Per fornire più scalabilità e affidabilità, ogni struttura di data center si trova in una determinata zona geografica, nota come una regione. Ogni regione è stato progettata per essere completamente isolata dalle altre, per avere il maggiore isolamento dell'errore e la maggiore stabilità possibili. Le metriche vengono archiviate separatamente nelle regioni, ma è possibile utilizzare la funzionalità CloudWatch interregionale per aggregare le statistiche di diverse regioni. Per ulteriori informazioni, consulta Console per più account e più regioni CloudWatch e Regioni ed endpoint nella Riferimenti generali di Amazon Web Services.

CloudWatch Concetti di Amazon

La terminologia e i concetti seguenti sono fondamentali per la comprensione e l'uso di Amazon CloudWatch:

- Spazi dei nomi
- Metriche
- Dimensioni
- Risoluzione
- Statistiche
- Percentili
- Allarmi

Per informazioni sulle quote di servizio per CloudWatch metriche, allarmi, richieste API e notifiche e-mail di allarme, consulta le quote di servizio. CloudWatch

Spazi dei nomi

Un namespace è un contenitore per le metriche. CloudWatch I parametri in diversi spazi dei nomi sono isolati tra loro, in modo che i parametri provenienti dalle diverse applicazioni non vengano erroneamente aggregati nelle stesse statistiche.

Non esistono spazi dei nomi predefiniti. È necessario specificare uno spazio dei nomi per ogni punto dati in cui si pubblica. CloudWatch Puoi specificare un nome per lo spazio dei dati al momento della creazione di un parametro. Questi nomi devono contenere un massimo di 255 caratteri ASCII validi. I caratteri possibili sono: caratteri alfanumerici (0-9a-zA-Z), punto (.), trattino (-), trattino basso (_), barra (/), cancelletto (#), due punti (:) e lo spazio. Uno spazio dei nomi deve contenere almeno un carattere diverso dallo spazio.

AWS I namespace utilizzano in genere la seguente convenzione di denominazione:. `AWS/service` Ad esempio, Amazon EC2 usa lo spazio dei nomi `AWS/EC2`. Per l'elenco dei namespace, vedere AWS . AWS servizi che pubblicano CloudWatch metriche

Metriche

Le metriche sono il concetto fondamentale in. CloudWatch Una metrica rappresenta un insieme di punti dati ordinati nel tempo su cui vengono pubblicati. CloudWatch Pensa a un parametro come a

una variabile da monitorare e ai punti di dati come i valori di questa variabile nel tempo. Ad esempio, l'utilizzo della CPU di una determinata istanza EC2 è un parametro fornito da Amazon EC2. I punti di dati possono provenire da qualsiasi applicazione o attività di business da cui raccogliere i dati.

Per impostazione predefinita, molti AWS servizi forniscono parametri gratuiti per le risorse (come istanze Amazon EC2, volumi Amazon EBS e istanze database Amazon RDS). A pagamento, puoi inoltre abilitare il monitoraggio dettagliato di alcune risorse, ad esempio le istanze Amazon EC2 o pubblicare i tuoi parametri relativi alle applicazioni. Per i parametri personalizzati, puoi aggiungere i punti di dati in qualsiasi ordine e tasso scelti. Puoi recuperare le statistiche su quei punti di dati come set ordinato di dati di serie temporali.

I parametri esistono solo nella regione in cui sono stati creati. Non possono essere eliminati, ma scadono automaticamente dopo 15 mesi se non vengono pubblicati altri nuovi dati. I punti di dati precedenti a 15 mesi scadono su base sequenziale; nel momento in cui sono disponibili nuovi punti di dati, i dati più vecchi di 15 mesi vengono messi da pare.

I parametri sono definiti in modo univoco da un nome, uno spazio dei nomi e da nessuna o più dimensioni. A ogni punto di dati in un parametro è associato un timestamp e (facoltativamente) un'unità di misura. Puoi recuperare statistiche da qualsiasi metrica. CloudWatch

Per ulteriori informazioni, consulta Visualizzazione di parametri disponibili e Pubblicare i parametri personalizzati di .

Timestamp

Ogni punto di dati del parametro deve essere associato a un timestamp. Il timestamp può andare indietro fino a due settimane e avanti fino a due ore. Se non fornisci un timestamp, CloudWatch crea automaticamente un timestamp basato sull'ora in cui il punto dati è stato ricevuto.

I timestamp sono oggetti `dateTime`, con la data completa più ore, minuti e secondi (ad esempio 2016-10-31T23:59:59 Z). Per ulteriori informazioni, consulta l'articolo relativo a dateTime. Anche se non è necessario, è consigliabile usare il formato UTC Quando recuperi le statistiche da CloudWatch, tutti gli orari sono in UTC.

CloudWatch gli allarmi controllano le metriche in base all'ora corrente in UTC. Le metriche personalizzate inviate CloudWatch con timestamp diversi dall'ora UTC corrente possono far sì che gli allarmi mostrino lo stato Dati insufficienti o generino allarmi ritardati.

Conservazione dei parametri

CloudWatch conserva i dati metrici come segue:

- I punti di dati con un periodo di meno di 60 secondi sono disponibili per 3 ore. Questi punti di dati sono parametri personalizzati ad alta risoluzione.
- I punti di dati con un periodo di 60 secondi (1 minuto) sono disponibili per 15 giorni.
- I punti di dati con un periodo di 300 secondi (5 minuti) sono disponibili per 63 giorni.
- I punti di dati con un periodo di 3.600 secondi (1 ora) sono disponibili per 455 giorni (15 mesi).

I punti di dati che vengono pubblicati inizialmente con un periodo più breve vengono aggregati per uno storage a lungo termine. Ad esempio, se raccogli dati usando un periodo di 1 minuto, i dati rimangono disponibili per 15 giorni con una risoluzione di 1 minuto. Dopo 15 giorni questi dati sono ancora disponibili, ma vengono aggregati e possono essere recuperati solo con una risoluzione di 5 minuti. Dopo 63 giorni, i dati vengono ulteriormente aggregato e sono disponibili con una risoluzione di 1 ora.

> **Note**
>
> I parametri che non hanno ricevuto nuovi punti di dati nelle ultime due settimane non vengono visualizzati nella console. Inoltre, non vengono visualizzati quando digiti il nome del parametro o i nomi delle dimensioni nella casella di ricerca della scheda Tutti i parametri della console e non vengono restituiti nei risultati del comando list-metrics . Il modo migliore per recuperare queste metriche è utilizzare i comandi get-metric-datao get-metric-statisticsin. AWS CLI

Dimensioni

Una dimensione è una coppia nome-valore che fa parte dell'identità di un parametro. Puoi assegnare a un parametro fino a 30 dimensioni.

Ogni parametro ha caratteristiche specifiche che lo descrivono ed puoi considerare le dimensioni come categorie di tali caratteristiche. Le dimensioni ti consentono di creare una struttura per il piano delle statistiche. Poiché le dimensioni fanno parte dell'identificatore univoco di un parametro, ogni volta che viene aggiunta una coppia nome/valore a uno dei parametri, crei una nuova variante di detto parametro.

AWS servizi che inviano dati per CloudWatch allegare dimensioni a ciascuna metrica. È possibile utilizzare le dimensioni per filtrare i risultati CloudWatch restituiti. Ad esempio, puoi ottenere le

statistiche per una determinata istanza EC2 specificando la dimensione `InstanceId` al momento della ricerca dei parametri.

Infatti, le metriche prodotte da determinati AWS servizi, come Amazon EC2 CloudWatch , possono aggregare i dati tra diverse dimensioni. Ad esempio, se cerchi metriche nello spazio dei `AWS/EC2` nomi ma non specifichi alcuna dimensione, CloudWatch aggrega tutti i dati per la metrica specificata per creare la statistica richiesta. CloudWatch non si aggrega tra le dimensioni per le metriche personalizzate.

Combinazioni delle dimensioni

CloudWatch considera ogni combinazione univoca di dimensioni come una metrica separata, anche se le metriche hanno lo stesso nome di metrica. Puoi recuperare le statistiche solo usando combinazioni di dimensioni che sono state specificamente pubblicate. Quando recuperi le statistiche, specifica gli stessi valori per lo spazio dei nomi, il nome parametro e i parametri della dimensione che sono stati usati quando sono stati creati i parametri. Puoi anche specificare l'ora di inizio e di fine da utilizzare CloudWatch per l'aggregazione.

Ad esempio, supponete di pubblicare quattro metriche distinte denominate ServerStats nello spazio dei DataCenterMetric nomi con le seguenti proprietà:

```
Dimensions: Server=Prod, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:30:00Z, Value: 105
Dimensions: Server=Beta, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:31:00Z, Value: 115
Dimensions: Server=Prod, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:32:00Z, Value: 95
Dimensions: Server=Beta, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:33:00Z, Value: 97
```

Se pubblichi solo i quattro parametri, puoi recuperare le statistiche per queste combinazioni di dimensioni:

- `Server=Prod,Domain=Frankfurt`
- `Server=Prod,Domain=Rio`
- `Server=Beta,Domain=Frankfurt`
- `Server=Beta,Domain=Rio`

Non puoi recuperare le statistiche per le seguenti dimensioni o se non hai specificato alcuna dimensione: (L'eccezione consiste nell'utilizzare la funzione RICERCA di matematica dei parametri, che consente di recuperare le statistiche per più parametri. Per ulteriori informazioni, consulta Utilizzo delle espressioni di ricerca nei grafici.)

- `Server=Prod`
- `Server=Beta`
- `Domain=Frankfurt`
- `Domain=Rio`

Risoluzione

Ogni parametro appartiene a una delle seguenti categorie:

- Risoluzione standard, con dati aventi una granularità di un minuto
- Alta risoluzione, con dati aventi una granularità di un secondo

Per impostazione predefinita, le metriche prodotte dai AWS servizi hanno una risoluzione standard. Quando pubblichi un parametro personalizzato, puoi definirlo sia come risoluzione standard che come alta risoluzione. Quando pubblichi una metrica ad alta risoluzione, la CloudWatch archivia con una risoluzione di 1 secondo e puoi leggerla e recuperarla con un periodo di 1 secondo, 5 secondi, 10 secondi, 30 secondi o qualsiasi multiplo di 60 secondi.

I parametri ad alta risoluzione ti offrono un'analisi più immediata sull'attività inferiore al minuto dell'applicazione. Tieni presente che ogni chiamata `PutMetricData` per un parametro personalizzato viene addebitata, quindi frequenti chiamate a `PutMetricData` su un parametro ad alta risoluzione potrebbero portare a costi più elevati. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Se imposti un allarme su un parametro ad alta risoluzione, puoi specificare un allarme ad alta risoluzione con un periodo di 10 secondi o 30 secondi, oppure puoi impostare un allarme regolare con un periodo di qualsiasi multiplo di più di 60 secondi. Viene addebitato un costo maggiore per gli allarmi ad alta risoluzione con un periodo di 10 o 30 secondi.

Statistiche

Le statistiche sono aggregazioni di dati metrici su periodi di tempo specifici. CloudWatch fornisce statistiche basate sui punti dati metrici forniti dai dati personalizzati o forniti da altri AWS servizi a. CloudWatch Le aggregazioni vengono effettuate usando lo spazio dei nomi, il nome parametro, le dimensioni e l'unità di misura del punto dati, entro un periodo di tempo specificato.

Per le definizioni dettagliate delle statistiche supportate da CloudWatch, vedereCloudWatch definizioni statistiche.

Unità

Ogni statistica è un'unità di misura. Le unità di esempio includono `Bytes`, `Seconds`, `Count` e `Percent`. Per l'elenco completo delle unità CloudWatch supportate, consulta il tipo di MetricDatumdati nell'Amazon CloudWatch API Reference.

Puoi specificare una unità al momento della creazione di un parametro personalizzato. Se non specifichi un'unità, CloudWatch viene utilizzata `None` come unità. Le unità consentono di fornire significati concettuali ai tuoi dati. Sebbene non CloudWatch attribuisca alcun significato a un'unità internamente, altre applicazioni possono ricavare informazioni semantiche basate sull'unità.

I punti di dati del parametro che specificano un'unità di misura sono aggregati separatamente. Quando si ottengono statistiche senza specificare un'unità, CloudWatch aggrega tutti i punti dati della stessa unità. Altrimenti, se disponi di due identici parametri con unità diverse, vengono restituiti due flussi di dati separati, uno per ogni unità.

Periodi

Un periodo è il periodo di tempo associato a una CloudWatch statistica Amazon specifica. Ogni statistica rappresenta un'aggregazione di dati di parametri raccolti per un periodo di tempo specificato. I periodi sono definiti in numero di secondi e i valori validi per il periodo sono 1, 5, 10, 30 o qualsiasi multiplo di 60. Ad esempio, per specificare un periodo di 6 minuti, usa come valore del periodo 360. Puoi modificare il modo in cui i dati vengono aggregati variando la durata del periodo. Il valore predefinito di un periodo è 60 secondi. Un periodo può durare solo un secondo e deve essere un multiplo di 60 se è superiore al valore predefinito di 60 secondi.

Solo i parametri personalizzati che definisci con una risoluzione di storage di 1 secondo supportano periodi inferiori al minuto. Anche se è sempre disponibile nella console la possibilità di impostare un

periodo inferiore 60, devi selezionare un periodo che si allinea con la modalità di archiviazione del parametro. Per ulteriori informazioni sui parametri che supportano periodi inferiori al minuto, consulta l'articolo sui Parametri ad alta risoluzione.

Quando recuperi le statistiche, puoi specificare un periodo, il momento di inizio e quello di fine. Questi parametri determinano la durata generale del tempo associato alle statistiche. Per i valori predefiniti per il momento di inizio e fine considera l'ultimo valore dell'ora delle statistiche. I valori specificati per l'ora di inizio e l'ora di fine determinano il numero di periodi CloudWatch restituiti. Ad esempio, il recupero di statistiche utilizzando i valori predefiniti per il periodo, i momenti di inizio e fine restituisce un set aggregato di statistiche per ogni minuto dell'ora precedente. Se preferisci statistiche aggregate in blocchi di dieci minuti, specifica un periodo di 600. Per le statistiche aggregate per l'intera ora, specifica un periodo di 3.600.

Quando le statistiche sono aggregate in un periodo di tempo, hanno un timestamp con l'orario che corrisponde all'inizio del periodo. Ad esempio, i dati aggregati dalle 19:00 alle 20:00 hanno un timestamp con le 19:00. Inoltre, i dati aggregati tra le 19:00 e le 20:00 iniziano a essere visibili alle 19:00, quindi i valori di tali dati aggregati possono cambiare man mano che vengono CloudWatch raccolti più campioni durante il periodo.

I periodi sono importanti anche per gli allarmi. CloudWatch Quando crei un allarme per monitorare una metrica specifica, chiedi CloudWatch di confrontare quella metrica con il valore di soglia che hai specificato. Hai un ampio controllo su come effettuare questo CloudWatch confronto. Non solo puoi specificare il periodo durante il quale viene effettuato il confronto, ma puoi anche specificare il numero di periodi di valutazione che vengono utilizzati per arrivare a una conclusione. Ad esempio, se si specificano tre periodi di valutazione, CloudWatch confronta una finestra di tre punti dati. CloudWatch ti avvisa solo se il punto dati più vecchio viene violato e gli altri sono violati o mancanti.

Aggregazione

Amazon CloudWatch aggrega le statistiche in base alla durata del periodo specificata al momento del recupero delle statistiche. Puoi pubblicare tutti i punti dati che desideri con timestamp uguali o simili. CloudWatch li aggrega in base alla durata del periodo specificata. CloudWatch non aggrega automaticamente i dati tra le regioni, ma puoi utilizzare la matematica metrica per aggregare le metriche di diverse regioni.

Puoi pubblicare punti dati per una metrica che condividono non solo lo stesso timestamp, ma anche lo stesso namespace e le stesse dimensioni. CloudWatch restituisce statistiche aggregate per tali punti dati. Puoi anche pubblicare più punti di dati per lo stesso parametro o per parametri diversi, con qualsiasi timestamp

Per set di dati di grandi dimensioni, puoi inserire un set di dati aggregati chiamato set statistico. Con i set di statistiche, si forniscono CloudWatch i valori Min, Max, Sum e SampleCount per un certo numero di punti dati. Questo viene comunemente utilizzato quando è necessario raccogliere i dati più volte in un minuto. Ad esempio, supponi di disporre di un parametro per una latenza di richieste di una pagina Web. Non ha senso pubblicare dati con ogni occorrenza della pagina Web. Ti consigliamo di raccogliere la latenza di tutti gli accessi a quella pagina web, aggregarli una volta al minuto e inviare la statistica impostata su. CloudWatch

Amazon CloudWatch non differenzia la fonte di una metrica. Se pubblichi una metrica con lo stesso spazio dei nomi e le stesse dimensioni da fonti diverse, la CloudWatch considera come un'unica metrica. Questo può essere utile per i parametri di servizi in un sistema distribuito e scalato. Ad esempio, tutti gli host di un'applicazione server Web potrebbero pubblicare metriche identiche che rappresentano la latenza delle richieste che stanno elaborando. CloudWatchle tratta come un'unica metrica, consentendoti di ottenere le statistiche relative al minimo, al massimo, alla media e alla somma di tutte le richieste nell'applicazione.

Percentili

Un percentile indica lo stato relativo di un valore in un set di dati. Ad esempio, 95° percentile vuol dire che il 95% dei dati è inferiore a questo valore e il 5% dei dati è superiore a questo valore. I percentili aiutano a comprendere meglio la distribuzione dei dati del parametro.

I percentili sono spesso utilizzati per isolare le anomalie. In una normale distribuzione, il 95% dei dati è tra due deviazioni standard dalla media e il 99,7% dei dati è all'interno di tre deviazioni standard dal significato. Qualsiasi dato che non rientra nelle tre deviazioni standard è spesso considerato come un'anomalia perché differisce in modo notevole dal valore medio. Ad esempio, supponiamo che desideri monitorare l'utilizzo della CPU di istanze EC2 per assicurare che i tuoi clienti vivano un'esperienza positiva. Se monitori la media, questa può nascondere delle anomalie. Se monitori il massimo, una singola anomalia può stravolgere i risultati. Utilizzando i percentili, puoi monitorare il 95° percentile di utilizzo della CPU per verificare la presenza di istanze con un carico insolitamente elevato.

Alcune CloudWatch metriche supportano i percentili come statistica. Per queste metriche, puoi monitorare il sistema e le applicazioni utilizzando i percentili come faresti con le altre CloudWatch statistiche (media, minimo, massimo e somma). Ad esempio, quando crei un allarme, puoi usare i percentili come la funzione statistica. Puoi specificare il percentile, utilizzando fino a dieci decimali (ad esempio, p95.0123456789).

Sono disponibili statistiche basate su percentile per i parametri personalizzati, purché pubblichi i punti dati non riepilogati e non elaborati per il parametro personalizzato. Le statistiche di percentile non sono disponibili per i parametri quando uno qualsiasi dei valori dei parametri è un numero negativo.

CloudWatch necessita di punti dati grezzi per calcolare i percentili. Se pubblichi dati utilizzando un set di statistiche, invece, puoi solamente recuperare le statistiche dei percentili per questi dati se risulta vera una delle seguenti condizioni:

- Il SampleCount valore del set di statistiche è 1 e Min, Max e Sum sono tutti uguali.
- Min e Max sono uguali e Sum è uguale a Min moltiplicato per. SampleCount

I seguenti AWS servizi includono metriche che supportano le statistiche percentili.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Sistema di bilanciamento del carico elastico
- Kinesis
- Amazon RDS

CloudWatch supporta anche la media ridotta e altre statistiche sulle prestazioni, che possono avere un uso simile ai percentili. Per ulteriori informazioni, consulta CloudWatch definizioni statistiche.

Allarmi

Puoi usare un allarme per iniziare automaticamente le operazioni per conto tuo. Un allarme controlla un singolo parametro in un periodo di tempo specificato ed esegue una o più operazioni specificate in base al valore del parametro relativo a una determinata soglia durante un periodo di tempo. L'operazione corrisponde all'invio di una notifica a un argomento di Amazon SNS o a una policy di Auto Scaling. Puoi anche aggiungere allarmi ai pannelli di controllo.

Gli allarmi richiamano azioni solo per cambiamenti di stato sostenuti. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare. Lo stato deve essere cambiato e restare costante per un numero specificato di periodi.

Quando crei un allarme, seleziona un periodo di monitoraggio degli allarmi maggiore o uguale alla risoluzione del parametro. Ad esempio, il monitoraggio base per Amazon EC2 fornisce parametri

per le istanze ogni 5 minuti. Quando imposti un allarme su uno dei parametri del monitoraggio base, seleziona un periodo di almeno 300 secondi (5 minuti). Il monitoraggio dettagliato per Amazon EC2 fornisce parametri per le istanze con una risoluzione di 1 minuto. Quando imposti un allarme su un parametro del monitoraggio dettagliato, seleziona un periodo di almeno 60 secondi (1 minuto).

Se imposti un allarme su un parametro ad alta risoluzione, puoi specificare un allarme ad alta risoluzione con un periodo di 10 secondi o 30 secondi, oppure puoi impostare un allarme regolare con un periodo di qualsiasi multiplo di più di 60 secondi. Per gli allarmi ad alta risoluzione il costo è più elevato. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Pubblicare i parametri personalizzati di .

Per ulteriori informazioni, consulta Utilizzo degli CloudWatch allarmi Amazon e Creazione di un allarme a partire da un parametro in un grafico.

Fatturazione e costi

Per informazioni complete sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Per informazioni su come analizzare la fattura e possibilmente ottimizzare e ridurre i costi, consulta CloudWatch fatturazione e costi.

CloudWatch Risorse Amazon

Le seguenti risorse correlate possono rivelarsi utili durante l'utilizzo di questo servizio.

Risorsa	Descrizione
CloudWatchDomande frequenti su Amazon	La sezione Domande frequenti include le domande principali che gli sviluppatori pongo in merito a questo prodotto.
AWS Centro per sviluppatori	Un punto di partenza centrale per trovare documenta zione, esempi di codice, note di rilascio e altre informazi oni con cui aiutarti a creare applicazioni innovative AWS.
AWS Management Console	La console consente di eseguire la maggior parte delle funzioni di Amazon CloudWatch e di varie altre AWS offerte senza programmazione.

Risorsa	Descrizione
Forum di CloudWatch discussione Amazon	Forum basato sulla community per sviluppatori per discutere di questioni tecniche relative ad Amazon. CloudWatch
AWS Support	L'hub per la creazione e la gestione dei casi. AWS Support Include anche collegamenti ad altre risorse utili, come forum, domande frequenti tecniche, stato di integrità del servizio e AWS Trusted Advisor.
Informazioni CloudWatch sui prodotti Amazon	La pagina web principale per informazioni su Amazon CloudWatch.
Contattaci	Un punto di contatto centrale per domande relative a AWS fatturazione, account, eventi, abusi, ecc.

Configurazione delle impostazioni

Per utilizzare Amazon CloudWatch è necessario un AWS account. Il tuo AWS account ti consente di utilizzare servizi (ad esempio Amazon EC2) per generare metriche che puoi visualizzare nella CloudWatch console, un' point-and-clickinterfaccia basata sul Web. Inoltre, è possibile installare e configurare l'interfaccia a riga di AWS comando (CLI).

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegnate l'accesso amministrativo a un utente e utilizzate solo l'utente root per eseguire attività che richiedono l'accesso da parte dell'utente root.

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con le impostazioni predefinite IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso ad altri utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Per istruzioni, consulta Creare un set di autorizzazioni nella Guida per l'utente.AWS IAM Identity Center

2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente.AWS IAM Identity Center

Accedi alla CloudWatch console Amazon

Per accedere alla CloudWatch console Amazon

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Se necessario, usa la barra di navigazione per cambiare la regione con la regione in cui hai AWS le tue risorse.
3. Anche se è la prima volta che utilizzi la CloudWatch console, Your Metrics potrebbe già riportare le metriche, perché hai utilizzato un AWS prodotto che invia automaticamente le metriche ad Amazon CloudWatch gratuitamente. Altri servizi richiedono l'abilitazione dei parametri.

   Se non disponi di alcun allarme, la sezione Your Alarms (I tuoi allarmi) presenterà un pulsante Create Alarm (Crea allarme).

Configura il AWS CLI

Puoi utilizzare AWS CLI o l'Amazon CloudWatch CLI per eseguire CloudWatch i comandi. Tieni presente che AWS CLI sostituisce la CloudWatch CLI; includiamo CloudWatch nuove funzionalità solo in. AWS CLI

Per informazioni su come installare e configurare AWS CLI, vedere Getting Set Up with the AWS Command Line Interface nella Guida per l'AWS Command Line Interface utente.

Per informazioni su come installare e configurare Amazon CloudWatch CLI, consulta Configurare l'interfaccia a riga di comando nell'Amazon CLI CloudWatch Reference.

Guida introduttiva ad Amazon CloudWatch

Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

Viene visualizzata la home page CloudWatch panoramica.

La panoramica contiene i seguenti elementi, aggiornati automaticamente.

- Allarmi per AWS servizio mostra un elenco dei AWS servizi che usi nel tuo account, insieme allo stato degli allarmi in tali servizi. Inoltre, vengono visualizzati due o quattro allarmi nel tuo account. Il numero dipende dal numero di AWS servizi che utilizzi. Gli allarmi mostrati sono quelli nello stato ALARM o quelli il cui stato è stato modificato di recente.

  Queste aree superiori consentono di valutare rapidamente lo stato dei AWS servizi, visualizzando gli stati di allarme di ogni servizio e gli allarmi che hanno cambiato stato più di recente. Questo consente di monitorare e diagnosticare rapidamente i problemi.
- Sotto queste aree è disponibile il pannello di controllo predefinito, se esistente. La dashboard predefinita è una dashboard personalizzata che hai creato e denominata CloudWatch-Default. Si tratta di un modo pratico per aggiungere metriche relative ai servizi o alle applicazioni personalizzati alla pagina di panoramica o per presentare ulteriori metriche chiave relative ai AWS servizi che si desidera monitorare di più.

> **Note**
>
> Le dashboard automatiche sulla CloudWatch home page visualizzano solo le informazioni dell'account corrente, anche se l'account è un account di monitoraggio configurato per CloudWatch l'osservabilità tra account. Per ulteriori informazioni sulla creazione di pannelli di controllo tra account, consulta la sezione CloudWatch dashboard di osservabilità tra account.

Da questa panoramica, puoi visualizzare un pannello di controllo delle metriche relative a più AWS servizi oppure concentrare la visualizzazione su un gruppo di risorse o un servizio specifico. AWS Questo consente di limitare la visualizzazione a un sottoinsieme di risorse di interesse. Per ulteriori informazioni, consultare le sezioni indicate di seguito.

Visualizza la dashboard automatica predefinita per un singolo servizio

Per visualizzare la dashboard automatica preconfigurata per un singolo servizio

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

   Viene visualizzata la home page.
2. Nel riquadro di navigazione a sinistra, scegli Dashboard.
3. Scegli la scheda Dashboard automatiche, quindi scegli il servizio che desideri visualizzare.
4. Per passare alla visualizzazione degli allarmi per questo servizio, seleziona la casella di controllo In allarme, Dati insufficienti o OK nella parte superiore della schermata in cui è attualmente visualizzato il nome del servizio.
5. Durante la visualizzazione dei parametri, puoi concentrarti su un particolare parametro in diversi modi:

   a. Per visualizzare più dettagli relativi ai parametri in qualsiasi grafico, posiziona il puntatore del mouse sul grafico e scegli l'icona delle operazioni, View in metrics (Visualizza nei parametri).

      Il grafico viene visualizzato in una nuova scheda, con i parametri pertinenti elencati sotto il grafico. Puoi personalizzare la visualizzazione di questo grafico, modificando i parametri e le risorse mostrate, le statistiche, il periodo e altri fattori per comprendere meglio la situazione attuale.

b. Puoi visualizzare eventi di log dall'intervallo temporale mostrato nel grafico. Questo consente di scoprire eventi che si sono verificati nell'infrastruttura che causano una variazione imprevista dei parametri.

   Per visualizzare eventi di log, posiziona il puntatore del mouse sul grafico e scegliere l'icona delle operazioni, View in logs (Visualizza in log).

   La vista CloudWatch Registri viene visualizzata in una nuova scheda, che mostra un elenco dei gruppi di log. Per visualizzare gli eventi di log in uno di questi gruppi di log che si sono verificati durante l'intervallo di tempo mostrato nel grafico originale, scegli tale gruppo di log.

6. Durante la visualizzazione degli allarmi, puoi concentrarti su un particolare allarme in diversi modi:

   - Per visualizzare ulteriori informazioni relative a un allarme, posiziona il puntatore del mouse sull'allarme e scegli l'icona delle operazioni, View in alarms (Visualizza in allarmi).

     La visualizzazione allarmi appare in una nuova scheda, insieme a un elenco degli allarmi e ai dettagli relativi all'allarme scelto. Per visualizzare la cronologia per questo allarme, seleziona la scheda History (Cronologia).

7. Gli allarmi vengono sempre aggiornati una volta al minuto. Per aggiornare la visualizzazione, scegli l'icona di aggiornamento (due frecce curve) nella parte superiore destra della schermata. Per modificare la frequenza di aggiornamento automatico per gli elementi sulla schermata diversa da allarmi, scegli la freccia giù accanto all'icona di aggiornamento e selezionare la frequenza di aggiornamento. Puoi anche scegliere di disattivare l'aggiornamento automatico.
8. Per modificare l'intervallo di tempo mostrato in tutti i grafici e gli allarmi attualmente visualizzati, accanto a Time range (Intervallo di tempo) nella parte superiore della schermata, scegli l'intervallo. Per scegliere tra più opzioni di intervallo di tempo rispetto a quelle visualizzate per impostazione predefinita, seleziona custom (personalizzato) .
9. Per tornare al pannello di controllo dei servizi trasversali, scegli Overview (Panoramica) nell'elenco nella parte superiore della schermata che mostra attualmente il servizio che si sta esaminando.

   In alternativa, da qualsiasi visualizzazione, puoi scegliere nella parte CloudWatchsuperiore dello schermo di cancellare tutti i filtri e tornare alla pagina di panoramica.

Consulta la dashboard multi-service preconfigurata

Puoi passare alla schermata Cross-service dashboard e interagire con le dashboard di tutti i AWS servizi che stai utilizzando. La CloudWatch console mostra le dashboard in ordine alfabetico e mostra una o due metriche chiave su ciascuna dashboard.

> **Note**
>
> Se utilizzi cinque o più AWS servizi, la CloudWatch Console non mostrerà la dashboard Cross-service nella schermata Panoramica.

Per aprire il pannello di controllo dei servizi trasversali

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

   Viene visualizzata la schermata Overview (Panoramica).
2. Nella schermata Overview (Panoramica), seleziona il menu a discesa Overview (Panoramica) e quindi scegli Cross service dashboard (Pannello di controllo dei servizi trasversali).

   Viene visualizzata la schermata del pannello di controllo dei servizi trasversali.
3. (Facoltativo) Se usi l'interfaccia originale, scorri fino alla sezione Cross-service dashboard (Pannello di controllo dei servizi trasversali) e quindi scegli View Cross-service dashboard (Visualizza pannello di controllo dei servizi trasversali).

   Viene visualizzata la schermata del pannello di controllo dei servizi trasversali.
4. Puoi concentrarti su un particolare servizio in due modi:

   a. Per visualizzare più parametri chiave per un servizio, scegli il suo nome dall'elenco nella parte superiore della schermata, in cui è attualmente visualizzato Cross service dashboard (Pannello di controllo dei servizi trasversali). In alternativa, puoi scegliere View Service dashboard (Visualizza pannello di controllo del servizio) accanto al nome del servizio.

      Viene visualizzato un pannello di controllo automatico per il servizio, che mostra più parametri per tale servizio. Inoltre, per alcuni servizi, nella parte inferiore del pannello di controllo del servizio vengono visualizzate risorse correlate a tale servizio. Puoi scegliere una di tali risorse per la console del servizio e concentrarti ulteriormente sulla risorsa.

b. Per visualizzare tutti gli allarmi correlati a un servizio, scegli il pulsante a destra della schermata accanto al nome del servizio. Il testo su questo pulsante indica il numero di allarmi creati in questo servizio e se alcuni sono nello stato ALARM.

   Quando vengono visualizzati gli allarmi, più allarmi che hanno impostazioni simili (ad esempio dimensioni, soglia, o periodo) possono essere visualizzati in un singolo grafico.

   Puoi visualizzare i dettagli relativi a un allarme e la cronologia dell'allarme. A questo scopo, posiziona il puntatore del mouse sul grafico di allarme e scegli l'icona delle operazioni, View in alarms (Visualizza in allarmi).

   La visualizzazione allarmi appare in una nuova scheda del browser, insieme un elenco degli allarmi e ai dettagli relativi all'allarme scelto. Per visualizzare la cronologia per questo allarme, seleziona la scheda History (Cronologia).

5. Puoi concentrarti su risorse in un particolare gruppo di risorse. A questo scopo, scegli il gruppo di risorse dall'elenco nella parte superiore della pagina dove è visualizzato All resources (Tutte le risorse).

   Per ulteriori informazioni, consulta Visualizza una dashboard predefinita per un gruppo di risorse.

6. Per modificare l'intervallo di tempo mostrato in tutti i grafici e gli allarmi attualmente visualizzati, seleziona l'intervallo desiderato accanto a Time range (Intervallo di tempo) nella parte superiore della schermata. Scegli custom (personalizzato) per scegliere tra più opzioni di intervallo di tempo di quelle visualizzate per impostazione predefinita.
7. Gli allarmi vengono sempre aggiornati una volta al minuto. Per aggiornare la visualizzazione, scegli l'icona di aggiornamento (due frecce curve) nella parte superiore destra della schermata. Per modificare la frequenza di aggiornamento automatico per gli elementi della schermata diversi da allarmi, scegli la freccia giù accanto all'icona di aggiornamento e seleziona la frequenza di aggiornamento desiderata. Puoi anche scegliere di disattivare l'aggiornamento automatico.

Rimuovere un servizio dalla dashboard inter-service

Puoi impedire la visualizzazione dei parametri di un servizio nel pannello di controllo dei servizi trasversali. In questo modo puoi dedicare il pannello di controllo dei servizi trasversali ai servizi che più desideri monitorare.

Se rimuovi un servizio dal pannello di controllo dei servizi trasversali, gli allarmi per tale servizio appariranno ancora nelle visualizzazioni degli allarmi.

Per rimuovere i parametri di un servizio dal pannello di controllo dei servizi trasversali

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

   Viene visualizzata la home page.
2. Nella parte superiore della pagina, in Overview (Panoramica), scegli il servizio da rimuovere.

   La visualizzazione cambia per visualizzare solo i parametri di tale servizio.
3. Scegli Actions (Operazioni), quindi deseleziona la casella di controllo accanto a Show on cross service dashboard (Mostra nel pannello di controllo dei servizi trasversali).

Visualizza una dashboard predefinita per un gruppo di risorse

Puoi concentrare la visualizzazione per mostrare parametri e allarmi di un singolo gruppo di risorse. L'uso di gruppi di risorse consente di utilizzare tag per organizzare progetti, concentrarsi su un sottoinsieme dell'architettura o distinguere tra ambienti di produzione e sviluppo. Inoltre, ti consentono di concentrarti su ciascuno di questi gruppi di risorse nella CloudWatch panoramica. Per ulteriori informazioni, consulta Che cos'è AWS Resource Groups

Quando ti concentri su un gruppo di risorse, la visualizzazione cambia per mostrare solo i servizi in cui sono presenti risorse taggate in questo gruppo di risorse. L'area degli allarmi recenti mostra solo gli allarmi associati alle risorse che fanno parte del gruppo di risorse. Inoltre, se è stata creata una dashboard con il nome CloudWatch-Default- ResourceGroupName, questa viene visualizzata nell'area Dashboard predefinita.

Puoi approfondire ulteriormente concentrandoti contemporaneamente su un singolo AWS servizio e su un gruppo di risorse. La procedura seguente spiega solo come concentrarsi su un gruppo di risorse.

Per concentrarti su un singolo gruppo di risorse

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nella parte superiore della pagina, in cui è visualizzato All resources (Tutte le risorse), scegli un gruppo di risorse.
3. Per visualizzare ulteriori parametri correlati a questo gruppo di risorse, nella parte inferiore della schermata, scegli View cross service dashboard (Visualizza pannello di controllo dei servizi trasversali).

Viene visualizzato il pannello di controllo dei servizi trasversali che mostra solo i servizi correlati a questo gruppo di risorse. Per ogni servizio, vengono visualizzati uno o due parametri chiave.

4. Per modificare l'intervallo di tempo mostrato in tutti i grafici e gli allarmi attualmente visualizzati, in Time range (Intervallo di tempo) nella parte superiore della schermata, seleziona un intervallo. Per scegliere tra più opzioni di intervallo di tempo rispetto a quelle visualizzate per impostazione predefinita, seleziona custom (personalizzato) .
5. Gli allarmi vengono sempre aggiornati una volta al minuto. Per aggiornare la visualizzazione, scegli l'icona di aggiornamento (due frecce curve) nella parte superiore destra della schermata. Per modificare la frequenza di aggiornamento automatico per gli elementi sulla schermata diversa da allarmi, scegli la freccia giù accanto all'icona di aggiornamento e selezionare la frequenza di aggiornamento. Puoi anche scegliere di disattivare l'aggiornamento automatico.
6. Per tornare a mostrare informazioni su tutte le risorse nell'account, nella parte superiore della schermata in cui è attualmente visualizzato il nome del gruppo di risorse, scegli All resources (Tutte le risorse).

Visualizza la dashboard multi-service preconfigurata

Puoi passare alla schermata della dashboard Cross-service e interagire con le dashboard di tutti i AWS servizi che stai utilizzando. La CloudWatch console mostra le dashboard in ordine alfabetico e mostra una o due metriche chiave per ciascun servizio.

> **Note**
>
> Se utilizzi cinque o più AWS servizi, la CloudWatch Console non mostrerà la dashboard Cross-service nella schermata Panoramica.

Per aprire il pannello di controllo dei servizi trasversali

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

   Viene visualizzata la schermata Overview (Panoramica).
2. Nella schermata Overview (Panoramica), seleziona il menu a discesa Overview (Panoramica) e quindi scegli Cross service dashboard (Pannello di controllo dei servizi trasversali).

   Viene visualizzata la schermata del pannello di controllo dei servizi trasversali.

3. (Facoltativo) Se usi l'interfaccia originale, scorri fino alla sezione Cross-service dashboard (Pannello di controllo dei servizi trasversali) e quindi scegli View Cross-service dashboard (Visualizza pannello di controllo dei servizi trasversali).

   Viene visualizzata la schermata del pannello di controllo dei servizi trasversali.

4. Puoi concentrarti su un particolare servizio in due modi:

   a. Per visualizzare più parametri chiave per un servizio, scegli il suo nome dall'elenco nella parte superiore della schermata, in cui è attualmente visualizzato Cross service dashboard (Pannello di controllo dei servizi trasversali). In alternativa, puoi scegliere View Service dashboard (Visualizza pannello di controllo del servizio) accanto al nome del servizio.

      Viene visualizzato un pannello di controllo automatico per il servizio, che mostra più parametri per tale servizio. Inoltre, per alcuni servizi, nella parte inferiore del pannello di controllo del servizio vengono visualizzate risorse correlate a tale servizio. Puoi scegliere una di tali risorse per la console del servizio e concentrarti ulteriormente sulla risorsa.

   b. Per visualizzare tutti gli allarmi correlati a un servizio, scegli il pulsante a destra della schermata accanto al nome del servizio. Il testo su questo pulsante indica il numero di allarmi creati in questo servizio e se alcuni sono nello stato ALARM.

      Quando vengono visualizzati gli allarmi, più allarmi che hanno impostazioni simili (ad esempio dimensioni, soglia, o periodo) possono essere visualizzati in un singolo grafico.

      Puoi visualizzare i dettagli relativi a un allarme e la cronologia dell'allarme. A questo scopo, posiziona il puntatore del mouse sul grafico di allarme e scegli l'icona delle operazioni, View in alarms (Visualizza in allarmi).

      La visualizzazione allarmi appare in una nuova scheda del browser, insieme un elenco degli allarmi e ai dettagli relativi all'allarme scelto. Per visualizzare la cronologia per questo allarme, seleziona la scheda History (Cronologia).

5. Puoi concentrarti su risorse in un particolare gruppo di risorse. A questo scopo, scegli il gruppo di risorse dall'elenco nella parte superiore della pagina dove è visualizzato All resources (Tutte le risorse).

   Per ulteriori informazioni, consulta Visualizza una dashboard predefinita per un gruppo di risorse.

6. Per modificare l'intervallo di tempo mostrato in tutti i grafici e gli allarmi attualmente visualizzati, seleziona l'intervallo desiderato accanto a Time range (Intervallo di tempo) nella parte superiore

della schermata. Scegli custom (personalizzato) per scegliere tra più opzioni di intervallo di tempo di quelle visualizzate per impostazione predefinita.

7. Gli allarmi vengono sempre aggiornati una volta al minuto. Per aggiornare la visualizzazione, scegli l'icona di aggiornamento (due frecce curve) nella parte superiore destra della schermata. Per modificare la frequenza di aggiornamento automatico per gli elementi della schermata diversi da allarmi, scegli la freccia giù accanto all'icona di aggiornamento e seleziona la frequenza di aggiornamento desiderata. Puoi anche scegliere di disattivare l'aggiornamento automatico.

Rimozione della visualizzazione di un servizio dal pannello di controllo dei servizi trasversali

Puoi impedire la visualizzazione dei parametri di un servizio nel pannello di controllo dei servizi trasversali. In questo modo puoi dedicare il pannello di controllo dei servizi trasversali ai servizi che più desideri monitorare.

Se rimuovi un servizio dal pannello di controllo dei servizi trasversali, gli allarmi per tale servizio appariranno ancora nelle visualizzazioni degli allarmi.

Per rimuovere i parametri di un servizio dal pannello di controllo dei servizi trasversali

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

   Viene visualizzata la home page.
2. Nella parte superiore della pagina, in Overview (Panoramica), scegli il servizio da rimuovere.

   La visualizzazione cambia per visualizzare solo i parametri di tale servizio.
3. Scegli Actions (Operazioni), quindi deseleziona la casella di controllo accanto a Show on cross service dashboard (Mostra nel pannello di controllo dei servizi trasversali).

Visualizza una dashboard predefinita per un singolo servizio AWS

Nella CloudWatch home page, puoi concentrare la visualizzazione su un singolo AWS servizio. Puoi approfondire ulteriormente concentrandoti contemporaneamente su un singolo AWS servizio e su un gruppo di risorse. La procedura seguente mostra solo come concentrarsi su un AWS servizio.

Per concentrarti su un singolo servizio

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

   Viene visualizzata la home page.
2. Per Panoramica, dove la panoramica è attualmente mostrata nel menu a discesa, scegli Pannelli di controllo dei servizi.
3. Scegli il servizio su cui vuoi concentrarti.

   La visualizzazione cambia per visualizzare grafici di parametri chiave del servizio selezionato.
4. Per passare alla visualizzazione degli allarmi per questo servizio, seleziona la casella di controllo In allarme, Dati insufficienti o OK nella parte superiore della schermata in cui è attualmente visualizzato il nome del servizio.
5. Durante la visualizzazione dei parametri, puoi concentrarti su un particolare parametro in diversi modi:

   a. Per visualizzare più dettagli relativi ai parametri in qualsiasi grafico, posiziona il puntatore del mouse sul grafico e scegli l'icona delle operazioni, View in metrics (Visualizza nei parametri).

      Il grafico viene visualizzato in una nuova scheda, con i parametri pertinenti elencati sotto il grafico. Puoi personalizzare la visualizzazione di questo grafico, modificando i parametri e le risorse mostrate, le statistiche, il periodo e altri fattori per comprendere meglio la situazione attuale.

   b. Puoi visualizzare eventi di log dall'intervallo temporale mostrato nel grafico. Questo consente di scoprire eventi che si sono verificati nell'infrastruttura che causano una variazione imprevista dei parametri.

      Per visualizzare eventi di log, posiziona il puntatore del mouse sul grafico e scegliere l'icona delle operazioni, View in logs (Visualizza in log).

      La vista CloudWatch Registri viene visualizzata in una nuova scheda, che mostra un elenco dei gruppi di log. Per visualizzare gli eventi di log in uno di questi gruppi di log che si sono verificati durante l'intervallo di tempo mostrato nel grafico originale, scegli tale gruppo di log.
6. Durante la visualizzazione degli allarmi, puoi concentrarti su un particolare allarme in diversi modi:

   - Per visualizzare ulteriori informazioni relative a un allarme, posiziona il puntatore del mouse sull'allarme e scegli l'icona delle operazioni, View in alarms (Visualizza in allarmi).

La visualizzazione allarmi appare in una nuova scheda, insieme a un elenco degli allarmi e ai dettagli relativi all'allarme scelto. Per visualizzare la cronologia per questo allarme, seleziona la scheda History (Cronologia).

7. Gli allarmi vengono sempre aggiornati una volta al minuto. Per aggiornare la visualizzazione, scegli l'icona di aggiornamento (due frecce curve) nella parte superiore destra della schermata. Per modificare la frequenza di aggiornamento automatico per gli elementi sulla schermata diversa da allarmi, scegli la freccia giù accanto all'icona di aggiornamento e selezionare la frequenza di aggiornamento. Puoi anche scegliere di disattivare l'aggiornamento automatico.
8. Per modificare l'intervallo di tempo mostrato in tutti i grafici e gli allarmi attualmente visualizzati, accanto a Time range (Intervallo di tempo) nella parte superiore della schermata, scegli l'intervallo. Per scegliere tra più opzioni di intervallo di tempo rispetto a quelle visualizzate per impostazione predefinita, seleziona custom (personalizzato) .
9. Per tornare al pannello di controllo dei servizi trasversali, scegli Overview (Panoramica) nell'elenco nella parte superiore della schermata che mostra attualmente il servizio che si sta esaminando.

   In alternativa, da qualsiasi visualizzazione, puoi scegliere nella parte CloudWatchsuperiore dello schermo di cancellare tutti i filtri e tornare alla pagina di panoramica.

Visualizza una dashboard predefinita per un gruppo di risorse

Puoi concentrare la visualizzazione per mostrare parametri e allarmi di un singolo gruppo di risorse. L'uso di gruppi di risorse consente di utilizzare tag per organizzare progetti, concentrarsi su un sottoinsieme dell'architettura o distinguere tra ambienti di produzione e sviluppo. Inoltre, ti consentono di concentrarti su ciascuno di questi gruppi di risorse nella CloudWatch panoramica. Per ulteriori informazioni, consulta Che cos'è AWS Resource Groups

Quando ti concentri su un gruppo di risorse, la visualizzazione cambia per mostrare solo i servizi in cui sono presenti risorse taggate in questo gruppo di risorse. L'area degli allarmi recenti mostra solo gli allarmi associati alle risorse che fanno parte del gruppo di risorse. Inoltre, se è stata creata una dashboard con il nome CloudWatch-Default- ResourceGroupName, questa viene visualizzata nell'area Dashboard predefinita.

Puoi approfondire ulteriormente concentrandoti contemporaneamente su un singolo AWS servizio e su un gruppo di risorse. La procedura seguente mostra solo come concentrarsi su un gruppo di risorse.

Per concentrarti su un singolo gruppo di risorse

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nella parte superiore della pagina, in cui è visualizzato All resources (Tutte le risorse), scegli un gruppo di risorse.
3. Per visualizzare ulteriori parametri correlati a questo gruppo di risorse, nella parte inferiore della schermata, scegli View cross service dashboard (Visualizza pannello di controllo dei servizi trasversali).

   Viene visualizzato il pannello di controllo dei servizi trasversali che mostra solo i servizi correlati a questo gruppo di risorse. Per ogni servizio, vengono visualizzati uno o due parametri chiave.
4. Per modificare l'intervallo di tempo mostrato in tutti i grafici e gli allarmi attualmente visualizzati, in Time range (Intervallo di tempo) nella parte superiore della schermata, seleziona un intervallo. Per scegliere tra più opzioni di intervallo di tempo rispetto a quelle visualizzate per impostazione predefinita, seleziona custom (personalizzato) .
5. Gli allarmi vengono sempre aggiornati una volta al minuto. Per aggiornare la visualizzazione, scegli l'icona di aggiornamento (due frecce curve) nella parte superiore destra della schermata. Per modificare la frequenza di aggiornamento automatico per gli elementi sulla schermata diversa da allarmi, scegli la freccia giù accanto all'icona di aggiornamento e selezionare la frequenza di aggiornamento. Puoi anche scegliere di disattivare l'aggiornamento automatico.
6. Per tornare a mostrare informazioni su tutte le risorse nell'account, nella parte superiore della schermata in cui è attualmente visualizzato il nome del gruppo di risorse, scegli All resources (Tutte le risorse).

CloudWatch fatturazione e costi

Questa sezione descrive come le CloudWatch funzionalità di Amazon generano costi. Fornisce inoltre metodi che possono aiutarti ad analizzare, ottimizzare e ridurre CloudWatch i costi. In questa sezione, a volte facciamo riferimento ai prezzi per descrivere CloudWatch le funzionalità. Per informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Argomenti

Analizza i dati di CloudWatch costi e utilizzo con Cost Explorer

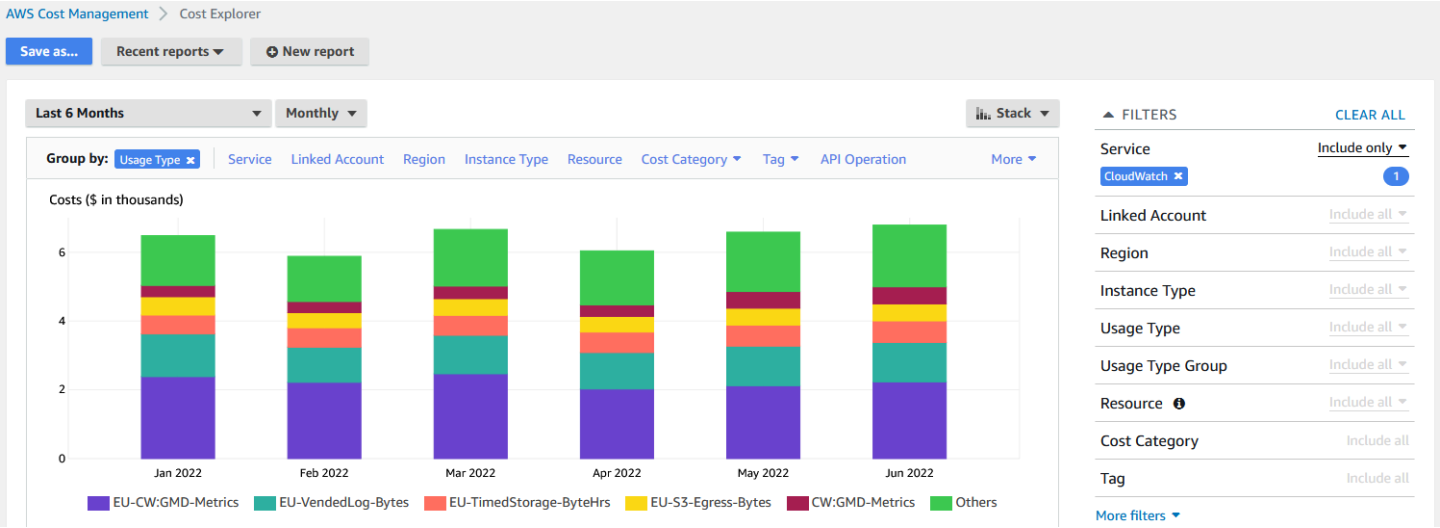
Con AWS Cost Explorer, puoi visualizzare e analizzare i dati sui costi e sull'utilizzo Servizi AWS nel tempo, tra cui. CloudWatch Per ulteriori informazioni, consulta Guida introduttiva AWS Cost Explorer.

La procedura seguente descrive come utilizzare Cost Explorer per visualizzare e analizzare i dati di CloudWatch costi e utilizzo.

Per visualizzare e analizzare i dati relativi a CloudWatch costi e utilizzo

1. Accedi alla console di Cost Explorer all'indirizzo https://console.aws.amazon.com/cost-management/home#/custom.
2. In FILTRI, per Assistenza, seleziona CloudWatch.
3. Per Group by (Gruppo per), scegli Usage Type (Tipo di utilizzo). Puoi anche raggruppare i risultati in base ad altre categorie, ad esempio le seguenti:
   - Operazione API: scopri quali operazioni API hanno generato la maggior parte dei costi.
   - Regione: scopri quali regioni hanno generato la maggior parte dei costi.

L'immagine seguente mostra un esempio dei costi generati dalle CloudWatch funzionalità nell'arco di sei mesi.



Per vedere quali CloudWatch funzionalità hanno generato il maggior numero di costi, guarda i valori di`UsageType`. Ad esempio, `EU-CW:GMD-Metrics` rappresenta i costi generati dalle richieste API in CloudWatch blocco.

> **Note**
>
> Le stringhe per `UsageType` corrispondono a caratteristiche e regioni specifiche. Ad esempio, la prima parte di `EU-CW:GMD-Metrics` (EU) corrisponde alla regione Europa (Irlanda) e la seconda parte di `EU-CW:GMD-Metrics` (`GMD-Metrics`) corrisponde alle richieste API in CloudWatch blocco.
>
> L'intera stringa per `UsageType` può essere formattata come segue: `<Region>-CW:<Feature>` o `<Region>-<Feature>`.
>
> Per migliorare la leggibilità, le stringhe per `UsageType` nelle tabelle di questo documento sono state abbreviate utilizzando i loro suffissi. Ad esempio, `EU-CW:GMD-Metrics` è abbreviata in `GMD-Metrics`.

La tabella seguente include i nomi di ciascuna CloudWatch funzionalità, elenca i nomi di ciascuna sottofunzionalità ed elenca le stringhe per. `UsageType`

CloudWatch caratteristica	CloudWatch caratteristica secondaria	**`UsageType`**
CloudWatch metriche	Parametri personalizzati	`MetricMonitorUsage`

CloudWatch caratteristica	CloudWatch caratteristica secondaria	**`UsageType`**
	Monitoraggio dettagliato	`MetricMonitorUsage`
	Parametri incorporati	`MetricMonitorUsage`
CloudWatch richieste API	Richieste API	`Requests`
	Bulk (Ottieni)	`GMD-Metrics`
	Contributor Insights	`GIRR-Metrics`
	Snapshot di immagini bitmap	`GMWI-Metrics`
CloudWatch flussi metrici	Flussi di parametri	`MetricStreamUsage`
CloudWatch cruscotti	Pannello di controllo con 50 parametri o meno	`DashboardsUsageHour-Basic`
	Pannello di controllo con più di 50 parametri	`DashboardsUsageHour`
CloudWatch allarmi	Standard (allarme parametro)	`AlarmMonitorUsage`
	Ad alta risoluzione (allarme parametro)	`HighResAlarmMonitorUsage`
	Allarme per le query di Approfondimenti sulle metriche	`MetricInsightAlarmUsage`

CloudWatch caratteristica	CloudWatch caratteristica secondaria	**`UsageType`**
	Composito (allarme aggregato)	`CompositeAlarmMonitorUsage`
CloudWatch Segnali applicativi	Segnali applicativi	`Application-Signals`
CloudWatch registri personalizzati	Raccogli (importazione)	`DataProcessing-Bytes`
	Archiviazione (archivio)	`TimedStorage-ByteHrs`
	Analizza (query)	`DataScanned-Bytes`
CloudWatch Registri di accesso non frequenti	Raccogli (importazione)	`DataProcessingIA-Bytes`
CloudWatch registri venduti	Spedizione (Amazon CloudWatch Logs)	`VendedLog-Bytes`
	Consegna (CloudWatch registri, registri di accesso non frequenti)	`VendedLogIA-Bytes`
	Consegna (Amazon Simple Storage Service)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`
	Consegna (Amazon Data Firehose)	`FH-Egress-Bytes`
Contributor Insights	CloudWatch Registri (regole)	`ContributorInsightRules`

CloudWatch caratteristica	CloudWatch caratteristica secondaria	**`UsageType`**
	CloudWatch Registri (eventi)	`ContributorInsightEvents`
	Amazon DynamoDB (regole)	`ContributorRulesManaged`
	Eventi DynamoDB	`ContributorEventsManaged`
Canary (Synthetics)	Esegui	`Canary-runs`
Evidently	Eventi	`Evidently-event`
	Unità di analisi	`Evidently-eau`
RUM	Eventi	`RUM-event`

Analizza i dati di CloudWatch costi e utilizzo con AWS Cost and Usage Report s e Athena

Un altro modo per analizzare i dati di CloudWatch costi e utilizzo consiste nell'utilizzare AWS Cost and Usage Report s con Amazon Athena. AWS Cost and Usage Report s contengono un set completo di dati su costi e utilizzo. Puoi creare report che tengono traccia dei costi e dell'utilizzo e puoi pubblicare questi report in un bucket S3 a scelta dell'utente. Puoi anche scaricare ed eliminare i report dal bucket S3. Per ulteriori informazioni, consulta What are AWS Cost and Usage Report s? nella Guida AWS Cost and Usage Report per l'utente s.

> **Note**
>
> Non ci sono costi per l'utilizzo di AWS Cost and Usage Report s. Paghi l'archiviazione solo quando pubblichi i report su Amazon Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta Quote e restrizioni nella Guida per l'utente di AWS Cost and Usage Report.

Athena è un servizio di interrogazione che puoi utilizzare con AWS Cost and Usage Report s per analizzare i dati di costi e utilizzo. Puoi eseguire query sui report nel bucket S3 senza bisogno di scaricarli prima. Per ulteriori informazioni, consulta Che cos'è Amazon Athena? nella Guida per l'utente di Amazon Athena. Per ulteriori informazioni, consulta Che cos'è Amazon Athena? nella Guida per l'utente di Amazon Athena. Per ulteriori informazioni sui prezzi, consulta Prezzi di Amazon Athena.

La procedura seguente descrive il processo per abilitare AWS Cost and Usage Report s e integrare il servizio con Athena. La procedura contiene due query di esempio che è possibile utilizzare per analizzare i dati CloudWatch sui costi e sull'utilizzo.

> **Note**
>
> Puoi utilizzare una qualsiasi delle query di esempio contenute in questo documento. Tutte le query di esempio in questo documento corrispondono a un database denominato costandusagereport e mostrano i risultati per il mese di aprile e l'anno 2022. Puoi modificare queste informazioni. Tuttavia, prima di eseguire una query, assicurati che il nome del database corrisponda al nome del database nella query.

Per analizzare i dati sui costi e sull'utilizzo con AWS Cost and Usage Report s e Athena

1. Abilita AWS Cost and Usage Report s. Per ulteriori informazioni, consulta Creazione di report su costi e utilizzo nella Guida per l'utente di AWS Cost and Usage Report.

   > **Tip**
   >
   > Quando crei i report, assicurati di selezionare Include resource IDs (Includi ID risorsa). In caso contrario, i rapporti non includeranno la colonna `line_item_resource_id`.

Questa riga consente di identificare ulteriormente i costi durante l'analisi dei dati relativi ai costi e all'utilizzo.

2. Integra AWS Cost and Usage Report s con Athena. Per ulteriori informazioni, consulta Configurazione di Athena utilizzando i AWS CloudFormation modelli nella Guida per l'utente AWS Cost and Usage Report s.
3. Esegui query sui rapporti relativi ai costi e all'utilizzo.

Esempio: query Athena

È possibile utilizzare la seguente query per mostrare quali CloudWatch funzionalità hanno generato il maggior numero di costi in un determinato mese.

```
SELECT
CASE
-- Metrics
WHEN line_item_usage_type LIKE '%%MetricMonitorUsage%%' THEN 'Metrics (Custom, Detailed
 monitoring management portal EMF)'
WHEN line_item_usage_type LIKE '%%Requests%%' THEN 'Metrics (API Requests)'
WHEN line_item_usage_type LIKE '%%GMD-Metrics%%' THEN 'Metrics (Bulk API Requests)'
WHEN line_item_usage_type LIKE '%%MetricStreamUsage%%' THEN 'Metric Streams'
-- Dashboard
WHEN line_item_usage_type LIKE '%%DashboardsUsageHour%%' THEN 'Dashboards'
-- Alarms
WHEN line_item_usage_type LIKE '%%AlarmMonitorUsage%%' THEN 'Alarms (Standard)'
WHEN line_item_usage_type LIKE '%%HighResAlarmMonitorUsage%%' THEN 'Alarms (High
 Resolution)'
WHEN line_item_usage_type LIKE '%%MetricInsightAlarmUsage%%' THEN 'Alarms (Metrics
 Insights)'
WHEN line_item_usage_type LIKE '%%CompositeAlarmMonitorUsage%%' THEN 'Alarms
 (Composite)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessing-Bytes%%' THEN 'Logs (Collect - Data
 Ingestion)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessingIA-Bytes%%' THEN 'Infrequent Access
 Logs (Collect - Data Ingestion)'
WHEN line_item_usage_type LIKE '%%TimedStorage-ByteHrs%%' THEN 'Logs (Storage -
 Archival)'
WHEN line_item_usage_type LIKE '%%DataScanned-Bytes%%' THEN 'Logs (Analyze - Logs
 Insights queries)'
```

```
-- Vended Logs
WHEN line_item_usage_type LIKE '%%VendedLog-Bytes%%' THEN 'Vended Logs (Delivered to
 CW)'
WHEN line_item_usage_type LIKE '%%VendedLogIA-Bytes%%' THEN 'Vended Infrequent Access
 Logs (Delivered to CW)'
WHEN line_item_usage_type LIKE '%%FH-Egress-Bytes%%' THEN 'Vended Logs (Delivered to
 Kinesis FH)'
WHEN (line_item_usage_type LIKE '%%S3-Egress-Bytes%%') OR (line_item_usage_type LIKE '%
%S3-Egress-
ComprBytes%%') THEN 'Vended Logs (Delivered to S3)'
-- Other
WHEN line_item_usage_type LIKE '%%Application-Signals%%' THEN 'Application Signals'
WHEN line_item_usage_type LIKE '%%Canary-runs%%' THEN 'Synthetics'
WHEN line_item_usage_type LIKE '%%Evidently%%' THEN 'Evidently'
WHEN line_item_usage_type LIKE '%%RUM-event%%' THEN 'RUM'
ELSE 'Others'
END AS UsageType,
-- REGEXP_EXTRACT(line_item_resource_id,'^(?:.+?:){5}(.+)$',1) as ResourceID,
-- SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
GROUP BY
1
ORDER BY
TotalSpend DESC,
UsageType;
```

Esempio: query Athena

La seguente query può essere usata per mostrare i risultati per `UsageType` e `Operation`. Questo mostra come le CloudWatch funzionalità hanno generato costi. I risultati mostrano anche i valori per `UsageQuantity` e `TotalSpend`, in modo da poter visualizzare i costi di utilizzo totali.

> **Tip**
>
> Per ulteriori informazioni su `UsageType`, aggiungi la riga seguente a questa query: `line_item_line_item_description`
> Questa riga crea una colonna denominata Description (Descrizione).

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
```

Best practice per ottimizzare e ridurre i costi

CloudWatch metriche

Molti Servizi AWS, come Amazon Elastic Compute Cloud (Amazon EC2), Amazon S3 e Amazon Data CloudWatch Firehose, inviano automaticamente i parametri gratuitamente a. Tuttavia, i parametri raggruppati nelle seguenti categorie possono comportare costi aggiuntivi:

- Parametri personalizzati, monitoraggio dettagliato e parametri incorporati
- Richieste API

- Flussi di parametri

Per ulteriori informazioni, consulta Usare i CloudWatch parametri di Amazon.

Parametri personalizzati, monitoraggio dettagliato e parametri incorporati

Parametri personalizzati

Puoi creare parametri personalizzati per organizzare i punti di dati in qualsiasi ordine e tasso.

Tutti i parametri personalizzati sono ripartiti proporzionalmente all'ora. Vengono misurati solo quando vengono inviati a. CloudWatch Per informazioni sui prezzi delle metriche, consulta la pagina dei prezzi di Amazon CloudWatch .

La tabella seguente elenca i nomi delle funzionalità secondarie pertinenti per le metriche. CloudWatch La tabella include le stringhe per `UsageType` e `Operation`, che possono aiutarti ad analizzare e identificare i costi correlati ai parametri.

> **Note**
>
> Per ottenere maggiori dettagli sui parametri elencati nella tabella seguente mentre esegui query sui dati relativi ai costi e all'utilizzo con Athena, abbina le stringhe per `Operation` con i risultati che vengono mostrati per `line_item_operation`.

CloudWatchfunzionalità secondaria	**`UsageType`**	**`Operation`**	Scopo
Parametri personalizzati	`MetricMonitorUsage`	`MetricStorage`	Parametri personalizzati
Monitoraggio dettagliato	`MetricMonitorUsage`	`MetricStorage:AWS/`*`{Service}`*	Monitoraggio dettagliato
Parametri incorporati	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Registri di parametri incorporati

CloudWatchfunzionalità secondaria	**`UsageType`**	**`Operation`**	Scopo
Filtri di log	`MetricMonitorUsage`	`MetricStorage:AWS/CloudWatchLogs`	Filtri per i parametri di gruppo di log

Monitoraggio dettagliato

CloudWatch dispone di due tipi di monitoraggio:

- Monitoraggio base

  Il monitoraggio di base è gratuito e abilitato automaticamente per tutti i Servizi AWS che supportano la funzionalità.
- Monitoraggio dettagliato

  Il monitoraggio dettagliato comporta costi e aggiunge diversi miglioramenti a seconda del. Servizio AWS Per ogni Servizio AWS che supporta il monitoraggio dettagliato, puoi scegliere se abilitarlo per quel servizio. Per ulteriori informazioni, consulta Monitoraggio di base e dettagliato.

> **Note**
>
> Altri Servizi AWS supportano il monitoraggio dettagliato e potrebbero fare riferimento a questa funzionalità con un nome diverso. Ad esempio, il monitoraggio dettagliato per Amazon S3 è indicato come parametri di richiesta.

Analogamente alle metriche personalizzate, il monitoraggio dettagliato viene ripartito proporzionalmente all'ora e misurato solo quando i dati vengono inviati a. CloudWatch Il monitoraggio dettagliato genera costi in base al numero di metriche a cui vengono inviate. CloudWatch Per ridurre i costi, abilita il monitoraggio dettagliato solo quando necessario. Per informazioni sui prezzi del monitoraggio dettagliato, consulta la pagina CloudWatch dei prezzi di Amazon.

Esempio: query Athena

La seguente query può essere usata per mostrare quali istanze EC2 hanno il monitoraggio dettagliato abilitato.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation='MetricStorage:AWS/EC2'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation,
line_item_line_item_description
ORDER BY line_item_operation
```

Parametri incorporati

Con il formato metrico CloudWatch integrato, puoi inserire i dati delle applicazioni come dati di registro, in modo da generare metriche utilizzabili. Per ulteriori informazioni, consultate Ingestione di log ad alta cardinalità e generazione di metriche con il formato metrico incorporato. CloudWatch

I parametri incorporati generano costi in base al numero di registri importati, al numero di registri archiviati e al numero di parametri personalizzati generati.

La tabella seguente elenca i nomi delle funzionalità secondarie pertinenti per il formato metrico incorporato. CloudWatch La tabella include le stringhe per `UsageType` e `Operation`, che possono aiutarti ad analizzare e identificare i costi.

CloudWatch funzionalità secondaria	**`UsageType`**	**`Operation`**	Scopo
Parametri personalizzati	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Registri di parametri incorporati
Importazione di registri	`DataProcessing-Bytes`	`PutLogEvents`	Carica un batch di log eventi sul gruppo di log o flusso di log specificato
Archiviazione dei log	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Memorizza i log per ora e i log per byte in Logs CloudWatch

Per analizzare i costi, usa AWS Cost and Usage Report s con Athena in modo da poter identificare quali metriche generano costi e determinare come vengono generati i costi.

Per sfruttare al meglio i costi generati dal formato metrico CloudWatch incorporato, evita di creare metriche basate su dimensioni ad alta cardinalità. In questo modo, CloudWatch non crea una metrica personalizzata per ogni combinazione di dimensioni unica. Per ulteriori informazioni, consulta Dimensioni.

Se utilizzi CloudWatch Container Insights per sfruttare il formato metrico incorporato, puoi utilizzare AWS Distro for Open Telemetry come alternativa per sfruttare al meglio i costi relativi alle metriche. Utilizza Container Insights per raccogliere, aggregare e riepilogare parametri e registri dalle applicazioni e dai microservizi containerizzati. Quando abiliti Container Insights, l' CloudWatch agente invia i log a, in modo che possa utilizzarli per generare CloudWatch metriche incorporate. Tuttavia, l' CloudWatch agente invia solo un numero fisso di metriche e ti vengono addebitate tutte le metriche disponibili, comprese quelle che non utilizzi. CloudWatch Con AWS Distro for Open Telemetry, puoi configurare e personalizzare le metriche e le dimensioni a cui inviare. CloudWatch Questo ti aiuta a ridurre il volume di dati e i costi generati da Container Insights. Per ulteriori informazioni, consulta le seguenti risorse:

- Utilizzo di Container Insights

- AWS Distro per Open Telemetry

Richieste API

CloudWatch ha i seguenti tipi di richieste API:

- Richieste API
- Bulk (Ottieni)
- Contributor Insights
- Snapshot di immagini bitmap

Le richieste API generano costi in base al tipo di richiesta e al numero di parametri richiesti.

Nella tabella seguente sono elencati i tipi di richieste API e include le stringhe per `UsageType` e `Operation`, che possono aiutarti ad analizzare e identificare i costi relativi alle API.

Tipo di richiesta API	**`UsageType`**	**`Operation`**	Scopo
Richieste API	`Requests`	`GetMetricStatistics`	Recupera statistiche per i parametri specificati
	`Requests`	`ListMetrics`	Elenca i parametri specificati
	`Requests`	`PutMetricData`	Pubblica punti dati metrici su CloudWatch
	`Requests`	`GetDashboard`	Visualizza i dettagli per i pannelli di controllo specificati
	`Requests`	`ListDashboards`	Elenca i pannelli di controllo presenti nell'account

Tipo di richiesta API	**`UsageType`**	**`Operation`**	Scopo
	`Requests`	`PutDashboard`	Crea o aggiorna un pannello di controllo
	`Requests`	`DeleteDashboards`	Elimina tutti i pannelli di controllo specificati
Bulk (Ottieni)	`GMD-Metrics`	`GetMetricData`	Recupera i valori delle metriche CloudWatch
Contributor Insights	`GIRR-Metrics`	`GetInsightRuleReport`	Restituisce i dati di serie temporali raccolti da una regola di Contributor Insights
Snapshot di immagini bitmap	`GMWI-Metrics`	`GetMetricWidgetImage`	Recupera un'istantanea di una o più CloudWatch metriche come immagine bitmap

Per analizzare i costi, utilizza Cost Explorer e raggruppa i risultati per API Operation (Operazione API).

I costi per le richieste API variano e quando superi il numero di chiamate API che ti vengono fornite entro il limite del piano gratuito, devi sostenere dei costi. AWS

> **Note**
>
> `GetMetricDatae GetMetricWidgetImage` non sono inclusi nel limite del piano AWS gratuito. Per ulteriori informazioni, consulta Utilizzo del piano AWS gratuito nella Guida AWS Billing per l'utente.

Le richieste API che in genere determinano i costi sono le richieste `Put` e `Get`.

`PutMetricData`

`PutMetricData` genera costi ogni volta che viene chiamato e può sostenere costi significativi a seconda del caso d'uso. Per ulteriori informazioni, PutMetricDataconsulta Amazon CloudWatch API Reference.

Per sfruttare al massimo i costi generati da `PutMetricData`, raggruppa più dati nelle tue chiamate API. A seconda del caso d'uso, prendi in considerazione l'utilizzo di CloudWatch Logs o del formato metrico CloudWatch incorporato per inserire i dati metrici. Per ulteriori informazioni, consulta le seguenti risorse:

- Che cos'è Amazon CloudWatch Logs? nella Guida per l'utente di Amazon CloudWatch Logs
- Inserimento di log ad alta cardinalità e generazione di metriche con formato metrico incorporato CloudWatch
- Ridurre i costi e concentrarsi sui nostri clienti con le metriche personalizzate CloudWatch integrate di Amazon

`GetMetricData`

`GetMetricData` può anche generare costi significativi. I casi d'uso comuni che determinano i costi riguardano strumenti di monitoraggio di terze parti che estraggono dati per generare informazioni. Per ulteriori informazioni, GetMetricDataconsulta Amazon CloudWatch API Reference.

Per ridurre i costi generati da `GetMetricData`, prendi in considerazione la possibilità di estrarre solo dati monitorati e utilizzati o considera di estrarre dati meno spesso. A seconda del caso d'uso, è possibile prendere in considerazione l'utilizzo di flussi di parametri anziché `GetMetricData`, in modo da poter inviare dati quasi in tempo reale a terze parti a un costo inferiore. Per ulteriori informazioni, consulta le seguenti risorse:

- Utilizzo dei flussi di parametri
- CloudWatch Metric Streams: invia i AWS parametri ai partner e alle tue app in tempo reale

`GetMetricStatistics`

A seconda del caso d'uso, potresti considerare l'utilizzo di `GetMetricStatistics` anziché di `GetMetricData`. Con `GetMetricData`, è possibile recuperare i dati in modo rapido e su larga scala. Tuttavia, `GetMetricStatistics` è incluso nel limite del piano AWS gratuito per un massimo di un milione di richieste API, il che può aiutarti a ridurre i costi se non hai bisogno di recuperare tante metriche e punti dati per chiamata. Per ulteriori informazioni, consulta le seguenti risorse:

- GetMetricStatisticsnell'Amazon CloudWatch API Reference
- Devo usare GetMetricData o GetMetricStatistics?

> **Note**
>
> I chiamanti esterni effettuano chiamate API. Attualmente, l'unico modo per identificare questi chiamanti è aprire una richiesta di supporto tecnico al CloudWatch team e chiedere informazioni su di loro. Per informazioni sulla creazione di una richiesta di supporto tecnico, vedi Come posso ottenere supporto tecnico da AWS? .

CloudWatch flussi metrici

Con i flussi CloudWatch metrici, puoi inviare metriche in modo continuo a AWS destinazioni e destinazioni di fornitori di servizi di terze parti.

I flussi di parametri generano costi in base al numero di aggiornamenti dei parametri. Gli aggiornamenti dei parametri includono sempre i valori per le seguenti statistiche:

- `Minimum`
- `Maximum`
- `Sample Count`
- `Sum`

Per ulteriori informazioni, consulta Statistiche che possono essere trasmesse.

Per analizzare i costi generati dai flussi CloudWatch metrici, usa AWS Cost and Usage Report s con Athena. In questo modo, è possibile identificare quali flussi di parametri generano costi e determinare come vengono generati i costi.

Esempio: query Athena

La seguente query può essere usata per monitorare quali flussi di parametri generano costi in base al relativo nome della risorsa Amazon (ARN).

```
SELECT
SPLIT_PART(line_item_resource_id,'/',2) AS "Stream Name",
```

```
line_item_resource_id as ARN,
SUM(CAST(line_item_unblended_cost AS decimal(16,2))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
AND line_item_usage_type LIKE '%%MetricStreamUsage%%'
GROUP BY line_item_resource_id
ORDER BY TotalSpend DESC
```

Per ridurre i costi generati dai flussi CloudWatch metrici, trasmetti in streaming solo le metriche che apportano valore alla tua azienda. Puoi anche interrompere o mettere in pausa qualsiasi flusso di parametri che non stai utilizzando.

CloudWatch allarmi

Con gli CloudWatch allarmi, puoi creare allarmi basati su una singola metrica, allarmi basati su una query di Metrics Insights e allarmi compositi che controllano altri allarmi.

> **Note**
>
> I costi degli allarmi di parametri e compositi sono ripartiti proporzionalmente all'ora. I costi per gli allarmi sono sostenuti solo fintanto che gli allarmi esistono. Per ottimizzare i costi, assicurati di non lasciarti alle spalle allarmi mal configurati o di basso valore. A tal fine, puoi automatizzare la pulizia degli CloudWatch allarmi che non ti servono più. Per ulteriori informazioni, consulta Automating Amazon CloudWatch Alarm Cleanup at Scale

Allarmi di parametri

Gli allarmi dei parametri hanno le seguenti impostazioni di risoluzione:

- Standard (valutato ogni 60 secondi)
- Alta risoluzione (valutato ogni 10 secondi)

Quando crei un allarme di parametro, i costi si basano sull'impostazione della risoluzione dell'allarme e sul numero di parametri a cui fa riferimento l'allarme. Ad esempio, un allarme di parametro che fa riferimento a un parametro comporta un costo parametro di allarme all'ora. Per ulteriori informazioni, consulta Usare gli CloudWatch allarmi Amazon.

Se crei un allarme di parametro contenente un'espressione matematica del parametro che fa riferimento a più parametri, dovrai sostenere un costo per ogni parametro di allarme a cui si fa riferimento nell'espressione matematica del parametro. Per informazioni su come creare un allarme metrico che contenga un'espressione matematica metrica, consulta Creazione di un CloudWatch allarme basato su un'espressione matematica metrica.

Se crei un allarme di rilevamento delle anomalie, in cui l'allarme analizza i dati dei parametri precedenti per creare un modello di valori attesi, dovrai sostenere un costo per ogni parametro di allarme a cui si fa riferimento nell'allarme più due parametri aggiuntivi, uno per i parametri della banda superiore e inferiore create dal modello di rilevamento delle anomalie. Per informazioni su come creare un allarme di rilevamento delle anomalie, vedere Creazione di un allarme basato sul rilevamento delle anomalie. CloudWatch

Allarmi per le query di Approfondimenti sulle metriche

Gli allarmi per le query di Approfondimenti sulle metriche rappresentano un tipo specifico di allarme, disponibile solo con risoluzione standard (valutata ogni 60 secondi).

Quando crei un allarme per le query di Approfondimenti sulle metriche, i costi si basano sul numero di parametri analizzati dalla query a cui fa riferimento l'allarme. Ad esempio, un allarme per le query di Approfondimenti sulle metriche che fa riferimento a una query il cui filtro corrisponde a dieci parametri comporta un costo orario di dieci parametri analizzati. Per ulteriori informazioni, consulta l'esempio di prezzo su Amazon CloudWatch Pricing.

Se crei un allarme contenente sia una query di Approfondimenti sulle metriche che un'espressione matematica del parametro, tale allarme viene segnalato come allarme per le query di Approfondimenti sulle metriche. Se l'allarme contiene un'espressione matematica del parametro che fa riferimento ad altre metriche, oltre a quelle analizzati dalla query di Approfondimenti sulle metriche, dovrai sostenere un costo aggiuntivo per ogni parametro di allarme a cui si fa riferimento nell'espressione matematica del parametro. Per informazioni su come creare un allarme metrico che contenga un'espressione matematica metrica, consulta Creazione di un CloudWatch allarme basato su un'espressione matematica metrica.

Allarmi compositi

Gli allarmi compositi contengono espressioni di regole che specificano come devono valutare gli stati di altri allarmi per determinare il loro stato. Gli allarmi compositi hanno un costo orario standard, indipendentemente dal numero di altri allarmi valutati. Gli allarmi a cui fanno riferimento gli allarmi compositi nelle espressioni delle regole comportano costi separati. Per ulteriori informazioni, consulta Creazione di allarmi compositi.

Tipi di utilizzo degli allarmi

La tabella seguente elenca i nomi delle caratteristiche secondarie rilevanti per gli allarmi. CloudWatch La tabella include le stringhe per `UsageType`, che possono aiutarti ad analizzare e identificare i costi relativi agli allarmi.

CloudWatchfunzionalità secondaria	**`UsageType`**
Allarme dei parametri standard	`AlarmMonitorUsage`
Allarme di parametro ad alta risoluzione	`HighResAlarmMonitorUsage`
Allarme per le query di Approfondimenti sulle metriche	`MetricInsightAlarmUsage`
Allarme composito	`CompositeAlarmMonitorUsage`

Riduzione dei costi degli allarmi

Per ottimizzare i costi generati dagli allarmi matematici metrici che aggregano quattro o più metriche, puoi aggregare i dati prima che vengano inviati a. CloudWatch In questo modo, puoi creare un allarme per un singolo parametro invece di uno che aggrega i dati per più parametri. Per ulteriori informazioni, consulta Pubblicazione di parametri personalizzati.

Per ottimizzare i costi generati dagli allarmi per le query di Approfondimenti sulle metriche, puoi assicurarti che il filtro utilizzato per la query corrisponda solo ai parametri che desideri monitorare.

Il modo migliore per ridurre i costi è rimuovere tutti gli allarmi non necessari o non utilizzati. Ad esempio, puoi eliminare gli allarmi che valutano le metriche emesse da risorse che non esistono più. AWS

Esempio: verifica della presenza di allarmi nello stato ***`INSUFFICIENT_DATA`*** con ***`DescribeAlarms`***

Se elimini una risorsa ma non gli allarmi dei parametri emessi dalla risorsa, gli allarmi continueranno ad esistere e passeranno allo stato `INSUFFICIENT_DATA`. Per verificare la presenza di allarmi presenti nello `INSUFFICIENT_DATA` stato, utilizzate il seguente AWS Command Line Interface comando ().AWS CLI

```
$ aws cloudwatch describe-alarms –state-value INSUFFICIENT_DATA
```

Altri modi per ridurre i costi sono descritti di seguito:

- Assicurati di creare allarmi per i parametri corretti.
- Assicurati di non avere alcun allarme abilitato nelle regioni in cui non stai lavorando.
- Ricorda che, sebbene gli allarmi compositi riducano il rumore, generano anche costi aggiuntivi.
- Quando decidi se creare un allarme standard o un allarme ad alta risoluzione, considera il caso d'uso e il valore che apporta ogni tipo di allarme.

CloudWatch Registri

Amazon CloudWatch Logs ha i seguenti tipi di log:

- Registri personalizzati (registri creati per le tue applicazioni)
- Log venduti (log che altri Servizi AWS, come Amazon Virtual Private Cloud (Amazon VPC) e Amazon Route 53, creano per tuo conto)

Per ulteriori informazioni sui log venduti, consulta Enabling logging from certain AWS services nella Amazon CloudWatch Logs User Guide.

I registri personalizzati e venduti generano costi in base al numero di registri che sono raccolti, archiviati, e analizzati. Separatamente, i log venduti generano costi per la consegna ad Amazon S3 e Firehose.

La tabella seguente elenca i nomi delle funzionalità di CloudWatch Logs e i nomi delle relative funzionalità secondarie. La tabella include le stringhe per `UsageType` e `Operation`, che possono aiutare ad analizzare e identificare i costi relativi ai log.

CloudWatch Funzionalità di registro	CloudWatch Sottofunzionalità dei registri	**`UsageType`**	**`Operation`**	Scopo
Registri personalizzati	Raccogli (importazione)	`DataProcessing-Bytes`	`PutLogEvents`	Carica un batch di registri in un flusso di log specifico
	Archiviazione (archivio))	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Memorizza i log per ora e i log per byte in Logs CloudWatch
	Analizza (query Logs Insights)	`DataScanned-Bytes`	`StartQuery`	Registra i dati scansionati dalle query di Logs Insights CloudWatch
Registri venduti	Consegna (registri) CloudWatch	`VendedLog-Bytes`	`PutLogEvents`	Carica un batch di registri in un flusso di log specifico
	Consegna (Amazon S3)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`	`LogDelivery`	Invia log forniti (CloudWatchAmazon S3 o Firehose)
	Consegna (Firehose)	`FH-Egress-Bytes`	`LogDelivery`	Invia log forniti (CloudWatchAmazon S3 o Firehose)

Per analizzare i costi, usa AWS Cost and Usage Report s con Athena, in modo da poter identificare quali registri generano costi e determinare come vengono generati i costi.

Esempio: query Athena

È possibile utilizzare la seguente query per tenere traccia dei log che generano costi in base all'ID della risorsa.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_resource_id AS ResourceID,
line_item_usage_type AS UsageType,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation IN
 ('PutLogEvents','HourlyStorageMetering','StartQuery','LogDelivery')
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
ORDER BY
TotalSpend DESC
```

Per sfruttare al meglio i costi generati dai CloudWatch registri, considera quanto segue:

- Registra solo gli eventi che apportano valore alla tua attività. In questo modo è possibile generare meno costi per l'importazione.
- Modifica le impostazioni di conservazione dei log, in modo da generare meno costi per l'archiviazione. Per ulteriori informazioni, consulta Change log data retention in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.

- Esegui le query che CloudWatch Logs Insights salva automaticamente nella tua cronologia. In questo modo, si generano meno costi per l'analisi. Per ulteriori informazioni, consulta Visualizza le query in esecuzione o la cronologia delle query nella Amazon CloudWatch Logs User Guide.
- Usa l' CloudWatch agente per raccogliere i log di sistema e delle applicazioni e inviarli a. CloudWatch In questo modo, puoi raccogliere solo i log eventi che soddisfano i tuoi criteri. Per ulteriori informazioni, consulta Amazon CloudWatch Agent adds Support for Log Filter Expressions.

Per ridurre i costi dei log venduti, considera il tuo caso d'uso e poi determina se i log devono essere inviati ad Amazon S3 o ad Amazon CloudWatch S3. Per ulteriori informazioni, consulta Logs sent to Amazon S3 nella CloudWatch Amazon Logs User Guide.

> **Tip**
>
> Se desideri utilizzare filtri metrici, filtri di abbonamento, CloudWatch Logs Insights e Contributor Insights, invia i log venduti a. CloudWatch
> In alternativa, se lavori con i registri di flusso VPC e li utilizzi per scopi di verifica e conformità, invia i registri venduti ad Amazon S3.
> Per informazioni su come tenere traccia degli addebiti generati dalla pubblicazione dei log di flusso VPC nei bucket S3, consulta Utilizzo di s e tag di allocazione dei costi per comprendere l'ingestione dei dati di VPC Flow AWS Cost and Usage Report Logs in Amazon S3.

Per ulteriori informazioni su come sfruttare al meglio i costi generati dai CloudWatch log, consulta Quale gruppo di log causa un aumento improvviso della mia fattura relativa ai log? CloudWatch .

Utilizzo delle CloudWatch dashboard di Amazon

Le CloudWatch dashboard di Amazon sono home page personalizzabili nella CloudWatch console che puoi utilizzare per monitorare le tue risorse in un'unica visualizzazione, anche quelle distribuite in diverse regioni. Puoi utilizzare CloudWatch le dashboard per creare visualizzazioni personalizzate delle metriche e degli allarmi relativi alle tue risorse. AWS

Con i pannelli di controllo puoi creare quanto segue:

- Una visualizzazione singola per i parametri e gli allarmi selezionati che consentono di valutare l'integrità delle tue risorse e applicazioni su una o più regioni. Puoi selezionare il colore utilizzato per ogni parametro su ogni grafico, in modo da monitorare facilmente lo stesso parametro su più grafici.
- Un playbook operativo che fornisce ai membri del team linee guida su come reagire a incidenti specifici durante gli eventi operativi.
- Una vista comune delle misure delle risorse e delle applicazioni critiche che possono essere condivise con i membri del team per velocizzare il flusso di comunicazione durante gli eventi operativi.

Se disponi di più AWS account, puoi impostare l'osservabilità CloudWatch tra account e quindi creare dashboard complete tra più account nei tuoi account di monitoraggio. Queste dashboard possono includere grafici di metriche provenienti dagli account di origine e dai widget di Logs Insights con query sui gruppi di CloudWatch log provenienti dagli account di origine. Inoltre, gli allarmi creati nell'account di monitoraggio possono controllare le metriche negli account di origine. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Puoi creare dashboard dalla console o utilizzando l'operazione o l'API. AWS CLI `PutDashboard` Puoi aggiungere i pannelli di controllo a un elenco dei preferiti, dal quale potrai accedere non solo ai pannelli di controllo preferiti ma anche a quelli consultati di recente. Per ulteriori informazioni, consulta la sezione Aggiungi un pannello di controllo all'elenco dei preferiti.

Per accedere ai CloudWatch dashboard, è necessario uno dei seguenti:

- La policy `AdministratorAccess`
- La policy `CloudWatchFullAccess`
- Una policy personalizzata che includa una o più di queste autorizzazioni specifiche:

- `cloudwatch:GetDashboard` e `cloudwatch:ListDashboards` per poter visualizzare i pannelli di controllo
- `cloudwatch:PutDashboard` per poter creare o modificare i pannelli di controllo
- `cloudwatch:DeleteDashboards` per poter eliminare i pannelli di controllo

Indice

Creazione di un pannello di controllo CloudWatch

Per iniziare, crea una CloudWatch dashboard. Puoi creare più pannelli di controllo e aggiungerli a un elenco dei preferiti. Nel tuo Account AWS puoi creare tutti i pannelli di controllo che desideri. Tutti i pannelli di controllo sono globali. Non sono specifici per una regione.

La procedura seguente mostra come creare una dashboard dalla CloudWatch console. Puoi utilizzare il l'operazione API `PutDashboard` per creare un pannello di controllo dall'interfaccia a riga di comando. L'operazione API contiene una stringa JSON che definisce il contenuto del pannello di controllo. Per ulteriori informazioni sulla creazione di un pannello di controllo con il funzionamento dell'`PutDashboard`API, PutDashboardconsulta Amazon CloudWatch API Reference.

> **Tip**
>
> Se stai creando un nuovo pannello di controllo con l'operazione API `PutDashboard`, puoi utilizzare la stringa JSON da un pannello di controllo già esistente.

Per creare un pannello di controllo dalla console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli Create dashboard (Crea pannello di controllo).
3. Nella finestra di dialogo Create new dashboard (Crea nuovo pannello di controllo), digita un nome per il pannello di controllo e scegli Create dashboard (Crea pannello di controllo).

   Se si utilizza il nome CloudWatch-Default o CloudWatch-Default- ***`ResourceGroupName`***, la dashboard viene visualizzata nella panoramica della CloudWatch home page in Dashboard predefinita. Per ulteriori informazioni, consulta Guida introduttiva ad Amazon CloudWatch.
4. Nella finestra di dialogo Add to this dashboard (Aggiungi a questo pannello di controllo), effettua una delle operazioni indicate di seguito:
   - Per aggiungere un grafico al pannello di controllo, seleziona Line (Linea) o Stacked area (Area in pila), quindi seleziona Configure (Configura). Nella finestra di dialogo Add metric graph (Aggiungi grafico dei parametri), seleziona i parametri da rappresentare graficamente e quindi seleziona Create widget (Crea widget). Se un parametro non viene visualizzato nella finestra di dialogo perché non ha pubblicato dati per oltre 14 giorni, è possibile aggiungerlo manualmente. Per ulteriori informazioni, consulta Rappresenta graficamente le metriche manualmente su una dashboard CloudWatch .
   - Per aggiungere al pannello di controllo un numero che rappresenti un parametro, scegli Number (Numero), quindi scegli Configure (Configura). Nella finestra di dialogo Add metric graph (Aggiungi grafico dei parametri), seleziona i parametri da rappresentare graficamente e quindi seleziona Create widget (Crea widget).
   - Per aggiungere un blocco di testo al pannello di controllo, scegli Text (Testo), quindi scegli Configure (Configura). Nella finestra di dialogo New text widget (Nuovo widget di testo), in Markdown, formatta il testo utilizzando Markdown, quindi scegli Create widget (Crea widget).
5. (Facoltativo) Per aggiungere un altro widget al pannello di controllo, scegli Add widget (Aggiungi widget) e ripeti il passaggio 4. Puoi ripetere questa fase diverse volte.

Per ogni grafico sul pannello di controllo, è presente un'icona delle informazioni in alto a destra. Scegli questa icona per visualizzare le descrizioni dei parametri nel grafico.

6. Seleziona Salva pannello di controllo.

CloudWatch dashboard di osservabilità tra account

Se disponi di più AWS account, puoi configurare l'osservabilità CloudWatch tra account e quindi creare dashboard complete su più account nei tuoi account di monitoraggio. Puoi cercare, visualizzare e analizzare senza problemi parametri, log e tracce senza limiti di account.

Per ulteriori informazioni sulla configurazione dell'osservabilità tra account, consulta CloudWatch . CloudWatch osservabilità tra più account

Con l' CloudWatch osservabilità tra account, puoi effettuare le seguenti operazioni in una dashboard di un account di monitoraggio:

- Cercare, visualizzare e creare grafici di metriche che risiedono negli account di origine. Un singolo grafico può includere metriche provenienti da più account.
- Creare allarmi nell'account di monitoraggio per controllare le metriche negli account di origine.
- Visualizza gli eventi di registro dei gruppi di log situati negli account di origine ed esegui le query di CloudWatch Logs Insights sui gruppi di log negli account di origine. Una singola query di CloudWatch Logs Insights in un account di monitoraggio può interrogare più gruppi di log in più account di origine contemporaneamente.
- Visualizza nodi da account di origine in una mappa di tracciamento in X-Ray. Puoi quindi filtrare la mappa in base ad account di origine specifici.

Quando si CloudWatch accede a un account di monitoraggio, in alto a destra di ogni pagina che supporta la funzionalità di osservabilità tra più account viene visualizzato un badge blu dell'account di monitoraggio.

Pannelli di controllo su più account tra più regioni

Puoi creare dashboard interregionali per più account, che riepilogano i CloudWatch dati di più AWS account e più regioni in un'unica dashboard. Da questo pannello di controllo di alto livello è possibile ottenere una visualizzazione dell'intera applicazione e anche analizzare pannelli di controllo più specifici senza dover accedere e uscire dagli account o spostarsi tra le regioni.

Puoi creare dashboard interregionali tra account in e in modo programmatico. AWS Management Console

Prerequisito

Prima di creare un pannello di controllo su più account tra più regioni è necessario abilitare almeno un account di condivisione e almeno un account di monitoraggio. Inoltre, per poter utilizzare la CloudWatch console per creare una dashboard per più account, è necessario abilitare la console per la funzionalità tra account. Per ulteriori informazioni, consulta Console per più account e più regioni CloudWatch .

Creazione e utilizzo di un pannello di controllo su più account tra più regioni con l'opzione AWS Management Console

Puoi utilizzare il AWS Management Console per creare una dashboard tra più account e più regioni.

Per creare un pannello di controllo su più account tra più regioni

1. Accedi all'account di monitoraggio.
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
4. Scegli un pannello di controllo o crearne uno nuovo.
5. Nella parte superiore della schermata, puoi passare dagli account alle regioni. Quando si crea il pannello di controllo, è possibile includere widget provenienti da più account e regioni. I widget includono grafici, allarmi e widget CloudWatch Logs Insights.

Creazione di un grafico con parametri provenienti da diversi account e regioni

1. Accedi all'account di monitoraggio.
2. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch
3. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri.
4. Seleziona l'account e la regione da cui desideri aggiungere i parametri. Puoi selezionare il tuo account e la tua regione dai relativi menu a discesa in alto a destra dello schermo.
5. Aggiungere i parametri desiderati al grafico. Per ulteriori informazioni, consulta la pagina Rappresentazione grafica dei parametri.
6. Ripetere le fasi 4 e 5 per aggiungere parametri da altri account e regioni.

7. (Opzionale) Scegli la scheda Graphed metrics (Parametri nel grafico) e aggiungi una funzione matematica dei parametri che utilizzi i parametri scelti. Per ulteriori informazioni, consulta la pagina Utilizzare la matematica dei parametri.

   È inoltre possibile impostare un singolo grafico per includere più funzioni `SEARCH`. Ogni ricerca può fare riferimento a un diverso account o regione.

8. Una volta terminato il grafico, scegli Actions (Operazioni), Add to dashboard (Aggiungi al pannello di controllo).

   Seleziona il pannello di controllo su più account e scegli Add to dashboard (Aggiungi al pannello di controllo).

Aggiunta di un allarme da un altro account al pannello di controllo su più account

1. Accedi all'account di monitoraggio.
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nella parte superiore della pagina, scegli l'account in cui si trova l'allarme.
4. Nel pannello di navigazione, seleziona Alarms (Allarmi).
5. Seleziona la casella di controllo accanto all'allarme che si desidera aggiungere e scegli Add to dashboard (Aggiungi al pannello di controllo).
6. Seleziona il pannello di controllo su più account a cui si desidera aggiungerlo e scegli Add to dashboard (Aggiungi al pannello di controllo).

Creare un pannello di controllo su più account tra più regioni in modo programmatico

Puoi utilizzare le AWS API e gli SDK per creare dashboard a livello di codice. Per ulteriori informazioni, consulta. PutDashboard

Per abilitare i pannelli di controllo su più account tra più regioni, sono stati aggiunti nuovi parametri alla struttura del corpo del pannello di controllo, come illustrato nella tabella e negli esempi seguenti. Per ulteriori informazioni sulla struttura complessiva del corpo dei pannelli di controllo, consulta Dashboard Body Structure and Syntax (Struttura del corpo e sintassi dei pannelli di controllo).

Parametro	Utilizzo	Ambito	Predefinita
`accountId`	Specificare l'ID dell'account in cui si trova il widget o il parametro.	Widget o parametro	Account attualmente connesso
`region`	Specifica la regione del parametro.	Widget o parametro	Regione corrente selezionata nella console

Negli esempi seguenti viene illustrata l'origine JSON per i widget in un pannello di controllo su più account tra più regioni.

Questo esempio imposta il campo `accountId` sull'ID dell'account di condivisione a livello di widget. Ciò specifica che tutte i parametri in questo widget proverranno da tale account e regione di condivisione.

```
{
    "widgets": [
        {
            ...
            "properties": {
              "metrics": [
                     …..
                     ],
                    "accountId": "111122223333",
                    "region": "us-east-1"
           }
       }
  ]
}
```

Questo esempio imposta il campo `accountId` in modo diverso a livello di ogni parametro. In questo esempio, i diversi parametri in questa espressione matematica dei parametri provengono da account di condivisione diversi e regioni diverse.

```
{
"widgets": [
    {
        ...
        "properties": {
```

```
            "metrics": [
                [ { "expression": "SUM(METRICS())", "label": "[avg: ${AVG}]
 Expression1", "id": "e1", "stat": "Sum" } ],
                [ "AWS/EC2", "CPUUtilization", { "id": "m2", "accountId":
 "5555666677778888", "region": "us-east-1", "label": "[avg: ${AVG}] ApplicationALabel
 " } ],
                [ ".", ".", { "id": "m1", "accountId": "9999000011112222", "region":
 "eu-west-1", "label": "[avg: ${AVG}] ApplicationBLabel" } ]
            ],
            "view": "timeSeries",
            "region":"us-east-1", ---> home region of the metric. Not present in above
 example
            "stacked": false,
            "stat": "Sum",
            "period": 300,
            "title": "Cross account example"
        }
    }
]
}
```

Questo esempio mostra un widget di allarme.

```
{
    "type": "metric",
    "x": 6,
    "y": 0,
    "width": 6,
    "height": 6,
    "properties": {
        "accountID": "111122223333",
        "title": "over50",
        "annotations": {
            "alarms": [
                "arn:aws:cloudwatch:us-east-1:379642911888:alarm:over50"
            ]
        },
        "view": "timeSeries",
        "stacked": false
    }
}
```

Questo esempio riguarda un widget CloudWatch Logs Insights.

```
{
    "type": "log",
    "x": 0,
    "y": 6,
    "width": 24,
    "height": 6,
    "properties": {
        "query": "SOURCE 'route53test' | fields @timestamp, @message\n| sort @timestamp
 desc\n| limit 20",
        "accountId": "111122223333",
        "region": "us-east-1",
        "stacked": false,
        "view": "table"
    }
}
```

Un altro modo per creare dashboard a livello di codice consiste nel crearne prima uno in e quindi copiare il AWS Management Console codice sorgente JSON di questo dashboard. Per fare ciò, caricare il pannello di controllo e scegliere Actions (Operazioni), View/edit source (Visualizza/modifica origine). È quindi possibile copiare questo pannello di controllo JSON da utilizzare come modello per crearne altri simili.

Creazione di pannelli di controllo flessibili con variabili del pannello di controllo

Utilizza le variabili del pannello di controllo per creare pannelli di controllo flessibili in grado di visualizzare rapidamente contenuti diversi in più widget, a seconda del valore di un campo di input all'interno del pannello di controllo. Ad esempio, puoi creare una dashboard in grado di passare rapidamente tra diverse funzioni Lambda o ID di istanza Amazon EC2, oppure una che può passare a regioni diverse. AWS

Dopo aver creato un pannello di controllo che utilizza una variabile, puoi copiare lo stesso modello di variabile su altri pannelli di controllo esistenti.

L'utilizzo delle variabili del pannello di controllo migliora il flusso di lavoro operativo per le persone che utilizzano i pannelli di controllo. Può anche ridurre i costi perché utilizzi le variabili del pannello di controllo in un unico pannello di controllo invece di crearne altri simili.

> **Note**
>
> Se condividi un pannello di controllo contenente variabili del pannello di controllo, le persone con cui condividi il pannello di controllo non potranno modificare i valori delle variabili.

Tipi di variabili del pannello di controllo

La variabile del pannello di controllo può essere una variabile di proprietà o una variabile di modello.

- Le variabili di proprietà modificano tutte le istanze di una proprietà in tutti i widget del pannello di controllo. Questa proprietà può essere qualsiasi proprietà JSON nell'origine JSON di un pannello di controllo, ad esempio `region`. Oppure può essere il nome di una dimensione per un parametro, ad esempio `InstanceID` o `FunctionName`.

  Per un tutorial che utilizza una variabile di proprietà, consulta Tutorial: Creazione di un pannello di controllo Lambda con il nome della funzione come variabile.

  Per ulteriori informazioni sull'origine JSON dei pannelli di controllo, consulta Struttura e sintassi del corpo del pannello di controllo. Nella CloudWatch console, puoi vedere il codice sorgente JSON per qualsiasi dashboard personalizzato scegliendo Azioni, Visualizza/modifica sorgente.
- Le variabili di modello utilizzano un modello di espressione regolare per modificare tutta una proprietà JSON o solo una parte di essa.

  Per un tutorial che utilizza una variabile di modello, consulta Tutorial: Creazione di un pannello di controllo che utilizzi un modello di espressione regolare per passare da una Regione all'altra.

Le variabili di proprietà si applicano alla maggior parte dei casi d'uso e sono meno complesse da configurare.

Argomenti

- Tutorial: Creazione di un pannello di controllo Lambda con il nome della funzione come variabile
- Tutorial: Creazione di un pannello di controllo che utilizzi un modello di espressione regolare per passare da una Regione all'altra
- Copia di una variabile in un altro pannello di controllo

Tutorial: Creazione di un pannello di controllo Lambda con il nome della funzione come variabile

I passaggi di questa procedura illustrano come creare un pannello di controllo flessibile che mostri una varietà di grafici di parametri, utilizzando una variabile di proprietà. Ciò include una casella di selezione a discesa sul pannello di controllo che può essere utilizzata per cambiare i parametri in tutti i grafici tra diverse funzioni Lambda.

Altri esempi di casi d'uso per questo tipo di pannello di controllo includono l'utilizzo di `InstanceId` come variabile per creare un pannello di controllo di parametri con un menu a discesa per gli ID di esempio. In alternativa, puoi creare un pannello di controllo che utilizzi `region` come variabile per visualizzare lo stesso set di parametri provenienti da Regioni differenti.

Utilizzo di una variabile di proprietà del pannello di controllo per creare un pannello di controllo Lambda flessibile

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Dashboards (Pannelli di controllo), Create dashboard (Crea pannello di controllo).
3. Immetti un nome per il pannello di controllo e scegli Crea pannello di controllo.
4. Aggiungi widget al pannello di controllo che visualizzano i parametri per una funzione Lambda. Quando crei questi widget, specifica Lambda, Per nome di funzione per i parametri del widget. Per la funzione, specifica una delle funzioni Lambda che desideri includere in questo pannello di controllo.

   Per ulteriori informazioni sull'aggiunta di widget a un pannello di controllo, consulta Crea e lavora con i widget nelle dashboard CloudWatch .
5. Dopo aver aggiunto i widget, durante la visualizzazione del pannello di controllo, scegli Operazioni, Variabili, Crea una variabile.
6. Scegli Variabile di proprietà.
7. Per Proprietà modificata dalla variabile, scegliete FunctionName.
8. Per Tipo di input, per questo caso d'uso, consigliamo di scegliere Menu Seleziona (menu a discesa). Questo crea un menu a discesa nel pannello di controllo in cui è possibile selezionare il nome della funzione Lambda per cui visualizzare i parametri.

   Se si trattasse di un pannello di controllo che alterna solo due o tre valori diversi per una variabile, allora Pulsante d'opzione sarebbe una buona scelta.

Se dovessi preferire inserire o incollare i valori per la variabile, dovresti scegliere Input di testo. Questa opzione non include un elenco a discesa o pulsanti di opzione.

9. Quando scegli Menu Seleziona (menu a discesa), devi quindi scegliere se compilare il menu inserendo valori o utilizzando una ricerca di parametri. In questo caso d'uso, supponiamo di avere un gran numero di funzioni Lambda e di non volerle inserire tutte manualmente. Scegli Utilizza i risultati di una ricerca dei parametri, quindi procedi come segue:

   a. Scegli Query predefinite, Lambda, Errori.

      (La scelta degli errori non aggiunge la metrica Errors alla dashboard. Tuttavia, compila rapidamente la casella di selezione delle FunctionNamevariabili.)

   b. Scegli Per nome della funzione, quindi scegli Cerca.

      Sotto il pulsante Cerca, vedrai quindi FunctionNameselezionato. Viene inoltre visualizzato un messaggio che indica quanti valori di FunctionNamedimensione sono stati trovati per compilare la casella di input.

10. (Facoltativo) Per ulteriori impostazioni, scegli Impostazioni secondarie ed effettua una o più delle seguenti operazioni:

    - Per personalizzare il nome della variabile, inserisci il nome in Nome della variabile personalizzata.
    - Per personalizzare l'etichetta per il campo di immissione della variabile, inserisci l'etichetta in Etichetta di input.
    - Per impostare il valore predefinito per questa variabile alla prima apertura del pannello di controllo, inserisci il valore predefinito in Valore predefinito.

11. Scegli Aggiungi variabile.

    Nella parte superiore della dashboard viene visualizzata una casella di selezione a FunctionNamediscesa. È possibile selezionare una funzione Lambda in questa casella e tutti i widget che utilizzano la variabile mostreranno informazioni sulla funzione selezionata.

    Successivamente, se aggiungi altri widget alla dashboard che controllano le metriche Lambda con FunctionNamela dimensione, utilizzeranno automaticamente la variabile.

Tutorial: Creazione di un pannello di controllo che utilizzi un modello di espressione regolare per passare da una Regione all'altra

I passaggi di questa procedura illustrano come creare un pannello di controllo flessibile in grado di passare da una Regione all'altra. Questo tutorial utilizza una variabile di modello di espressione regolare anziché una variabile di proprietà. Per un tutorial che utilizza una variabile di proprietà, consulta Tutorial: Creazione di un pannello di controllo Lambda con il nome della funzione come variabile.

Per numerosi casi d'uso, puoi creare un pannello di controllo che passa da una Regione all'altra utilizzando una variabile di proprietà. Tuttavia, se i widget si basano su nomi della risorsa Amazon (ARN) che includono i nomi delle Regioni, per modificare i nomi delle Regioni all'interno degli ARN è necessario utilizzare una variabile di modello.

Utilizzo di una variabile di modello del pannello di controllo per creare un pannello di controllo flessibile multi-regione

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Dashboards (Pannelli di controllo), Create dashboard (Crea pannello di controllo).
3. Immetti un nome per il pannello di controllo e scegli Crea pannello di controllo.
4. Aggiungi i widget al pannello di controllo. Quando aggiungi i widget per visualizzare dati specifici della Regione, evita di specificare dimensioni con valori che appaiono in una sola Regione. Ad esempio, per i parametri di Amazon EC2, specifica i parametri aggregati anziché i parametri che utilizzano InstanceID come dimensione.

   Per ulteriori informazioni sull'aggiunta di widget a un pannello di controllo, consulta Crea e lavora con i widget nelle dashboard CloudWatch .
5. Dopo aver aggiunto i widget, durante la visualizzazione del pannello di controllo, scegli Operazioni, Variabili, Crea una variabile.
6. Scegli Variabile di modello.
7. Per Proprietà modificata dalla variabile, inserisci il nome della Regione corrente del pannello di controllo, ad esempio **`us-east-2`**.

   Avrai inserito la Regione corretta se l'etichetta sotto quella casella mostra i widget che saranno influenzati dalla variabile.
8. Per Tipo di input, per questo caso d'uso, seleziona Pulsante di opzione.

9. Per Definisci come vengono popolati gli input, scegli Crea un elenco di valori personalizzati.
10. In Crea i tuoi valori personalizzati, inserisci le Regioni da cui desideri passare, con una Regione per riga. Dopo ogni Regione, inserisci una virgola e poi l'etichetta da visualizzare per quel pulsante di opzione. Per esempio:

    ```
    us-east-1, N. Virginia

    us-east-2, Ohio

    eu-west-3, Paris
    ```

    Man mano che inserisci i valori personalizzati, il riquadro Anteprima si aggiorna per mostrare l'aspetto dei pulsanti di opzione.
11. (Facoltativo) Per ulteriori impostazioni, scegli Impostazioni secondarie ed effettua una o più delle seguenti operazioni:

    - Per personalizzare il nome della variabile, inserisci il nome in Nome della variabile personalizzata.
    - Per personalizzare l'etichetta per il campo di immissione della variabile, inserisci l'etichetta in Etichetta di input. Per questo tutorial, digita **`Region:`**.

      Se immetti un valore, il riquadro Anteprima si aggiorna per mostrare l'aspetto dei pulsanti di opzione.
    - Per impostare il valore predefinito per questa variabile alla prima apertura del pannello di controllo, inserisci il valore predefinito in Valore predefinito.
12. Scegli Aggiungi variabile.

    Viene visualizzato il pannello di controllo, con l'etichetta Regione: accanto ai pulsanti di opzione per le Regioni nella parte superiore. Quando passi da una Regione all'altra, tutti i widget che utilizzano la variabile mostreranno informazioni sulla Regione selezionata.

Copia di una variabile in un altro pannello di controllo

Dopo aver creato un pannello di controllo con variabili utili, potrai copiare le variabili in altri pannelli di controllo esistenti. Per ulteriori informazioni sulle variabili del pannello di controllo, consulta Creazione di pannelli di controllo flessibili con variabili del pannello di controllo.

Copia di una variabile di un pannello di controllo in un altro pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Pannelli di controllo, quindi scegli il pannello di controllo che contiene la variabile da copiare. Inserisci una stringa per trovare i pannelli di controllo con nomi corrispondenti, se necessario.
3. Scegli Operazioni, Variabili, Gestisci variabili.
4. Scegli il pulsante di opzione accanto alla variabile che desideri copiare, quindi scegli Copia in un altro pannello di controllo.
5. Scegli la casella di selezione e inizia a digitare il nome del pannello di controllo in cui desideri copiare la variabile.
6. Seleziona il nome del pannello di controllo e scegli Copia variabile.

Crea e lavora con i widget nelle dashboard CloudWatch

Utilizza gli argomenti di questa sezione per creare e utilizzare grafici, avvisi e widget di testo nei pannelli di controllo.

Indice

Aggiungere o rimuovere un grafico da una dashboard CloudWatch

Puoi aggiungere grafici che contengono una o più metriche alla dashboard. CloudWatch I tipi di grafici che è possibile aggiungere al pannello di controllo includono:Line (Linea), Stacked area (Area in pila), Number (Numero), Gauge (Calibro), Bar (Barra) e Pie (Torta). Puoi rimuovere i grafici dal pannello di controllo quando non ne hai più bisogno. Le procedure descritte in questa sezione descrivono come aggiungere e rimuovere grafici dal pannello di controllo. Per informazioni su come modificare un grafico sulla dashboard, consulta Modificare un grafico su una CloudWatch dashboard.

Aggiunta di un grafico a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Scegli il simbolo +, quindi seleziona il grafico che desideri aggiungere al pannello di controllo, poi scegli Avanti.

   - Se selezioni Line (Linea), Stacked area (Area in pila), Bar (Barra) o Pie (Torta), scegli Metrics (Parametri).
4. Nella scheda Sfoglia, cerca o sfoglia i parametri da rappresentare graficamente e seleziona quelli desiderati.
5. (Facoltativo) Per modificare l'intervallo di tempo del grafico, seleziona uno degli intervalli temporali predefiniti in alto nella schermata. Questi intervalli vanno da 1 ora a 1 settimana (1h [1 ora], 3h [3 ore], 12h [12 ore], 1d [1 giorno], 3d [3 giorni] oppure 1w [1 settimana]).

   Per impostare l'intervallo di tempo, scegli Custom (Personalizzato).

   - (Facoltativo) Per fare in modo che questo widget continui a utilizzare l'intervallo di tempo selezionato, anche se l'intervallo di tempo per il resto del pannello di controllo viene successivamente modificato, scegli Rendi permanente l'intervallo di tempo.
6. (Facoltativo) Puoi modificare il tipo di widget del grafico scegliendo il menu a discesa accanto agli intervalli di tempo predefiniti.
7. (Facoltativo) In Graphed metrics (Parametri definiti), puoi aggiungere un'etichetta dinamica al parametro e modificare l'etichetta, il colore dell'etichetta, la statistica e il periodo del parametro. Puoi anche determinare la posizione delle etichette da sinistra a destra sull'asse Y.

a. Per aggiungere un'etichetta dinamica, scegli Graphed metrics (Parametri definiti), quindi Add dynamic labels (Aggiungi etichette dinamiche). Le etichette dinamiche visualizzano una statistica sul parametro nella legenda del grafico. Le etichette dinamiche si aggiornano automaticamente ogni volta che il pannello di controllo o il grafico viene aggiornato. Per impostazione predefinita, i valori dinamici che aggiungi alle etichette vengono visualizzati all'inizio delle etichette. Per ulteriori informazioni, consulta Utilizzo di etichette dinamiche.

b. Per modificare il colore di un parametro, scegli il quadrato colorato accanto ad esso.

c. Per modificare la statistica, seleziona il menu a discesa sotto Statistic (Statistica), quindi scegli un nuovo valore. Per ulteriori informazioni, consulta la sezione Statistiche.

d. Per modificare il periodo, scegli il valore del periodo sotto la colonna Period (Periodo), quindi scegli un nuovo valore.

8. Se stai creando un widget di misurazione, devi scegliere la scheda Opzioni e specificare i valori Min e Max da utilizzare per le due estremità del misuratore.

9. (Facoltativo) Per personalizzare l'asse Y, scegli Options (Opzioni). Puoi aggiungere un'etichetta personalizzata nel campo dell'etichetta nella sezione Left Y-axis (Asse Y sinistro). Se il grafico mostra i valori sull'asse Y destro, puoi personalizzare anche questa etichetta. Puoi anche impostare i limiti minimo e massimo sull'asse Y in modo che il grafico mostri solo l'intervallo di valori specificato.

10. (Facoltativo) Per aggiungere o modificare annotazioni orizzontali su grafici a linee o ad aree impilate o per aggiungere soglie ai widget di misurazione, scegli Opzioni:

a. Per aggiungere un'annotazione orizzontale o una soglia, scegliete Aggiungi annotazione orizzontale o Aggiungi soglia.

b. Per Etichetta, inserisci un'etichetta per l'annotazione, quindi scegli l'icona del segno di spunta.

c. Per Value (Valore), scegli l'icona a forma di carta e penna accanto al valore corrente e inserisci il nuovo valore. Dopo aver inserito il valore, seleziona l'icona del segno di spunta.

d. Per Fill (Riempimento), seleziona il menu a discesa, quindi specifica in che modo l'annotazione utilizzerà l'ombreggiatura. È possibile scegliere Nessuna, Sopra, Tra, oppure Sotto. Per modificare il colore di riempimento, seleziona il quadrato colorato accanto all'annotazione.

e. Per Asse, specificare se l'annotazione viene visualizzata sul lato sinistro o destro dell'asse Y.

f. Per nascondere un'annotazione, deseleziona la casella di controllo accanto all'annotazione da nascondere.

g. Per eliminare un'annotazione, seleziona X nella colonna Actions (Operazioni).

> **Note**
>
> Ripeti questi passaggi per aggiungere più annotazioni orizzontali o soglie allo stesso grafico o indicatore.

11. (Facoltativo) Per aggiungere o modificare le annotazioni verticali, seleziona Options (Opzioni):

    a. Per aggiungere un'annotazione verticale, seleziona Add horizontal annotation (Aggiungi annotazione orizzontale).

    b. Per Label (Etichetta), scegli l'icona a forma di carta e penna accanto all'annotazione corrente e inserisci la nuova annotazione. Per visualizzare solo la data e l'ora, lascia il campo dell'etichetta vuoto.

    c. Per Date (Data), sceglie la data e l'ora correnti, quindi inserisci la nuova data e ora.

    d. Per Fill (Riempimento), seleziona il menu a discesa, quindi specifica in che modo l'annotazione utilizzerà l'ombreggiatura. È possibile scegliere Nessuna, Sopra, Tra, oppure Sotto. Per modificare il colore di riempimento, seleziona il quadrato colorato accanto all'annotazione.

    e. Per nascondere un'annotazione, deseleziona la casella di controllo accanto all'annotazione da nascondere.

    f. Per eliminare un'annotazione, seleziona X nella colonna Actions (Operazioni).

> **Note**
>
> Ripeti questi passaggi per aggiungere più annotazioni verticali allo stesso grafico.

12. Seleziona Crea widget.
13. Seleziona Salva pannello di controllo.

Rimozione di un grafico da un pannello di controllo

1. Apri la CloudWatch console all'[indirizzo https://console.aws.amazon.com/cloudwatch/](https://console.aws.amazon.com/cloudwatch/).

2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Nell'angolo in alto a destra del grafico da modificare, scegli Widget actions (Operazioni del widget), quindi scegli Delete (Elimina).
4. Seleziona Salva pannello di controllo.

Rappresenta graficamente le metriche manualmente su una dashboard CloudWatch

Se una metrica non ha pubblicato dati negli ultimi 14 giorni, non puoi trovarla quando cerchi metriche da aggiungere a un grafico su una dashboard. CloudWatch Usa le fasi seguenti per aggiungere manualmente qualsiasi parametro a un grafico esistente.

Per aggiungere a un grafico un parametro che non è possibile trovare in una ricerca

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Il pannello di controllo deve già contenente un grafico in cui aggiungere il parametro. In caso contrario, creare il grafico e aggiungere ad esso qualsiasi parametro. Per ulteriori informazioni, consulta la pagina Aggiungere o rimuovere un grafico da una dashboard CloudWatch .
4. Seleziona Actions (Operazioni), View/edit source (Visualizza/modifica origine).

   Viene visualizzato un blocco JSON. Il blocco specifica i widget presenti nel pannello di controllo e i relativi contenuti. Di seguito è illustrato un esempio di una parte di questo blocco, che definisce un grafico.

```
{
            "type": "metric",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 3,
            "properties": {
                "view": "singleValue",
                "metrics": [
                    [ "AWS/EBS", "VolumeReadOps", "VolumeId",
 "vol-1234567890abcdef0" ]
```

```
                ],
                "region": "us-west-1"
            }
        },
```

Nell'esempio, la sezione seguente definisce il parametro visualizzato sul grafico.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ]
```

5. Aggiungi una virgola dopo la parentesi quadra chiusa, se non è già presente, e quindi aggiungi una sezione simile tra parentesi quadre dopo la virgola. In questa nuova sezione specifica lo spazio dei nomi, il nome parametro e, se necessario, le dimensioni del parametro da aggiungere al grafico. Di seguito è riportato un esempio.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ],
[ "MyNamespace", "MyMetricName", "DimensionName", "DimensionValue" ]
```

   Per ulteriori informazioni sulla formattazione dei parametri in JSON, consulta la pagina relativa alle proprietà di un oggetto widget di un parametro.
6. Scegli Update (Aggiorna).

Utilizzo di grafici esistenti

Seguire le procedure descritte in queste sezioni per modificare e modificare i widget del grafico del pannello di controllo esistenti.

Argomenti

- Modificare un grafico su una dashboard CloudWatch
- Sposta o ridimensiona un grafico su una dashboard CloudWatch
- Rinomina un grafico su un pannello di controllo CloudWatch

Modificare un grafico su una dashboard CloudWatch

Puoi modificare i grafici che aggiungi alla CloudWatch dashboard. È possibile modificare il titolo, la statistica o il periodo di un grafico. Inoltre, puoi aggiungere, aggiornare e rimuovere parametri dai grafici. Se il grafico contiene più parametri, puoi ridurre l'ingombro nascondendo i parametri non in uso. Le procedure descritte in questa sezione descrivono come modificare un grafico nel pannello di

controllo. Per informazioni sulla creazione di un grafico, consulta Aggiungere o rimuovere un grafico da una CloudWatch dashboard.

New interface

Modifica di un grafico in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Nell'angolo in alto a destra del grafico da modificare, scegli Widget actions (Operazioni del widget), quindi scegli Edit (Modifica).
4. Per modificare il titolo del grafico, scegli l'icona a forma di carta e penna accanto al titolo corrente. Inserisci il nuovo titolo, quindi scegli Applica.
5. (Facoltativo) Per modificare l'intervallo di tempo del grafico, seleziona uno degli intervalli temporali predefiniti nell'area in alto del grafico. Questi intervalli vanno da 1 ora a 1 settimana (1h [1 ora], 3h [3 ore], 12h [12 ore], 1d [1 giorno], 3d [3 giorni] oppure 1w [1 settimana]).

   Per impostare l'intervallo di tempo, scegli Custom (Personalizzato).

   - (Facoltativo) Per fare in modo che questo widget continui a utilizzare l'intervallo di tempo selezionato, anche se l'intervallo di tempo per il resto del pannello di controllo viene successivamente modificato, scegli Rendi permanente l'intervallo di tempo.
6. Per modificare il tipo di widget del grafico, utilizza il menu a discesa accanto agli intervalli di tempo predefiniti.
7. In Graphed metrics (Parametri definiti), puoi aggiungere un'etichetta dinamica al parametro e modificare l'etichetta, il colore dell'etichetta, la statistica e il periodo del parametro. Puoi anche determinare la posizione delle etichette da sinistra a destra sull'asse Y.

   a. Per aggiungere un'etichetta dinamica a un parametro, scegli Dynamic labels (Etichette dinamiche). Le etichette dinamiche visualizzano una statistica sul parametro nella legenda del grafico. Le etichette dinamiche si aggiornano automaticamente ogni volta che il pannello di controllo o il grafico viene aggiornato. Per impostazione predefinita, i valori dinamici che aggiungi alle etichette vengono visualizzati all'inizio delle etichette. Per ulteriori informazioni, consulta Utilizzo di etichette dinamiche.

   b. Per modificare il colore di un parametro, scegli il quadrato colorato accanto ad esso.

c. Per modificare la statistica, scegli il valore statistico sotto la colonna Statistic (Statistica), quindi scegli un nuovo valore. Per ulteriori informazioni, consulta Statistiche.

d. Per modificare il periodo, scegli il valore del periodo sotto la colonna Period (Periodo), quindi scegli un nuovo valore.

8. Per aggiungere o modificare le annotazioni orizzontali, seleziona Opzioni:

a. Per aggiungere un'annotazione orizzontale, seleziona Add horizontal annotation (Aggiungi annotazione orizzontale).

b. Per Etichetta, scegli l'icona con carta e penna accanto all'annotazione corrente. Quindi inserisci la tua nuova annotazione. Dopo aver inserito l'annotazione, seleziona l'icona del segno di spunta.

c. Per Valore, scegli l'icona con carta e penna accanto al valore del parametro corrente. Quindi inserisci il nuovo valore del parametro. Dopo aver inserito il valore, seleziona l'icona del segno di spunta.

d. Per Riempi, scegli il menu a discesa sotto la colonna, quindi specifica in che modo l'annotazione utilizzerà l'ombreggiatura. È possibile scegliere Nessuna, Sopra, Tra, oppure Sotto. Se scegli Tra, viene visualizzato un nuovo campo etichetta e valore.

> **Tip**
>
> Puoi modificare il colore di riempimento selezionando il quadrato colorato accanto all'annotazione.

e. Per Asse, specificare se l'annotazione viene visualizzata sul lato sinistro o destro dell'asse Y.

f. Per nascondere un'annotazione, deseleziona la casella di controllo accanto all'annotazione da nascondere sul grafico.

g. Per cancellare un'annotazione, seleziona X nella colonna Operazioni.

> **Note**
>
> Ripeti questi passaggi per aggiungere più annotazioni orizzontali allo stesso grafico.

9. Per aggiungere o modificare le annotazioni verticali, seleziona Opzioni:

a. Per aggiungere un'annotazione verticale, seleziona Add horizontal annotation (Aggiungi annotazione orizzontale).

b. Per Etichetta, scegli l'icona con carta e penna accanto all'annotazione corrente. Quindi inserisci la tua nuova annotazione. Dopo aver inserito l'annotazione, seleziona l'icona del segno di spunta.

> **Tip**
>
> Per mostrare solo la data e l'ora dell'annotazione, lascia il campo etichetta vuoto.

c. Per Data, scegli la data e l'ora correnti. Quindi inserisci la nuova data e ora.

d. Per Riempi, scegli il menu a discesa sotto la colonna, quindi specifica in che modo l'annotazione utilizzerà l'ombreggiatura. È possibile scegliere Nessuna, Sopra, Tra, oppure Sotto. Se scegli Tra, viene visualizzato un nuovo campo etichetta e valore.

> **Tip**
>
> Puoi modificare il colore di riempimento selezionando il quadrato colorato accanto all'annotazione.

> **Note**
>
> Ripeti questi passaggi per aggiungere più annotazioni verticali allo stesso grafico.

e. Per nascondere un'annotazione, deseleziona la casella di controllo accanto all'annotazione da nascondere sul grafico.

f. Per cancellare un'annotazione, seleziona X nella colonna Operazioni.

10. Per personalizzare l'asse Y, seleziona la scheda Options (Opzioni). In Left Y-axis (Asse Y sinistro), puoi inserire un'etichetta personalizzata per Label (Etichetta). Se il grafico mostra i valori sull'asse Y destro, puoi personalizzare anche questa etichetta. Puoi anche impostare i valori minimo e massimo sull'asse Y in modo che il grafico mostri solo l'intervallo di valori specificato.

11. Una volta completate le modifiche, scegli Aggiorna widget.

Per nascondere o modificare la posizione di una legenda del grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Nell'angolo in alto a destra del grafico da modificare, seleziona Widget actions (Operazioni del widget). Scegli Legend (Legenda), quindi seleziona Hidden (Nascosta), Bottom (In basso) o Right (Destra).

Nascondere temporaneamente i parametri di un grafico in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Seleziona il quadrato colorato del parametro da nascondere nel piè di pagina del grafico. Passando il mouse sul quadrato colorato verrà visualizzata una X, che diventerà grigia quando la selezioni.
4. Per ripristinare il parametro nascosto, deseleziona la X nel quadrato grigio.

Original interface

Modifica di un grafico in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Passa il mouse sull'angolo in alto a destra del grafico che desideri modificare. Scegli Widget actions (Operazioni del widget), quindi scegli Edit (Modifica).
4. Per modificare il titolo del grafico, scegli l'icona a forma di carta e penna accanto al titolo corrente, quindi inserisci il nuovo titolo.
5. Per modificare l'intervallo di tempo del grafico, scegli uno degli intervalli temporali predefiniti nell'area in alto del grafico. Questi intervalli vanno da 1 ora a 1 settimana (1h,3h,12h,1d,3d, oppure 1w).

    - Per impostare l'intervallo temporale, scegli Custom (Personalizzato).

6. Per modificare il tipo di widget del grafico, seleziona la scheda Opzioni del grafico. È possibile scegliere Linea, Area impilata, Numero, Barra o Torta.

   > **Tip**
   >
   > Puoi modificare il tipo di widget del grafico scegliendo il menu a discesa accanto agli intervalli di tempo predefiniti.

7. In Graphed metrics (Parametri definiti), puoi aggiungere un'etichetta dinamica al parametro e modificare l'etichetta, il colore dell'etichetta, la statistica e il periodo del parametro. Puoi anche determinare la posizione delle etichette da sinistra a destra sull'asse Y.

   a. Per aggiungere un'etichetta dinamica a un parametro, scegli Dynamic labels (Etichette dinamiche). Le etichette dinamiche visualizzano una statistica sul parametro nella legenda del grafico. Le etichette dinamiche si aggiornano automaticamente ogni volta che il pannello di controllo o il grafico viene aggiornato. Per impostazione predefinita, i valori dinamici che aggiungi alle etichette vengono visualizzati all'inizio delle etichette. Per ulteriori informazioni, consulta Utilizzo di etichette dinamiche.

   b. Per modificare il colore di un parametro, scegli il quadrato colorato accanto ad esso.

   c. Per modificare la statistica, scegli il valore statistico sotto la colonna Statistic (Statistica), quindi scegli un nuovo valore. Per ulteriori informazioni, consulta Statistiche.

   d. Per modificare il periodo, scegli il valore del periodo sotto la colonna Period (Periodo), quindi scegli un nuovo valore.

8. Per aggiungere o modificare le annotazioni orizzontali, seleziona Graph options (Opzioni grafici):

   a. Per aggiungere un'annotazione orizzontale, seleziona Add horizontal annotation (Aggiungi annotazione orizzontale).

   b. Per Etichetta, scegli l'icona a forma di matita accanto all'annotazione corrente. Quindi inserisci la tua nuova annotazione. Dopo aver inserito l'annotazione, seleziona l'icona del segno di spunta.

   c. Per Valore, scegli l'icona a forma di matita accanto al valore della parametro corrente. Quindi inserisci il nuovo valore del parametro. Dopo aver inserito il valore, seleziona l'icona del segno di spunta.

d. Per Riempi, scegli il menu a discesa sotto la colonna, quindi specifica in che modo l'annotazione utilizzerà l'ombreggiatura. È possibile scegliere Nessuna, Sopra, Tra, oppure Sotto. Se scegli Tra, viene visualizzato un nuovo campo etichetta e valore.

> **Tip**
>
> Puoi modificare il colore di riempimento selezionando il quadrato colorato accanto all'annotazione.

e. Per Asse, specificare se l'annotazione viene visualizzata sul lato sinistro o destro dell'asse Y.

f. Per nascondere un'annotazione, deseleziona la casella di controllo accanto all'annotazione da nascondere sul grafico.

g. Per cancellare un'annotazione, seleziona X nella colonna Operazioni.

> **Note**
>
> Ripeti questi passaggi per aggiungere più annotazioni orizzontali allo stesso grafico.

9. Per aggiungere o modificare le annotazioni verticali, seleziona Opzioni grafici:

a. Per aggiungere un'annotazione verticale, seleziona Add horizontal annotation (Aggiungi annotazione orizzontale).

b. Per Etichetta, scegli l'icona a forma di matita accanto all'annotazione corrente. Quindi inserisci la tua nuova annotazione. Dopo aver inserito l'annotazione, seleziona l'icona del segno di spunta.

> **Tip**
>
> Per mostrare solo la data e l'ora dell'annotazione, lascia il campo etichetta vuoto.

c. Per Data, scegli l'icona a forma di matita accanto alla data e ora correnti. Quindi inserisci la nuova data e ora.

d. Per Riempi, scegli il menu a discesa sotto la colonna, quindi specifica in che modo l'annotazione utilizzerà l'ombreggiatura. È possibile scegliere Nessuna, Sopra, Tra, oppure Sotto. Se scegli Tra, viene visualizzato un nuovo campo etichetta e valore.

> **Tip**
>
> Puoi modificare il colore di riempimento selezionando il quadrato colorato accanto all'annotazione.

> **Note**
>
> Ripeti questi passaggi per aggiungere più annotazioni verticali allo stesso grafico.

e. Per nascondere un'annotazione, deseleziona la casella di controllo accanto all'annotazione da nascondere sul grafico.

f. Per cancellare un'annotazione, seleziona X nella colonna Operazioni.

10. Per personalizzare l'asse Y, scegli Graph options (Opzioni grafici). In Left Y-axis (Asse Y sinistro), puoi inserire un'etichetta personalizzata per Label (Etichetta). Se il grafico mostra i valori sull'asse Y destro, puoi personalizzare anche questa etichetta. Puoi anche impostare i valori minimo e massimo sull'asse Y in modo che il grafico mostri solo l'intervallo di valori specificato.
11. Una volta completate le modifiche, scegli Aggiorna widget.

Per nascondere o modificare la posizione di una legenda del grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Passa il mouse sull'angolo in alto a destra del grafico che desideri modificare, quindi scegli Widget actions (Operazioni del widget). Scegli Legend (Legenda), quindi seleziona Hidden (Nascosta), Bottom (In basso) o Right (Destra).

Nascondere temporaneamente i parametri di un grafico in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.

3. Seleziona il quadrato colorato del parametro da nascondere nel piè di pagina del grafico. Passando il mouse sul quadrato colorato verrà visualizzata una X, che diventerà grigia quando la selezioni.
4. Per ripristinare il parametro nascosto, deseleziona la X nel quadrato grigio.

Sposta o ridimensiona un grafico su una dashboard CloudWatch

Puoi disporre e ridimensionare i grafici sulla dashboard. CloudWatch

Spostamento di un grafico in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Esegui una di queste operazioni:
   - Passa il puntatore del mouse sul titolo del grafico fino a visualizzare l'icona di selezione. Seleziona il grafico e trascinalo in una nuova posizione nel pannello di controllo.
   - Per spostare il widget nell'area in alto a sinistra o in basso a sinistra del pannello di controllo, scegli i puntini sospensivi verticali in alto a destra nel widget e apri il menu Operazioni widget. Quindi scegli Sposta e scegli dove spostare il widget.
4. Seleziona Salva pannello di controllo.

Ridimensionamento di un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Per aumentare o diminuire le dimensioni, passa con il mouse sopra il grafico e trascina l'angolo inferiore destro del grafico. Interrompi il trascinamento dell'angolo quando hai la dimensione desiderata.
4. Seleziona Salva pannello di controllo.

Ingrandimento temporaneo di un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Seleziona il grafico oppure passa con il mouse sul titolo del grafico e seleziona Widget actions (Operazioni widget), Enlarge (Ingrandisci).

Rinomina un grafico su un pannello di controllo CloudWatch

Puoi modificare il nome predefinito CloudWatch assegnato a un grafico sulla dashboard.

Rinominare un grafico in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Passa con il mouse sul titolo del grafico e seleziona Widget actions (Operazioni widget), Edit (Modifica).
4. Nella schermata in alto Edit graph (Modifica grafico), seleziona il titolo del grafico.
5. In Title (Titolo), immetti un nuovo nome e scegli Ok (segno di spunta). In basso a destra nella schermata Edit graph (Modifica grafico), scegli Update widget (Aggiorna widget).

Aggiungi un widget Metrics Explorer a una dashboard CloudWatch

I widget di esplorazione dei parametri includono grafici di più risorse che hanno lo stesso tag o condividono la stessa proprietà della risorsa, ad esempio un tipo di istanza. Questi widget rimangono aggiornati, poiché le risorse corrispondenti vengono create o eliminate. L'aggiunta di widget di esplorazione dei parametri al pannello di controllo consente di risolvere i problemi dell'ambiente in modo più efficiente.

Ad esempio, è possibile monitorare il parco istanze EC2 assegnando tag che rappresentano i rispettivi ambienti, ad esempio produzione o test. È quindi possibile utilizzare questi tag per filtrare e aggregare i parametri operativi, ad esempio `CPUUtilization`, per comprendere l'integrità e le prestazioni delle istanze EC2 associate a ciascun tag.

La procedura seguente illustra come aggiungere un widget di esplorazione dei parametri a un pannello di controllo utilizzando la console. Puoi anche aggiungerlo a livello di codice o utilizzando. AWS CloudFormation Per ulteriori informazioni, consulta Metrics Explorer Widget Object Definition and. AWS::CloudWatch::Dashboard

Per aggiungere un widget di esplorazione dei parametri a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Seleziona il nome del pannello di controllo in cui desideri aggiungere il widget di esplorazione dei parametri.
4. Scegli il simbolo +.
5. Scegli Explorer e poi scegli Next (Avanti).

> **Note**
>
> Per poter aggiungere un widget di esplorazione dei parametri, è necessario accedere alla nuova visualizzazione del pannello di controllo. Per acconsentire, scegli Dashboards (Pannelli di controllo) nel pannello di navigazione, quindi scegli try out the new interface (prova la nuova interfaccia) nel banner nella parte superiore della pagina.

6. Esegui una di queste operazioni:
   - Per utilizzare un modello, sceglie Pre-filled Explorer widget (Widget di esplorazione precompilato) e seleziona un modello da utilizzare.
   - Per creare una visualizzazione personalizzata, scegli Empty Explorer widget (Svuota widget di esplorazione).
7. Scegli Create (Crea).

   Se è stato utilizzato un modello, il widget viene visualizzato nel pannello di controllo con i parametri selezionati. Se il widget di esplorazione e il pannello di controllo ti soddisfano, scegli Save dashboard (Salva pannello di controllo).

   Se non hai utilizzato un modello, procedi come segue.
8. Nel nuovo widget sotto Explorer, nella casella Metrics (Parametri), scegli un singolo parametro o tutti i parametri disponibili da un servizio.

   Dopo aver scelto un parametro, puoi ripetere facoltativamente questo passaggio per aggiungere ulteriori parametri.
9. Per ogni metrica selezionata, CloudWatch visualizza la statistica che verrà utilizzata immediatamente dopo il nome della metrica. Per modificare il parametro, scegli il nome della statistica, quindi scegli il parametro desiderato.

10. In From (Da), scegli un tag o una proprietà della risorsa per filtrare i risultati.

    Dopo aver eseguito questa operazione, facoltativamente puoi ripetere questo passaggio per scegliere più tag o proprietà delle risorse.

    Se scegli più valori della stessa proprietà, ad esempio due tipi di istanza EC2, verranno visualizzate tutte le risorse che corrispondono a una delle proprietà selezionate. Questa operazione viene trattata come operazione OR.

    Se si scelgono proprietà o tag diversi, ad esempio il tag **`Production`** e il tipo di istanza M5, vengono visualizzate solo le risorse che corrispondono a tutte queste selezioni. Questa operazione viene trattata come operazione AND.

11. (Facoltativo) In Aggregate by (Aggrega per), scegli una statistica da utilizzare per aggregare i parametri. Quindi, accanto a for (per), scegli come aggregare il parametro dall'elenco. Puoi aggregare tutte le risorse attualmente visualizzate oppure aggregare in base a un singolo tag o proprietà della risorsa.

    A seconda di come scegli di aggregare, il risultato potrebbe essere una singola serie temporale o più serie temporali.

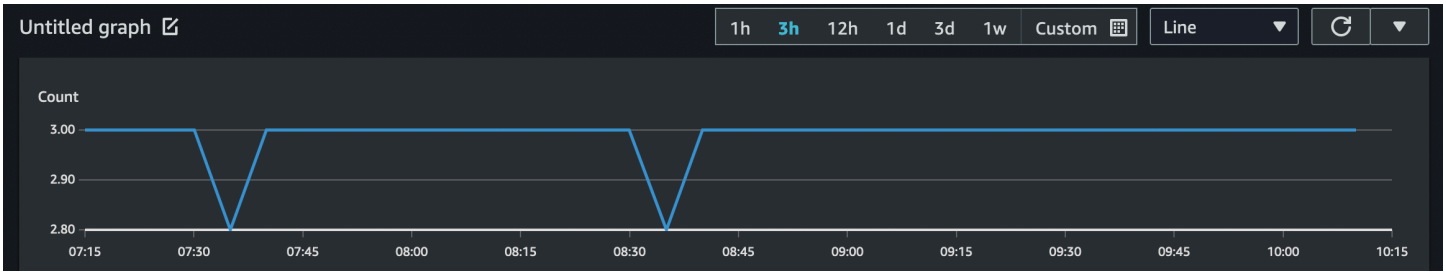
12. In Split by (Dividi per), puoi scegliere di dividere un singolo grafico con più serie temporali in più grafici. La divisione può essere fatta in base a una varietà di criteri che puoi scegliere in Split by (Dividi per).

13. In Graph options (Opzioni del grafico), puoi perfezionare il grafico modificando il periodo, il tipo di grafico, il posizionamento della legenda e il layout.

14. Se il widget di esplorazione e il pannello di controllo ti soddisfano, scegli Save dashboard (Salva pannello di controllo).

Aggiungi o rimuovi un widget di linea su una dashboard CloudWatch

Con il widget linea, puoi confrontare i parametri nel corso di diversi periodi di tempo. Inoltre, puoi utilizzare la funzione di zoom mini-mappa del widget per ispezionare sezioni di grafici lineari senza passare tra viste ingrandite e ridimensionate. Le procedure in questa sezione descrivono come aggiungere e rimuovere un widget di linea su una CloudWatch dashboard. Per informazioni sull'utilizzo della funzione di ingrandimento mini-mappa del widget con grafici lineari, consulta la sezione Ingrandimento di un grafico lineare o aree in pila.



Per aggiungere un widget linea a un pannello di controllo

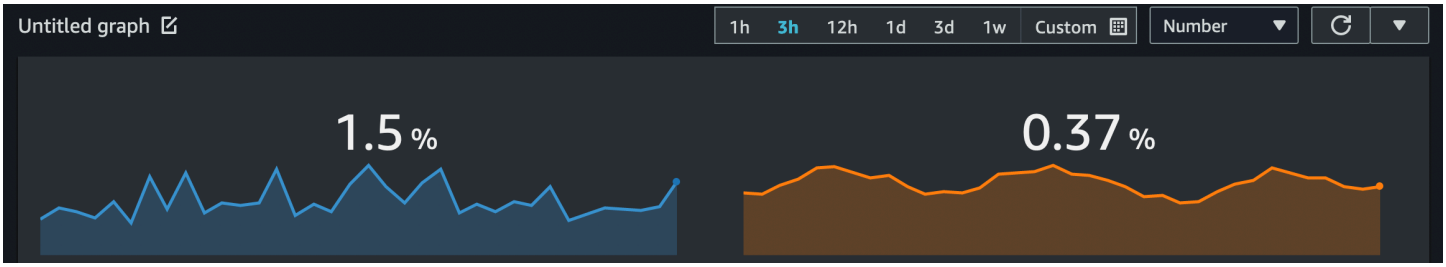
1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Scegli il simbolo + e seleziona Linea.
4. Seleziona Parametri.
5. Scegli Browse (Sfoglia), quindi seleziona il parametro di cui desideri tracciare il grafico.
6. Scegli Create widget (Crea widget), quindi scegli Save dashboard (Salva pannello di controllo).

Per rimuovere un widget linea da un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Nell'angolo in alto a destra del widget linea da rimuovere, scegli Widget actions (Operazioni del widget), quindi scegli Delete (Elimina).
4. Seleziona Salva pannello di controllo.

Aggiungere o rimuovere un widget numerico da una CloudWatch dashboard

Con il widget numerico, puoi esaminare i valori e le tendenze più recenti dei parametri non appena appaiono. Poiché il widget numerico include la funzione sparkline, puoi visualizzare le metà superiore e inferiore delle tendenze del parametro in un singolo grafico. Le procedure in questa sezione descrivono come aggiungere e rimuovere un widget numerico da una CloudWatch dashboard.



> **Note**
>
> Solo la nuova interfaccia supporta la funzione sparkline. Quando crei un widget numerico, la funzione sparkline viene inclusa automaticamente.

Aggiunta di un widget numerico a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Scegli il simbolo + e seleziona Numero.
4. Nella scheda Sfoglia, cerca o sfoglia il parametro da visualizzare.
5. (Facoltativo) Per modificare il colore della funzione sparkline, scegli Graphed metrics (Parametri definiti) e seleziona il riquadro del colore accanto all'etichetta del parametro. Viene visualizzato un menu in cui è possibile scegliere un colore diverso o inserire un codice colore esadecimale a sei cifre per specificare un colore.
6. (Facoltativo) Per disattivare la funzione sparkline, scegli Options (Opzioni). In Sparkline, seleziona la casella di controllo.
7. (Facoltativo) Per modificare l'intervallo di tempo del widget di numeri, seleziona uno degli intervalli temporali predefiniti nell'area in alto nel widget. Questi intervalli vanno da 1 ora a 1 settimana (1h [1 ora], 3h [3 ore], 12h [12 ore], 1d [1 giorno], 3d [3 giorni] oppure 1w [1 settimana]).

   Per impostare l'intervallo di tempo, scegli Custom (Personalizzato).

   - (Facoltativo) Per fare in modo che questo widget continui a utilizzare l'intervallo di tempo selezionato, anche se l'intervallo di tempo per il resto del pannello di controllo viene successivamente modificato, scegli Rendi permanente l'intervallo di tempo.

8. (Facoltativo) Per fare in modo che il widget numerico mostri un aggregato (1h, 3h, 12h, 1d, 3d o 1w).

   Per impostare l'intervallo di tempo, scegli Personalizzato.

   - (Facoltativo) Per fare in modo che questo widget mostri una media del valore metrico sull'intero intervallo di tempo, anziché il valore più recente, scegli Opzioni, Il valore dell'intervallo di tempo mostra il valore dell'intero intervallo di tempo.
9. Scegli Create widget (Crea widget), quindi scegli Save dashboard (Salva pannello di controllo).

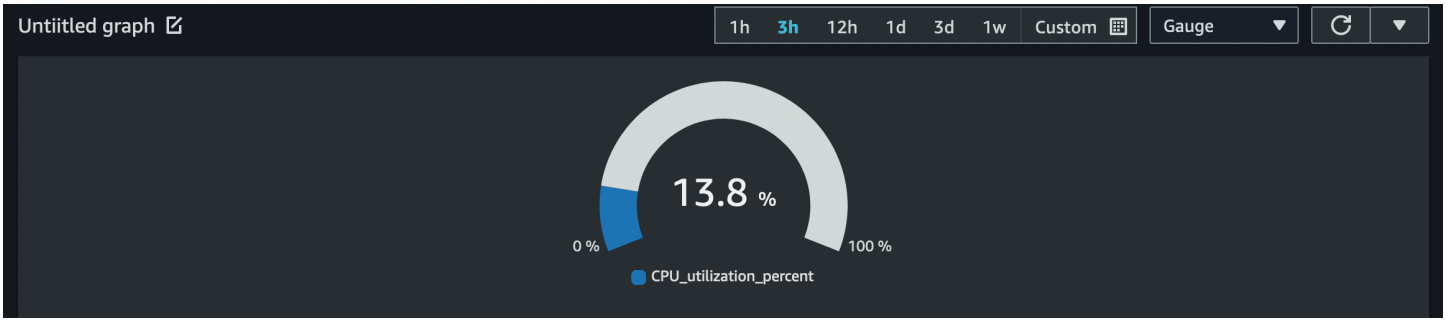
> **Tip**
>
> Puoi disattivare la funzione sparkline dal widget numerico sullo schermo del pannello di controllo. Nell'angolo in alto a destra del widget numerico da modificare, scegli Widget actions (Operazioni del widget). Seleziona Sparkline, quindi scegli Hide sparkline (Nascondi sparkline).

Rimozione di un widget numerico da un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli il pannello di controllo che contiene il widget numerico da eliminare.
3. Nell'angolo in alto a destra del widget numerico da rimuovere, scegli Widget actions (Operazioni del widget), quindi scegli Delete (Elimina).
4. Seleziona Salva pannello di controllo.

Aggiungi o rimuovi un widget di misurazione da una dashboard CloudWatch

Con il widget calibro, è possibile visualizzare i valori dei parametri che rientrano negli intervalli. Ad esempio, è possibile utilizzare il widget calibro per rappresentare graficamente le percentuali e l'utilizzo della CPU, in modo da monitorare e diagnosticare gli eventuali problemi di prestazioni. Le procedure in questa sezione descrivono come aggiungere e rimuovere un widget di misurazione da una CloudWatch dashboard.



> **Note**
>
> Solo la nuova interfaccia nella CloudWatch console supporta la creazione del widget Gauge. Quando crei questo widget, devi impostare un intervallo per il calibro.

Per aggiungere un widget calibro a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Dalla schermata del pannello di controllo, scegli il simbolo +, quindi seleziona Calibro.
4. Scegli Browse (Sfoglia), quindi seleziona il parametro di cui desideri tracciare il grafico.
5. Scegli Opzioni. In Gauge range (Intervallo calibro), imposta i valori per Min (minimo) e Max (massimo). Per le percentuali, come l'utilizzo della CPU, ti consigliamo di impostare i valori di `Min` su `0` e di `Max` su `100`.
6. (Facoltativo) Per modificare il colore del widget calibro, scegli Graphed metrics (Parametri definiti) e seleziona il riquadro del colore accanto all'etichetta del parametro. Viene visualizzato un menu in cui è possibile scegliere un colore diverso o inserire un codice colore esadecimale a sei cifre per specificare un colore.
7. Scegli Create widget (Crea widget), quindi scegli Save dashboard (Salva pannello di controllo).

Per rimuovere un widget calibro da un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli il pannello di controllo che contiene il widget calibro da eliminare.

3. Nell'angolo in alto a destra del widget calibro da eliminare, scegli Widget actions (Operazioni del widget), quindi scegli Delete (Elimina).
4. Seleziona Salva pannello di controllo.

Aggiungi un widget personalizzato a una CloudWatch dashboard

Un widget personalizzato è un widget di CloudWatch dashboard che può richiamare qualsiasi AWS Lambda funzione con parametri personalizzati. Viene quindi visualizzato il codice HTML o JSON restituito. I widget personalizzati sono un modo semplice per creare una visualizzazione dati personalizzata in un pannello di controllo. Se sai scrivere un codice Lambda e creare HTML, puoi creare un widget personalizzato utile. Inoltre, Amazon offre diversi widget personalizzati predefiniti che puoi creare senza alcun codice.

Quando crei una funzione Lambda da utilizzare come widget personalizzato, ti consigliamo vivamente di includere il prefisso customWidget nel nome della funzione. Questo ti aiuta a sapere quali funzioni Lambda sono sicure da usare quando aggiungi widget personalizzati al tuo pannello di controllo.

I widget personalizzati si comportano come altri widget nel pannello di controllo. Possono essere aggiornati e aggiornati automaticamente, ridimensionati e spostati. Reagiscono all'intervallo di tempo del pannello di controllo.

Se hai configurato la funzionalità multiaccount CloudWatch della console, puoi aggiungere un widget personalizzato creato in un account alle dashboard degli altri account. Per ulteriori informazioni, consulta Console per più account e più regioni CloudWatch .

Puoi anche utilizzare widget personalizzati sul tuo sito Web utilizzando la funzione di condivisione della CloudWatch dashboard. Per ulteriori informazioni, consulta CloudWatch Dashboard di condivisione.

Argomenti

- Dettagli sui widget personalizzati
- Sicurezza e JavaScript
- Interattività nel widget personalizzato
- Creazione di un widget personalizzato
- Esempio di widget personalizzati

Dettagli sui widget personalizzati

I widget personalizzati funzionano nel seguente modo:

1. La CloudWatch dashboard richiama la funzione Lambda contenente il codice del widget. Passa in tutti i parametri personalizzati definiti nel widget.
2. La funzione Lambda restituisce una stringa di HTML, JSON o Markdown. Il markdown viene restituito come JSON nel formato seguente:

   ```
   {"markdown":"markdown content"}
   ```

3. Il pannello di controllo visualizza l'HTML o JSON restituito.

Se la funzione restituisce HTML, la maggior parte dei tag HTML sono supportati. Puoi utilizzare gli stili CSS (Cascading Style Sheets) e SVG (Scalable Vector Graphics) per creare viste complesse.

Lo stile predefinito degli elementi HTML come link e tabelle segue lo stile dei CloudWatch dashboard. Puoi personalizzare questo stile utilizzando gli stili in linea, utilizzando il tag `<style>`. Puoi anche disattivare gli stili predefiniti includendo un singolo elemento HTML con la classe di `cwdb-no-default-styles`. Nell'esempio seguente vengono disattivati gli stili predefiniti: `<div class="cwdb-no-default-styles"></div>`.

Ogni chiamata da un widget personalizzato a Lambda include un elemento `widgetContext` con i seguenti contenuti, per fornire informazioni utili sul contesto allo sviluppatore della funzione Lambda.

```
{
    "widgetContext": {
        "dashboardName": "Name-of-current-dashboard",
        "widgetId": "widget-16",
        "accountId": "012345678901",
        "locale": "en",
        "timezone": {
            "label": "UTC",
            "offsetISO": "+00:00",
            "offsetInMinutes": 0
        },
        "period": 300,
        "isAutoPeriod": true,
        "timeRange": {
            "mode": "relative",
```

```
            "start": 1627236199729,
            "end": 1627322599729,
            "relativeStart": 86400012,
            "zoom": {
                "start": 1627276030434,
                "end": 1627282956521
             }
        },
        "theme": "light",
        "linkCharts": true,
        "title": "Tweets for Amazon website problem",
        "forms": {
            "all": {}
        },
        "params": {
            "original": "param-to-widget"
        },
        "width": 588,
        "height": 369
    }
}
```

Stile CSS predefinito

I widget personalizzati forniscono i seguenti elementi stilistici CSS predefiniti:

- È possibile utilizzare la classe CSS btn per aggiungere un pulsante. Trasforma un'ancora (<a>) in un pulsante come nell'esempio seguente:

  ```
  <a class="btn" href=https://amazon.com”>Open Amazon</a>
  ```

- Puoi utilizzare la classe CSS btn btn-primary per aggiungere un pulsante primario.
- Per impostazione predefinita, i seguenti elementi hanno lo stile: table (tabella), select (selezione), headers (h1, h2, and h3) (intestazioni (h1, h2 e h3)), preformatted text (pre) testo preformattato (pre)), input (inserimento) e text area (area di testo).

Utilizzo del parametro describe

Al termine, ti consigliamo di supportare il parametro describe nelle tue funzioni, anche se restituisce solo una stringa vuota. Se non lo supporti e viene chiamato nel widget personalizzato, visualizza il contenuto del widget come se fosse documentazione.

Se includi l'opzione describe, la funzione Lambda restituisce la documentazione in formato Markdown e non fa altro.

Quando crei un widget personalizzato nella console, dopo aver selezionato la funzione Lambda viene visualizzato un pulsante Get documentation (Ottieni documentazione). Se scegli questo pulsante, la funzione viene richiamata con il parametro describe e viene restituita la documentazione della funzione. Se la documentazione è ben formattata in markdown, CloudWatch analizza la prima voce della documentazione circondata da tre caratteri singoli (```) in YAML. Quindi, popola automaticamente la documentazione nei parametri. Di seguito è riportato un esempio di documentazione ben formattata.

```
``` yaml
echo: <h1>Hello world</h1>
```
```

Sicurezza e JavaScript

Per motivi di sicurezza, non JavaScript è consentito nel codice HTML restituito. La rimozione dell' JavaScript opzione previene i problemi di escalation delle autorizzazioni, in cui l'autore della funzione Lambda inietta codice che potrebbe essere eseguito con autorizzazioni più elevate rispetto all'utente che visualizza il widget sulla dashboard.

Se l'HTML restituito contiene JavaScript codice o altre vulnerabilità di sicurezza note, viene rimosso dall'HTML prima di essere visualizzato sulla dashboard. Ad esempio, i tag <iframe> e <use> non sono consentiti e vengono rimossi.

I widget personalizzati non vengono eseguiti per impostazione predefinita in un pannello di controllo. Invece, è necessario consentire esplicitamente l'esecuzione di un widget personalizzato se si considera attendibile la funzione Lambda che richiama. È possibile scegliere di consentirla una volta o consentire sempre, sia per i singoli widget che per l'intero pannello di controllo. È inoltre possibile negare l'autorizzazione per singoli widget e l'intero pannello di controllo.

Interattività nel widget personalizzato

Anche se non JavaScript è consentito, esistono altri modi per consentire l'interattività con l'HTML restituito.

- A qualsiasi elemento nell'HTML restituito può essere assegnato un tag con una configurazione speciale in un tag `<cwdb-action>`, che può visualizzare le informazioni nei popup, chiedere

conferma sui clic e chiamare qualsiasi funzione Lambda quando viene scelto quell'elemento. Ad esempio, puoi definire pulsanti che chiamano qualsiasi AWS API utilizzando una funzione Lambda. L'HTML restituito può essere impostato in modo da sostituire il contenuto del widget Lambda esistente o da essere visualizzato all'interno di un elemento modale.

- Il codice HTML restituito può includere collegamenti che aprono nuove console, aprono altre pagine dei clienti o caricano altri pannelli di controllo.
- L'HTML può includere l'attributo `title` per un elemento, che offre informazioni aggiuntive se l'utente passa con il mouse su tale elemento.
- L'elemento può contenere selettori CSS, ad esempio `:hover`, che può richiamare animazioni o altri effetti CSS. Inoltre, puoi visualizzare o nascondere elementi nella pagina.

Definizione e utilizzo di <cwdb-action>

L'elemento `<cwdb-action>` definisce un comportamento sull'elemento immediatamente precedente. Il contenuto di `<cwdb-action>` è HTML da visualizzare o un blocco JSON di parametri da passare a una funzione Lambda.

Di seguito è illustrato un esempio di un elemento `<cwdb-action>`:

```
<cwdb-action
     action="call|html"
     confirmation="message"
     display="popup|widget"
     endpoint="<lambda ARN>"
     event="click|dblclick|mouseenter">

     html | params in JSON
</cwdb-action>
```

- action (operazione): i valori validi sono `call`, che chiama una funzione Lambda, e `html`, che visualizza qualsiasi codice HTML contenuto in `<cwdb-action>`. Il valore predefinito è `html`.
- confirmation (conferma): visualizza un messaggio di conferma che deve essere accettato prima di iniziare l'operazione, consentendo al cliente di annullarla.
- display (visualizza): i valori validi sono popup e `widget`, che sostituisce il contenuto del widget stesso. Il valore predefinito è `widget`.
- endpoint: Amazon Resource Name (ARN) della funzione Lambda da chiamare. Questo è necessario se `action` è `call`.

- event (evento): definisce l'evento sull'elemento precedente che richiama l'azione. I valori validi sono `click`, `dblclick` e `mouseenter`. È possibile utilizzare l'evento `mouseenter` solo in combinazione con l'operazione `html`. Il valore predefinito è `click`.

Examples (Esempi)

Di seguito è riportato un esempio di come utilizzare il tag `<cwdb-action>` per creare un pulsante che riavvia un'istanza Amazon EC2 utilizzando una chiamata di funzione Lambda. Visualizza l'esito della chiamata in un pop-up.

```
<a class="btn">Reboot Instance</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:rebootInstance" display="popup">
       { "instanceId": "i-342389adbfef" }
</cwdb-action>
```

Nell'esempio seguente vengono visualizzate ulteriori informazioni in un pop-up.

```
<a>Click me for more info in popup</a>
<cwdb-action display="popup">
   <h1>Big title</h1>
   More info about <b>something important</b>.
</cwdb-action>
```

Questo esempio è un pulsante Next (Successivo) che sostituisce il contenuto di un widget con una chiamata a una funzione Lambda.

```
<a class="btn btn-primary">Next</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:nextPage">
   { "pageNum": 2 }
</cwdb-action>
```

Creazione di un widget personalizzato

Per creare un widget personalizzato, puoi utilizzare uno degli esempi forniti da AWS oppure puoi crearne uno personalizzato. Gli AWS esempi includono esempi sia JavaScript in Python che in Python e vengono creati da uno AWS CloudFormation stack. Per un elenco di esempi, consulta la pagina Esempio di widget personalizzati.

Per creare un widget personalizzato in una dashboard CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Scegli il simbolo +.
4. Scegli Custom widget (Widget personalizzato).
5. Seleziona uno dei seguenti metodi:
   - Per utilizzare un widget personalizzato di esempio fornito da AWS, procedi come segue:
     a. Seleziona l'esempio nella casella a discesa.

        La AWS CloudFormation console viene avviata in un nuovo browser. Nella AWS CloudFormation console, procedi come segue:
     b. (Facoltativo) Personalizzate il nome AWS CloudFormation dello stack.
     c. Effettua le selezioni per i parametri utilizzati dall'esempio.
     d. Seleziona Riconosco che AWS CloudFormation potrebbe creare risorse IAM e scegli Create stack.
   - Per creare il tuo widget personalizzato fornito da AWS, procedi come segue:
     a. Seleziona Successivo.
     b. Scegli se selezionare la funzione Lambda da un elenco o inserisci il relativo Amazon Resource Name (ARN). Se lo selezioni da un elenco, specifica anche la regione in cui si trova la funzione e la versione da utilizzare.
     c. Per Parameters (Parametri), effettua le selezioni per qualsiasi parametro utilizzato dalla funzione.
     d. Inserisci un titolo per il widget.
     e. Per Update on (Aggiorna il), configura quando il widget deve essere aggiornato (quando la funzione Lambda deve essere richiamata di nuovo). Può scegliere una o più delle seguenti opzioni: Refresh (Aggiorna) per aggiornarlo quando il pannello di controllo si aggiorna automaticamente, Resize (Ridimensiona) per aggiornarlo ogni volta che il widget viene ridimensionato, o Time Range (Intervallo di tempo) per aggiornarlo ogni volta che l'intervallo di tempo del pannello di controllo viene regolato, incluso quando i grafici vengono ingranditi.
     f. Se l'anteprima ti soddisfa, scegli Create widget (Crea widget).

Esempio di widget personalizzati

AWS fornisce esempi di widget personalizzati sia in Python che JavaScript in Python. Puoi creare questi widget di esempio utilizzando il link per ciascun widget in questo elenco. In alternativa, puoi creare e personalizzare un widget utilizzando la CloudWatch console. I collegamenti in questo elenco aprono una AWS CloudFormation console e utilizzano un collegamento di AWS CloudFormation creazione rapida per creare il widget personalizzato.

Puoi anche accedere agli esempi di widget personalizzati su. GitHub

Dopo questo elenco, vengono mostrati esempi completi del widget Echo per ogni lingua.

JavaScript

Esempi di widget personalizzati in JavaScript

- Echo: un eco di base che puoi utilizzare per verificare come viene visualizzato HTML in un widget personalizzato, senza dover scrivere un nuovo widget.
- Hello World: un widget di avvio molto semplice.
- Custom widget debugger (Debugger del widget personalizzato): un widget di debug che visualizza informazioni utili sull'ambiente runtime Lambda.
- Query CloudWatch Logs Insights: esegue e modifica le query di CloudWatch Logs Insights.
- Run Amazon Athena queries (Esegui query di Amazon Athena): consente di eseguire e modificare le query Athena.
- Call AWS API: richiama qualsiasi API di sola lettura e visualizza i AWS risultati in formato JSON.
- Grafico CloudWatch bitmap veloce: esegui il rendering CloudWatch dei grafici sul lato server, per una visualizzazione rapida.
- Widget di testo dal CloudWatch pannello di controllo: visualizza il primo widget di testo dal pannello di controllo specificato CloudWatch .
- CloudWatch dati metrici come tabella: visualizza i dati CloudWatch metrici non elaborati in una tabella.
- Amazon EC2 table (Tabella Amazon EC2): visualizza le principali istanze EC2 in base all'utilizzo della CPU. Questo widget include anche un pulsante Reebot (Riavvia), che è disabilitato per impostazione predefinita.
- AWS CodeDeploy distribuzioni: visualizza le distribuzioni. CodeDeploy

- AWS Cost Explorer rapporto: visualizza un rapporto sul costo di ogni AWS servizio per un intervallo di tempo selezionato.
- Display content of external URL (Visualizza il contenuto dell'URL esterno): visualizza il contenuto di un URL accessibile esternamente.
- Display an Amazon S3 object (Visualizza un oggetto Amazon S3): visualizza un oggetto in un bucket Amazon S3 nel tuo account.
- Simple SVG pie chart (Semplice grafico a torta SVG): esempio di widget grafico basato su SVG.

Python

Esempio di widget personalizzati in Python

- Echo: un eco di base che puoi utilizzare per verificare come viene visualizzato HTML in un widget personalizzato, senza dover scrivere un nuovo widget.
- Hello World: un widget di avvio molto semplice.
- Custom widget debugger (Debugger del widget personalizzato): un widget di debug che visualizza informazioni utili sull'ambiente runtime Lambda.
- Call AWS API: richiama qualsiasi AWS API di sola lettura e visualizza i risultati in formato JSON.
- Grafico CloudWatch bitmap veloce: esegui il rendering CloudWatch dei grafici sul lato server, per una visualizzazione rapida.
- Send dashboard snapshot by email (Invia snapshot del pannello di controllo tramite e-mail): scatta uno snapshot del pannello di controllo corrente e invialo ai destinatari dell'e-mail.
- Send dashboard snapshot to Amazon S3 (Invia snapshot del pannello di controllo ad Amazon S3): scatta uno snapshot del pannello di controllo corrente e archivialo in Amazon S3.
- Widget di testo dal CloudWatch pannello di controllo: visualizza il primo widget di testo dal pannello di controllo specificato CloudWatch .
- Display content of external URL (Visualizza il contenuto dell'URL esterno): visualizza il contenuto di un URL accessibile esternamente.
- RSS reader (Lettore RSS): visualizza i feed RSS.
- Display an Amazon S3 object (Visualizza un oggetto Amazon S3): visualizza un oggetto in un bucket Amazon S3 nel tuo account.
- Simple SVG pie chart (Semplice grafico a torta SVG): esempio di widget grafico basato su SVG.

Widget Echo in JavaScript

Di seguito è riportato il widget di esempio di Echo in. JavaScript

```
const DOCS = `
## Echo
A basic echo script. Anything passed in the \`\`\`echo\`\`\` parameter is returned as
 the content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
\`\`\` yaml
echo: <h1>Hello world</h1>
\`\`\`
`;

exports.handler = async (event) => {
    if (event.describe) {
        return DOCS;
    }

    let widgetContext = JSON.stringify(event.widgetContext, null, 4);
    widgetContext = widgetContext.replace(/</g, '&lt;');
    widgetContext = widgetContext.replace(/>/g, '&gt;');

    return `${event.echo || ''}<pre>${widgetContext}</pre>`;
};
```

Widget Echo in Python

Di seguito è riportato il widget di esempio Echo in Python.

```
import json

DOCS = """
## Echo
A basic echo script. Anything passed in the ```echo``` parameter is returned as the
 content of the custom widget.
### Widget parameters
Param | Description
```

```
---|---
**echo** | The content to echo back

### Example parameters
``` yaml
echo: <h1>Hello world</h1>
```"""

def lambda_handler(event, context):
    if 'describe' in event:
        return DOCS

    echo = event.get('echo', '')
    widgetContext = event.get('widgetContext')
    widgetContext = json.dumps(widgetContext, indent=4)
    widgetContext = widgetContext.replace('<', '<')
    widgetContext = widgetContext.replace('>', '>')

    return f'{echo}<pre>{widgetContext}</pre>'
```

Widget Echo in Java

Di seguito è riportato il widget di esempio Echo in Java.

```
package example;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

public class Handler implements RequestHandler<Event, String>{

  static String DOCS = ""
    + "## Echo\n"
    + "A basic echo script. Anything passed in the ```echo``` parameter is returned as
 the content of the custom widget.\n"
    + "### Widget parameters\n"
    + "Param | Description\n"
    + "---|---\n"
    + "**echo** | The content to echo back\n\n"
    + "### Example parameters\n"
    + "```yaml\n"
```

```
    + "echo: <h1>Hello world</h1>\n"
    + "```\n";

  Gson gson = new GsonBuilder().setPrettyPrinting().create();

  @Override
  public String handleRequest(Event event, Context context) {

    if (event.describe) {
      return DOCS;
    }

    return (event.echo != null ? event.echo : "") + "<pre>" +
 gson.toJson(event.widgetContext) + "</pre>";
  }
}

class Event {

    public boolean describe;
    public String echo;
    public Object widgetContext;

    public Event() {}

    public Event(String echo, boolean describe, Object widgetContext) {
        this.describe = describe;
        this.echo = echo;
        this.widgetContext = widgetContext;
    }
}
```

Aggiungere o rimuovere un widget di testo da una dashboard CloudWatch

Un widget di testo contiene un blocco di testo in formato Markdown. Puoi aggiungere, modificare o rimuovere widget di testo dalla CloudWatch dashboard.

Aggiunta di un widget di testo a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.

3. Scegli il simbolo +.
4. Scegli Testo.
5. In Markdown, aggiungi e formatta il testo utilizzando Markdown, quindi seleziona Create widget (Crea widget).
6. Per rendere trasparente il widget di testo, scegli Sfondo trasparente.
7. Seleziona Salva pannello di controllo.

Modifica di un widget di testo in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Passa con il mouse sull'angolo in alto a destra del blocco di testo, quindi scegli Widget actions (Operazioni del widget). Quindi scegli Edit (Modifica).
4. Aggiorna il testo in base alle esigenze, quindi seleziona Update widget (Aggiorna widget).
5. Seleziona Salva pannello di controllo.

Rimozione di un widget di testo da un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Passa con il mouse sull'angolo in alto a destra del blocco di testo, quindi scegli Widget actions (Operazioni del widget). Quindi, scegli Elimina.
4. Seleziona Salva pannello di controllo.

Aggiungere o rimuovere un widget di allarme da una CloudWatch dashboard

Per aggiungere un widget allarme a un pannello di controllo, scegli una delle seguenti opzioni:

- Aggiungi a un widget un singolo allarme che visualizza sia il grafico del parametro dell'allarme sia lo stato dell'allarme.

- Aggiungi un widget di stato degli allarmi che mostra lo stato di più allarmi in una griglia. Vengono visualizzati solo i nomi e lo stato corrente degli allarmi; i grafici non vengono visualizzati. In un widget di stato degli allarmi puoi includere fino a 100 allarmi.

Per aggiungere un singolo avviso, incluso il relativo grafico, a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Alarms (Allarmi), seleziona l'allarme da aggiungere e quindi seleziona Add to Dashboard (Aggiungi al pannello di controllo).
3. Seleziona un pannello di controllo, scegli un tipo di widget (Line (Linea), Stacked (In pila), o Number (Numero)) e quindi seleziona Add to dashboard (Aggiungi al pannello di controllo).
4. Per visualizzare l'allarme nel pannello di controllo, seleziona Dashboards (Pannelli di controllo) nel pannello di navigazione e seleziona il pannello di controllo.
5. (Facoltativo) Per ingrandire temporaneamente un grafico di allarme, seleziona il grafico.
6. (Facoltativo) Per modificare il tipo di widget, passa con il mouse sul titolo del grafico e seleziona Operazioni widget, Tipo di widget.

Per aggiungere un widget dello stato dell'allarme a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Scegli il simbolo +.
4. Scegli Alarm status (Stato allarme).
5. Seleziona le caselle di spunta accanto agli allarmi da aggiungere al widget, quindi seleziona Create widget (Crea widget).
6. Scegli Add to dashboard (Aggiungi a pannello di controllo).

Rimozione di un widget dell'allarme da un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Passa con il mouse sul widget, scegli Operazioni widget, quindi scegli Elimina.

4. Seleziona Salva pannello di controllo. Se tenti di uscire dal pannello di controllo prima di salvare le modifiche, ti verrà chiesto di salvare o eliminare le modifiche.

Aggiungere o rimuovere un widget di tabella dati da una CloudWatch dashboard

Con il widget tabella di dati, puoi vedere i punti dati non elaborati del tuo parametro e un breve riepilogo di tali dati. Poiché il widget tabella di dati non è un grafico per astrarre i dati reali dall'utente, è più facile comprendere i punti dati presentati. Le procedure in questa sezione descrivono come aggiungere e rimuovere un widget di tabella dati da una CloudWatch dashboard.

Label	Min	Max	Sum	Average	11/20 06:00	11/20 00:00	11/19 18:00	11/19 12:00	11/ 06:
TestMetric295	991	1,000	12k	998	996	1,000	997	999	
TestMetric296	995	1,000	12k	998	995	1,000	1,000	998	
TestMetric297	991	1,000	12k	998	998	1,000	999	997	
TestMetric298	994	1,000	12k	997	996	999	995	995	
TestMetric3	993	1,000	12k	998	1,000	999	999	1,000	
TestMetric299	995	999	12k	998	999	995	999	998	
TestMetric30	994	999	12k	998	999	998	999	999	
StackMetric2	99	99.9	1.2k	99.6	99.2	99.7	99.5	99.8	
StackMetric20	99	100	1.19k	99.5	100	99.1	99.4	99.4	
StackMetric21	97.5	100	1.19k	99.4	99.6	99.7	97.6	99.8	

Proprietà tabella

Una tabella di dati ha un set predefinito di proprietà che non richiedono alcuna modifica alle opzioni o all'origine. Queste proprietà includono una colonna con etichette permanenti, tutte le colonne di riepilogo abilitate, i punti dati arrotondati e le relative unità convertite.

Ogni widget tabella di dati può avere le seguenti proprietà. Le informazioni su ogni proprietà includono come configurarla nell'origine JSON del pannello di controllo. Per ulteriori informazioni sul pannello di controllo JSON, consulta Struttura e sintassi del corpo del pannello di controllo.

Riepilogo

Le colonne di riepilogo sono una nuova proprietà introdotta con il widget tabella di dati. Queste colonne sono un sottoinsieme specifico di riepiloghi della tabella corrente. Ad esempio, il riepilogo Sum è la somma di tutti i punti dati visualizzati nella riga corrispondente. Le colonne di riepilogo non sono le stesse CloudWatch delle statistiche. Rappresentate nell'origine come:

```
 "table": {
        "summaryColumns": [
            "MIN",
            "MAX",
            "SUM",
            "AVG"
        ]
    },
```

Soglie

Da utilizzare per applicare soglie alla tabella. Quando un punto dati rientra in una soglia, la relativa cella viene evidenziata con il colore della soglia. Rappresentate nell'origine come:

```
"annotations": {
    "horizontal": [
        {
            "label": string,
            "value": int,
            "fill": "above" | "below"
        }
    ]
}
```

Unità nella colonna dell'etichetta

Per visualizzare l'unità associata al parametro, puoi abilitare questa opzione per visualizzare l'unità nella colonna dell'etichetta accanto all'etichetta. Rappresentate nell'origine come:

```
"yAxis": {
    "left": {
        "showUnits": true | false
    }
}
```

Inverti righe e colonne

Trasforma la tabella in modo che i punti dati da colonne passino a righe e i parametri diventino colonne. Rappresentate nell'origine come:

```
 "table": {
```

```
    "layout": "vertical" | "horizontal"
}
```

Colonne di riepilogo permanenti

Rende le colonne di riepilogo permanenti, in modo che rimangano visibili durante lo scorrimento. L'etichetta è già permanente. Rappresentate nell'origine come:

```
"table": {
    "stickySummary": true | false
}
```

Visualizza solo le colonne di riepilogo

Impedisce la visualizzazione delle colonne dei punti dati, in modo che vengano visualizzate solo le colonne di etichette e riepilogo. Rappresentate nell'origine come:

```
 "table": {
    "showTimeSeriesData": false | true
}
```

Dati in tempo reale

Visualizza il punto dati più recente, anche se non è ancora completamente aggregato. Rappresentate nell'origine come:

```
"liveData": true | false
```

Formato del widget numerico

Visualizza tutte le cifre che può contenere la cella, prima dell'arrotondamento e della conversione. Rappresentate nell'origine come:

```
"singleValueFullPrecision": true | false
```

Per aggiungere un widget tabella di dati a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione, seleziona Pannelli di controllo, quindi scegli un pannello di controllo.
3. Seleziona il pulsante +, seleziona Tabella dati e scegli Successivo.
4. Nella scheda Sfoglia, cerca o sfoglia i parametri da visualizzare nel widget tabella. Seleziona la scheda Parametri.
5. (Facoltativo) Per modificare il layout della tabella, scegli la scheda Opzioni e seleziona Inverti righe e colonne.

   Puoi anche utilizzare la scheda Opzioni per modificare le colonne visualizzate nella tabella e mostrare l'unità utilizzata nella colonna Etichetta.

   > **Tip**
   >
   > Per visualizzare soglie più accurate, scegli Mostra tutte le cifre possibili prima dell'arrotondamento.

6. (Facoltativo) Per modificare l'intervallo di tempo del widget tabella di dati, seleziona uno degli intervalli temporali predefiniti nell'area in alto nel widget. L'intervallo di tempo va da 1 ora a 1 settimana. Per impostare l'intervallo di tempo, scegli Personalizzato.
7. (Facoltativo) Per modificare l'intervallo di tempo del widget tabella di dati, seleziona uno degli intervalli temporali predefiniti nell'area in alto nel widget. L'intervallo di tempo va da 1 ora a 1 settimana. Per impostare l'intervallo di tempo, scegli Personalizzato.
8. (Facoltativo) Per fare in modo che questo widget continui a utilizzare l'intervallo di tempo selezionato, anche se l'intervallo di tempo per il resto del pannello di controllo viene successivamente modificato, scegli Rendi permanente l'intervallo di tempo.
9. Scegli Crea widget, quindi scegli Salva pannello di controllo.

Per rimuovere un widget tabella di dati da un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Nell'angolo in alto a destra del widget da rimuovere, scegli Operazioni del widget, quindi scegli Elimina.
4. Seleziona Salva pannello di controllo.

Collega e scollega i grafici su una dashboard CloudWatch

Puoi collegare insieme i grafici nel pannello di controllo, in modo che, quando ingrandisci o rimpicciolisci un grafico, gli altri grafici si ingrandiscono o rimpiccioliscono allo stesso tempo. Puoi scollegare i grafici per limitare la modifica delle dimensioni a un solo grafico.

Per collegare i grafici in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Scegli Actions (Operazioni) e quindi Link graphs (Collega grafici).

Per scollegare i grafici in un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Deseleziona Actions (Operazioni) e quindi Link graphs (Collega grafici).

CloudWatch Dashboard di condivisione

Puoi condividere le tue CloudWatch dashboard con persone che non hanno accesso diretto al tuo AWS account. In questo modo puoi condividere pannelli di controllo tra i team, con le parti interessate e con persone esterne all'organizzazione. Puoi anche visualizzare pannelli di controllo su grandi schermi nelle aree del team, o incorporarli in pagine Wiki e altre pagine Web.

> **Warning**
>
> A tutte le persone con cui condividi il pannello di controllo vengono concesse le autorizzazioni elencate in Autorizzazioni concesse agli utenti con cui condividi il pannello di controllo per l'account. Se condividi pubblicamente il pannello di controllo, tutti coloro che dispongono del collegamento al pannello di controllo dispongono di queste autorizzazioni.
> Le `ec2:DescribeTags` autorizzazioni `cloudwatch:GetMetricData` e non possono essere limitate a metriche o istanze EC2 specifiche, quindi le persone con accesso alla

dashboard possono interrogare tutte le CloudWatch metriche e i nomi e i tag di tutte le istanze EC2 presenti nell'account.

Quando si condividono pannelli di controllo, è possibile specificare chi può visualizzare il pannello di controllo in tre modi:

- Condividi un'unica dashboard e designa fino a cinque indirizzi e-mail di persone che possono visualizzarla. Ognuno di questi utenti crea la propria password che deve immettere per visualizzare il pannello di controllo.
- Condividi pubblicamente un singolo pannello di controllo, in modo che chiunque disponga del collegamento possa visualizzarlo.
- Condividi tutte le CloudWatch dashboard del tuo account e specifica un provider Single Sign-On (SSO) di terze parti per l'accesso alla dashboard. Tutti gli utenti membri dell'elenco di questo provider SSO possono accedere a tutti i pannelli di controllo dell'account. Per abilitarlo, integra il provider SSO con Amazon Cognito. Il provider SSO deve supportare Security Assertion Markup Language (SAML). Per ulteriori informazioni su Amazon Cognito, consulta Che cos'è Amazon Cognito?

La condivisione di una dashboard non comporta costi, ma i widget all'interno di una dashboard condivisa comportano addebiti a tariffe standard. CloudWatch Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Quando condividi una dashboard, le risorse di Amazon Cognito vengono create nella regione Stati Uniti orientali (Virginia settentrionale).

Important

Non modificare i nomi e gli identificatori delle risorse creati dal processo di condivisione del pannello di controllo. Ciò include le risorse Amazon Cognito e IAM. La modifica di queste risorse può causare funzionalità impreviste e non corrette dei pannelli di controllo condivisi.

Note

Se condividi un pannello di controllo contenente widget parametrici con annotazioni di allarme, le persone con cui condividi il pannello di controllo non visualizzeranno tali widget.

Vedranno invece un widget vuoto con un testo che indica che il widget non è disponibile. Quando visualizzi personalmente il pannello di controllo, visualizzerai comunque i widget parametrici con le annotazioni di allarme.

Autorizzazioni necessarie per condividere un pannello di controllo

Per poter condividere pannelli di controllo utilizzando uno dei metodi descritti di seguito e per vedere quali pannelli di controllo sono già stati condivisi, devi essere connesso come utente IAM o con un ruolo IAM che dispone di determinate autorizzazioni.

Per poter condividere pannelli di controllo, l'utente o il ruolo IAM deve includere le autorizzazioni incluse nella seguente istruzione di policy:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/service-role/CWDBSharing*",
        "arn:aws:iam::*:policy/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*",
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:GetDashboard",
    ],
```

```
    "Resource": [
        "*"
        // or the ARNs of dashboards that you want to share
    ]
}
```

Per vedere quali pannelli di controllo sono condivisi, ma senza poterli condividere, un utente IAM o un ruolo IAM può includere un'istruzione di policy simile alla seguente:

```
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:ListDashboards",
    ],
    "Resource": [
        "*"
    ]
}
```

Autorizzazioni concesse agli utenti con cui condividi il pannello di controllo

Quando condividi una dashboard, CloudWatch crea un ruolo IAM nell'account che concede le seguenti autorizzazioni alle persone con cui condividi la dashboard:

- `cloudwatch:GetInsightRuleReport`
- `cloudwatch:GetMetricData`
- `cloudwatch:DescribeAlarms`
- `ec2:DescribeTags`

> **Warning**
>
> A tutte le persone con cui condividi il pannello di controllo vengono concesse queste autorizzazioni per l'account. Se condividi pubblicamente il pannello di controllo, tutti coloro che dispongono del collegamento al pannello di controllo dispongono di queste autorizzazioni.
> Le `ec2:DescribeTags` autorizzazioni `cloudwatch:GetMetricData` e non possono essere limitate a metriche o istanze EC2 specifiche, quindi le persone con accesso alla dashboard possono interrogare tutte le CloudWatch metriche e i nomi e i tag di tutte le istanze EC2 presenti nell'account.

Quando condividi una dashboard, per impostazione predefinita le autorizzazioni CloudWatch create limitano l'accesso solo agli allarmi e alle regole di Contributor Insights presenti nella dashboard quando viene condivisa. Se si aggiungono nuovi avvisi o regole di Contributor Insights dei collaboratori al pannello di controllo e si desidera che vengano visualizzati anche dagli utenti con cui è stato condiviso il pannello di controllo, devi aggiornare la policy per consentire queste risorse.

Condivisione di un singolo pannello di controllo con utenti specifici

Utilizza i passaggi descritti in questa sezione per condividere una dashboard con un massimo di cinque indirizzi email a tua scelta.

> **Note**
>
> Per impostazione predefinita, tutti CloudWatch i widget Logs sulla dashboard non sono visibili alle persone con cui condividi la dashboard. Per ulteriori informazioni, consulta Consentire alle persone con cui condividi di visualizzare i widget della tabella dei log.
> Per impostazione predefinita, gli eventuali widget di allarme compositi presenti nel pannello di controllo non sono visibili agli utenti con cui condividi il pannello di controllo. Per ulteriori informazioni, consulta la pagina Consentire alle persone con cui condividi di vedere allarmi compositi.

Per condividere un pannello di controllo con utenti specifici

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).

3. Scegli il nome del tuo pannello di controllo.
4. Scegli Actions (Operazioni), Share dashboard (Condividi pannello di controllo).
5. Accanto a Share your dashboard and require a username and password (Condividi il pannello di controllo e richiedi nome utente e password), scegli Start sharing (Inizia condivisione).
6. In Add email addresses (Aggiungi indirizzi e-mail) immetti l'indirizzo e-mail con cui desideri condividere il pannello di controllo. Puoi includere fino a cinque indirizzi e-mail.
7. Dopo aver inserito tutti gli indirizzi e-mail, leggi l'accordo e seleziona la casella di conferma. Seleziona Preview policy (Anteprima policy).
8. Conferma che le risorse che verranno condivise siano quelle che desideri e scegli Confirm and generate shareable link (Conferma e genera collegamento condivisibile).
9. Nella pagina successiva, scegli Copy link to clipboard (Copia collegamento negli appunti). Puoi quindi incollare questo collegamento nell'e-mail e inviarlo agli utenti invitati. Riceveranno automaticamente un'e-mail separata con il loro nome utente e una password temporanea da utilizzare per connettersi al pannello di controllo.

Condividere pubblicamente un singolo pannello di controllo

Per condividere pubblicamente un pannello di controllo, segui la procedura descritta in questa sezione. Può essere utile per visualizzare il pannello di controllo su un grande schermo in una stanza del team o incorporarlo in una pagina Wiki.

> **Important**
>
> La condivisione di un pannello di controllo lo rende pubblicamente accessibile a chiunque disponga del collegamento, senza autenticazione. Effettua questa operazione solo per i pannelli di controllo che non contengono informazioni riservate.

> **Note**
>
> Per impostazione predefinita, tutti CloudWatch i widget Logs sulla dashboard non sono visibili alle persone con cui condividi la dashboard. Per ulteriori informazioni, consulta Consentire alle persone con cui condividi di visualizzare i widget della tabella dei log.
> Per impostazione predefinita, gli eventuali widget di allarme compositi presenti nel pannello di controllo non sono visibili agli utenti con cui condividi il pannello di controllo. Per ulteriori

informazioni, consulta la pagina Consentire alle persone con cui condividi di vedere allarmi compositi.

Per condividere pubblicamente un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Scegli il nome del tuo pannello di controllo.
4. Scegli Actions (Operazioni), Share dashboard (Condividi pannello di controllo).
5. Accanto a Share your dashboard publicly (Condividi pubblicamente il tuo pannello di controllo), scegli Start sharing (Inizia condivisione).
6. Immetti **`Confirm`** nella casella di testo.
7. Leggi l'accordo e seleziona la casella di conferma. Seleziona Preview policy (Anteprima policy).
8. Conferma che le risorse che verranno condivise siano quelle che desideri e scegli Confirm and generate shareable link (Conferma e genera collegamento condivisibile).
9. Nella pagina successiva, scegli Copy link to clipboard (Copia collegamento negli appunti). È quindi possibile condividere questo collegamento. Chiunque condivida il collegamento può accedere al pannello di controllo, senza fornire credenziali.

Condividi tutte le CloudWatch dashboard dell'account utilizzando SSO

Segui la procedura descritta in questa sezione per condividere tutti i pannelli di controllo dell'account con gli utenti tramite Single Sign-On (SSO).

> **Note**
>
> Per impostazione predefinita, tutti CloudWatch i widget Logs sulla dashboard non sono visibili alle persone con cui condividi la dashboard. Per ulteriori informazioni, consulta Consentire alle persone con cui condividi di visualizzare i widget della tabella dei log.
> Per impostazione predefinita, gli eventuali widget di allarme compositi presenti nel pannello di controllo non sono visibili agli utenti con cui condividi il pannello di controllo. Per ulteriori informazioni, consulta Consentire alle persone con cui condividi di vedere allarmi compositi.

Per condividere le CloudWatch dashboard con gli utenti inclusi nell'elenco di un provider SSO

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Scegli il nome del tuo pannello di controllo.
4. Scegli Actions (Operazioni), Share dashboard (Condividi pannello di controllo).
5. Scegli Vai alle CloudWatch impostazioni.
6. Se il provider SSO desiderato non è elencato in Available SSO providers (Provider SSO disponibili), scegli Manage SSO providers (Gestisci provider SSO) e segui le istruzioni in Configura SSO per CloudWatch la condivisione della dashboard.

   Quindi torna alla CloudWatch console e aggiorna il browser. Il provider SSO abilitato dovrebbe ora comparire nell'elenco.
7. Scegli il provider SSO che desideri nell'elenco Available SSO providers (Provider SSO disponibili).
8. Sceglie Save changes (Salva modifiche).

Configura SSO per CloudWatch la condivisione della dashboard

Per impostare la condivisione del pannello di controllo tramite un provider Single Sign-On di terze parti che supporta SAML, procedi come segue.

> **Important**
>
> Ti consigliamo di non condividere pannelli di controllo utilizzando un provider SSO non SAML. In questo modo rischierai di consentire inavvertitamente a terze parti di accedere ai pannelli di controllo del tuo account.

Per impostare un provider SSO per abilitare la condivisione del pannello di controllo

1. Integra il provider SSO con Amazon Cognito. Per ulteriori informazioni, consulta Integrazione di provider di identità SAML di terze parti con bacini d'utenza di Amazon Cognito.
2. Scarica il file XML dei metadati dal provider SSO.
3. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

4. Nel pannello di navigazione scegli Impostazioni.
5. Nella sezione Dashboard sharing (Condivisione del pannello), scegli Configure (Configura).
6. Scegli Manage SSO providers (Gestisci provider SSO).

   Si aprirà la console Amazon Cognito nella regione degli Stati Uniti orientali (Virginia settentrionale) (us-east-1). Se non vengono visualizzati User Pools (Bacini d'utenza), la console Amazon Cognito potrebbe essere stata aperta in una regione diversa. In questo caso, modifica la regione in US East (N. Virginia) us-east-1 (Stati Uniti orientali (Virginia settentrionale) us-east-1) e procedi con i passaggi successivi.
7. Scegli la CloudWatchDashboardSharingpiscina.
8. Nel pannello di navigazione, scegli Identity providers (Provider di identità).
9. Scegli SAML.
10. Immetti un nome per il provider SSO in Provider name (Nome provider).
11. Scegli Select file (Seleziona file) e seleziona il file XML dei metadati scaricato nel passaggio 1.
12. Scegli Create provider (Crea provider).

Scopri quanti pannelli di controllo sono condivisi

Puoi utilizzare la CloudWatch console per vedere quante delle tue CloudWatch dashboard sono attualmente condivise con altri.

Per vedere quanti pannelli di controllo vengono condivisi

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Settings (Impostazioni).
3. La sezione Dashboard sharing (Condivisione del pannello) consente di visualizzare il numero di pannelli di controllo condivisi.
4. Per vedere quali pannelli di controllo sono condivisi, scegli ***number*** dashboards shared (numero di pannelli di controllo condivisi) in Username and password (Nome utente e password) e in Public dashboards (Pannelli di controllo pubblici).

Guarda quali pannelli di controllo sono condivisi

Puoi utilizzare la CloudWatch console per vedere quali delle tue dashboard sono attualmente condivise con altri.

Per vedere quali pannelli di controllo sono condivisi

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Nell'elenco dei pannelli di controllo, guarda la colonna Share (Condividi). I pannelli di controllo con l'icona in questa colonna compilata sono attualmente in fase di condivisione.
4. Per vedere con quali utenti viene condiviso un pannello di controllo, scegli il nome del pannello di controllo, quindi scegli Actions (Operazioni), Share dashboard (Condividi pannello di controllo).

   La pagina Share dashboard ***dashboard name*** (Condividi pannello di controllo - Nome pannello di controllo) mostra la modalità di condivisione del pannello di controllo. Se lo desideri, puoi interrompere la condivisione del pannello di controllo scegliendo Stop sharing (Interrompi condivisione).

Interrompi la condivisione di uno o più pannelli di controllo

Puoi interrompere la condivisione di un singolo pannello di controllo condiviso o interrompere la condivisione di tutti i pannelli di controllo condivisi contemporaneamente.

Per interrompere la condivisione di un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Scegli il nome del pannello di controllo condiviso.
4. Scegli Actions (Operazioni), Share dashboard (Condividi pannello di controllo).
5. Scegli Stop sharing (Interrompi condivisione).
6. Nella finestra di conferma scegli Stop sharing (Interrompi condivisione).

Per interrompere la condivisione di tutti i pannelli di controllo condivisi

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Impostazioni.
3. In Dashboard sharing (Condivisione del pannello di controllo), scegli Stop sharing all dashboards (Interrompi la condivisione di tutti i pannelli di controllo).

4. Nella finestra di conferma scegli Stop sharing all dashboards (Interrompi la condivisione di tutti i pannelli di controllo).

Esamina le autorizzazioni del pannello di controllo condiviso e modifica l'ambito delle autorizzazioni

Segui la procedura descritta in questa sezione se si desidera esaminare le autorizzazioni degli utenti dei pannelli di controllo condivisi o modificare l'ambito delle autorizzazioni del pannello di controllo condiviso.

Per esaminare le autorizzazioni del pannello di controllo condiviso

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Scegli il nome del pannello di controllo condiviso.
4. Scegli Actions (Operazioni), Share dashboard (Condividi pannello di controllo).
5. In Resources (Risorse), scegli IAM Role (Ruolo IAM).
6. Nella console IAM, sceglie la policy visualizzata.
7. (Facoltativo) Per limitare gli allarmi che gli utenti del pannello di controllo condiviso possono visualizzare, scegli Edit policy (Modifica policy) e sposta l'autorizzazione `cloudwatch:DescribeAlarms` dalla sua posizione corrente a una nuova istruzione `Allow` che elenca gli ARN solo degli allarmi che vuoi mostrare agli utenti del pannello di controllo condiviso. Guarda l'esempio seguente.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "AlarmARN1",
        "AlarmARN2"
    ]
}
```

Se esegui questa operazione, assicurati di rimuovere l'autorizzazione `cloudwatch:DescribeAlarms` da una sezione della policy corrente che assomiglia a questa:

```
{
```

```
   "Effect": "Allow",
    "Action": [
        "cloudwatch:GetInsightRuleReport",
        "cloudwatch:GetMetricData",
        "cloudwatch:DescribeAlarms",
        "ec2:DescribeTags"
    ],
    "Resource": "*"
}
```

8. (Facoltativo) Per limitare l'ambito delle regole di Contributor Insights che gli utenti del pannello di controllo condiviso possono vedere, scegli Edit policy (Modifica policy) e sposta `cloudwatch:GetInsightRuleReport` dalla sua posizione attuale a una nuova istruzione `Allow` che elenca gli ARN delle sole regole di Contributor Insights vuoi mostrare agli utenti del pannello di controllo condiviso. Guarda l'esempio seguente.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:GetInsightRuleReport",
    "Resource": [
        "PublicContributorInsightsRuleARN1",
        "PublicContributorInsightsRuleARN2"
    ]
}
```

Se esegui questa operazione, assicurati di rimuovere `cloudwatch:GetInsightRuleReport` da una sezione della policy corrente che assomiglia a questa:

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetInsightRuleReport",
                "cloudwatch:GetMetricData",
                "cloudwatch:DescribeAlarms",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        }
```

Consentire alle persone con cui condividi di vedere allarmi compositi

Quando condividi un pannello di controllo, per impostazione predefinita i widget di allarme composito sul pannello di controllo non sono visibili agli utenti con cui si condivide il pannello di controllo. Affinché i widget di allarme composito siano visibili, devi aggiungere un'autorizzazione `DescribeAlarms: *` alle policy di condivisione del pannello di controllo. L'autorizzazione avrebbe il seguente aspetto:

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
}
```

Warning

L'istruzione precedente offre l'accesso a tutti gli allarmi presenti nell'account. Per ridurre l'ambito di `cloudwatch:DescribeAlarms`, devi utilizzare un'istruzione `Deny`. È possibile aggiungere un'istruzione `Deny` alla policy e specificare gli ARN degli allarmi che desideri bloccare. Questa istruzione di negazione dovrebbe essere simile alla seguente:

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
},
{
    "Effect": "Deny",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "SensitiveAlarm1ARN",
        "SensitiveAlarm1ARN"
    ]
}
```

Consentire alle persone con cui condividi di visualizzare i widget della tabella dei log

Quando condividi una dashboard, per impostazione predefinita i widget di CloudWatch Logs Insights presenti nella dashboard non sono visibili alle persone con cui condividi la dashboard. Ciò influisce sia sui widget di CloudWatch Logs Insights esistenti sia su quelli aggiunti alla dashboard dopo la condivisione.

Se vuoi che queste persone possano vedere CloudWatch i widget di Logs, devi aggiungere le autorizzazioni al ruolo IAM per la condivisione della dashboard.

Per consentire alle persone con cui condividi una dashboard di visualizzare i widget Logs CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Scegli il nome del pannello di controllo condiviso.
4. Scegli Actions (Operazioni), Share dashboard (Condividi pannello di controllo).
5. In Resources (Risorse), scegli IAM Role (Ruolo IAM).
6. Nella console IAM, scegli la policy visualizzata.
7. Scegli Edit policy (Modifica policy) e aggiungi la seguente istruzione. Nella nuova istruzione, ti consigliamo di specificare gli ARN solo dei gruppi di log che desideri condividere. Guarda l'esempio seguente.

```
{
            "Effect": "Allow",
            "Action": [
                "logs:FilterLogEvents",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:GetLogRecord",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "SharedLogGroup1ARN",
                "SharedLogGroup2ARN"
           ]
        },
```

8. Seleziona Salva modifiche.

Se la policy IAM per la condivisione del pannello di controllo include già queste cinque autorizzazioni con * come risorsa, ti consigliamo vivamente di modificare la policy e specificare solo gli ARN dei gruppi di log che desideri condividere. Ad esempio, se la sezione `Resource` per queste autorizzazioni era la seguente:

```
"Resource": "*"
```

Modifica la policy per specificare solo gli ARN dei gruppi di log che desideri condividere, come nell'esempio seguente:

```
"Resource": [
  "SharedLogGroup1ARN",
  "SharedLogGroup2ARN"
]
```

Consentire alle persone con cui condividi di visualizzare i widget personalizzati

Quando condividi un pannello di controllo, per impostazione predefinita i widget personalizzati presenti nel pannello di controllo non sono visibili agli utenti con cui condividi il pannello di controllo. Ciò influisce sia sui widget personalizzati esistenti che su quelli aggiunti al pannello di controllo dopo la condivisione.

Se desideri che queste persone siano in grado di visualizzare widget personalizzati, devi aggiungere autorizzazioni al ruolo IAM per la condivisione del pannello di controllo.

Per consentire agli utenti con cui condividi un pannello di controllo di visualizzare i widget personalizzati

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Scegli il nome del pannello di controllo condiviso.
4. Scegli Actions (Operazioni), Share dashboard (Condividi pannello di controllo).
5. In Resources (Risorse), scegli IAM Role (Ruolo IAM).

6. Nella console IAM, scegli la policy visualizzata.
7. Scegli Edit policy (Modifica policy) e aggiungi la seguente istruzione. Nella nuova istruzione, ti consigliamo di specificare gli ARN solo delle funzioni Lambda che desideri condividere. Guarda l'esempio seguente.

```
{
  "Sid": "Invoke",
  "Effect": "Allow",
  "Action": [
      "lambda:InvokeFunction"
  ],
  "Resource": [
    "LambdaFunction1ARN",
    "LambdaFunction2ARN"
  ]
 }
```

8. Seleziona Salva modifiche.

Se la policy IAM per la condivisione del pannello di controllo include già tale autorizzazione con *come risorsa, ti consigliamo vivamente di modificare la olicy e di specificare solo gli ARN delle funzioni Lambda che desideri condividere. Ad esempio, se la sezione `Resource` per queste autorizzazioni era la seguente:

```
"Resource": "*"
```

Modifica la policy per specificare solo gli ARN dei widget personalizzati che desideri condividere, come nell'esempio seguente:

```
"Resource": [
  "LambdaFunction1ARN",
  "LambdaFunction2ARN"
]
```

Uso dei dati in tempo reale

Puoi scegliere se i widget parametro visualizzano i dati in tempo reale. I dati in tempo reale sono dati pubblicati all'ultimo minuto che non sono stati completamente aggregati.

- Se i dati in tempo reale sono disattivati, vengono visualizzati solo i punti dati con un periodo di aggregazione di almeno un minuto nel passato. Ad esempio, quando si utilizzano periodi di 5 minuti, il punto dati per le 12:35 verrà aggregato dalle 12:35 alle 12:40 e visualizzato alle 12:41.
- Se i dati in tempo reale sono attivati, il punto dati più recente viene visualizzato non appena i dati vengono pubblicati nell'intervallo di aggregazione corrispondente. Ogni volta che si aggiorna la visualizzazione, il punto dati più recente può cambiare se vengono pubblicati nuovi dati all'interno di tale periodo di aggregazione. Se usi una statistica cumulativa, ad esempio Somma o Conteggio del campione, l'utilizzo di dati in tempo reale potrebbe comportare un calo alla fine del grafico.

Puoi scegliere di abilitare i dati in tempo reale per un intero pannello di controllo o per i singoli widget nel pannello di controllo.

Per scegliere se utilizzare i dati in tempo reale sull'intero pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Per attivare o disattivare definitivamente i dati in tempo reale per tutti i widget sul pannello di controllo, effettua le seguenti operazioni:

   a. Scegli Actions (Operazioni), Settings (Impostazioni), Bulk update live data (Aggiornamento in blocco dati in tempo reale).

   b. Scegli Live Data on (Dati in tempo reale attivati) o Live Data off (Dati in tempo reale disattivati) e seleziona Set (Imposta).
4. Per sovrascrivere temporaneamente le impostazioni dei dati in tempo reale di ciascun widget, scegli Actions (Operazioni). Quindi, in Overrides (Sostituzioni), accanto a Live data (Dati live effettua una delle seguenti operazioni:

   - Scegli On (Attiva) per attivare temporaneamente i dati in tempo reale per tutti i widget.
   - Scegli Off (Disattiva) per disattivare temporaneamente i dati in tempo reale per tutti i widget.
   - Scegli Do not override (Non sostituire) per preservare l'impostazione dei dati in tempo reale di ogni widget.

Per scegliere se utilizzare i dati in tempo reale su un singolo widget

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Seleziona un widget e scegli Actions (Operazioni), Edit (Modifica).
4. Scegli la scheda Graph options (Opzioni grafica).
5. Seleziona o deseleziona la casella di controllo in Live Data (Dati in tempo reale).

Visualizzazione di un pannello di controllo animato

Puoi visualizzare una dashboard animata che riproduce i dati CloudWatch metrici acquisiti nel tempo. In questo modo puoi visualizzare le tendenze, creare presentazioni o analizzare i problemi dopo che si sono verificati.

I widget animati nel pannello di controllo includono widget linea, widget area impilata, widget numerici e widget di esplorazione dei parametri. I grafici a torta, i grafici a barre, i widget di testo e i widget dei log vengono visualizzati nel pannello di controllo ma non sono animati.

Per visualizzare un pannello di controllo animato

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Scegli il nome del pannello di controllo.
4. Scegli Actions (Operazioni), Replay dashboard (Riproduci pannello di controllo).
5. (Facoltativo) Per impostazione predefinita, quando avvii l'animazione, questa viene visualizzata come una finestra scorrevole. Se invece desideri che l'animazione appaia come point-by-point animazione, scegli l'icona della lente di ingrandimento mentre l'animazione è in pausa e ripristina lo zoom.
6. Per avviare l'animazione, scegli il pulsante Play (Riproduci). Puoi anche scegliere i pulsanti Back (Indietro) e Forward (Avanti) per spostarti in altri punti nel tempo.
7. (Facoltativo) Per modificare la finestra temporale dell'animazione, scegli il calendario e seleziona il periodo di tempo.
8. Per modificare la velocità dell'animazione, scegli Auto speed (Velocità automatica) e seleziona la nuova velocità.
9. Al termine, seleziona Exit animate (Esci dall'animazione).

Aggiungi una CloudWatch dashboard all'elenco dei preferiti

Nella CloudWatch console, puoi aggiungere dashboard, allarmi e gruppi di log a un elenco di preferiti. Puoi accedere all'elenco dei preferiti dal riquadro di navigazione. La procedura seguente descrive come aggiungere un pannello di controllo all'elenco dei preferiti.

Per aggiungere un pannello di controllo all'elenco dei preferiti

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Dall'elenco dei pannelli di controllo, seleziona il simbolo della stella accanto al nome del pannello di controllo che desideri aggiungere ai preferiti.
   - (Facoltativo) Puoi aggiungere ai preferiti un pannello di controllo anche selezionandolo dall'elenco e scegliendo il simbolo della stella accanto al suo nome.
4. Per accedere all'elenco dei preferiti, scegli Favorites and recents (Preferiti e recenti) nel riquadro di navigazione. Il menu contiene due colonne: una contiene i pannelli di controllo, gli allarmi e i gruppi di flussi di log preferiti, mentre l'altra contiene i pannelli di controllo, gli allarmi e i gruppi di flussi di log visitati di recente.

> **Tip**
>
> Puoi aggiungere dashboard ai preferiti, così come allarmi e gruppi di log, dal menu Preferiti e recenti nel pannello di navigazione della CloudWatch console. Nella colonna Recently visited (Visitati di recente), passa il mouse sul gruppo di flussi di log che desideri aggiungere ai preferiti e scegli il simbolo della stella accanto al suo nome.

Modifica l'impostazione di sostituzione del periodo o l'intervallo di aggiornamento per la dashboard CloudWatch

Puoi specificare il modo in cui le impostazioni del periodo dei grafici aggiunto al pannello di controllo vengono conservate o modificate.

Quando a un widget viene applicato un periodo automatico o un intervallo di tempo persistente, l'intervallo di tempo complessivo del grafico può influire sui periodi impostati.

- Se l'intervallo di tempo è pari o inferiore a un giorno, le impostazioni del periodo non vengono modificate.
- Se l'intervallo di tempo è compreso tra uno e tre giorni, i periodi impostati su un valore inferiore a cinque minuti vengono modificati in 5 minuti.
- Se l'intervallo di tempo è superiore a tre giorni, i periodi impostati su un valore inferiore a un'ora vengono modificati in un'ora.

Nei passaggi seguenti viene descritto come utilizzare la console per modificare le opzioni di sostituzione periodo. Puoi anche modificarle utilizzando il campo `period0verride` nella struttura JSON del pannello di controllo. Per ulteriori informazioni, consulta la pagina relativa alla struttura generale del pannello di controllo.

Per modificare le opzioni di sostituzione periodo

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch
2. Scegli Actions (Azioni).
3. In Period override (Sostituzione periodo), scegli una delle opzioni seguenti:

   - Scegli Auto per fare in modo che il periodo dei parametri di ciascun grafico si adatti automaticamente all'intervallo di tempo del pannello di controllo.
   - Scegli Do not override (Non sostituire) per assicurarti che l'impostazione del periodo di ciascun grafico venga sempre rispettata.
   - Scegli una delle altre opzioni per fare in modo che i grafici aggiunti al pannello di controllo adattino sempre l'ora specificata come impostazione del periodo.

   L'opzione Period override (Sostituzione periodo) torna sempre all'impostazione Auto se il pannello di controllo è chiuso o se il browser viene aggiornato. Non è possibile salvare le varie impostazioni di Period override (Sostituzione periodo).

Puoi modificare la frequenza di aggiornamento dei dati sulla CloudWatch dashboard.

Modifica dell'intervallo di aggiornamento del pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.

3. Nel menu Refresh options (Opzioni di aggiornamento) nell'angolo in alto a destra, seleziona 10 seconds (10 secondi), 1 minute (1 minuto), 2 minutes (2 minuti), 5 minutes (5 minuti) o 15 minutes (15 minuti).

Modifica l'intervallo di tempo o il formato del fuso orario di un CloudWatch pannello di controllo

Puoi modificare l'intervallo di tempo per visualizzare i dati nel pannello di controllo in minuti, ore, giorni o settimane. Puoi anche modificare il formato di data e ora per visualizzare i dati del pannello di controllo in UTC o nel fuso orario locale. L'ora locale è il fuso orario specificato nel sistema operativo del computer.

> **Note**
>
> Se crei un pannello di controllo con grafici che contengono intorno a 100 o più parametri ad alta risoluzione, ti consigliamo di impostare l'intervallo di tempo per non più di un'ora. Per ulteriori informazioni, consulta Parametri ad alta risoluzione.

> **Note**
>
> Se l'intervallo di tempo di un pannello di controllo è più breve del periodo utilizzato per un widget sul pannello di controllo, accade quanto segue:
>
> - Il widget viene modificato per visualizzare la quantità di dati corrispondente a un periodo completo per quel widget, anche se questo è più lungo dell'intervallo di tempo del pannello di controllo. Ciò garantisce che nel grafico sia presente almeno un punto dati.
> - L'ora di inizio del periodo per questo punto dati viene regolata all'indietro per garantire che sia possibile visualizzare almeno un punto dati.

New console

Modifica dell'intervallo di tempo del pannello di controllo

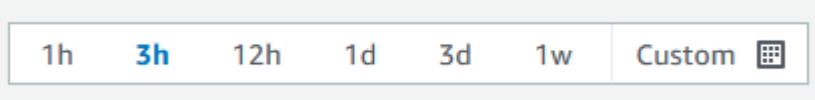
1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Dalla schermata del pannello di controllo, procedi in uno dei seguenti modi:
   - Nell'area superiore del pannello di controllo, seleziona uno degli intervalli di tempo predefiniti. Questi intervalli vanno da 1 ora a 1 settimana (1h, 3h, 12h, 1d, oppure 1w).
   - In alternativa, puoi scegliere una delle seguenti opzioni di intervallo temporale personalizzato:
     - Scegli Personalizza e quindi seleziona la scheda Relativo. Scegli un intervallo di tempo da 1 minuto a 15 mesi.
     - Seleziona Personalizza, quindi seleziona la scheda Assoluto. Utilizza la selezione calendario o i campi di testo per specificare l'intervallo di tempo.

> **Tip**
>
> Se il periodo di aggregazione è impostato su Automatico quando si modifica l'intervallo di tempo di un grafico, CloudWatch potrebbe modificare il periodo. Per impostare il periodo manualmente, scegli il menu a discesa Operazioni, quindi scegli Sostituzione del periodo.

Per modificare il formato di data e ora del pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Nell'area in alto nel pannello di controllo, scegli Personalizza.



4. Nell'angolo in alto a destra della casella che viene visualizzata, dal menu a discesa seleziona UTC o Local time (Ora locale).
5. Scegli Applica.

Old console

Modifica dell'intervallo di tempo del pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Dalla schermata del pannello di controllo, procedi in uno dei seguenti modi:
   - Nell'area superiore del pannello di controllo, seleziona uno degli intervalli di tempo predefiniti. Questi intervalli vanno da 1 ora a 1 settimana (1h,3h,12h,1d,3d, oppure 1w).
   - In alternativa, puoi scegliere una delle seguenti opzioni di intervallo temporale personalizzato:
     - Seleziona il menu a discesa personalizzato, quindi scegli la scheda Relativo. Seleziona uno degli intervalli predefiniti, che partono da 1 minuto fino a 15 mesi.
     - Seleziona il menu a discesa personalizzato, quindi scegli la scheda Assoluto. Utilizza la selezione calendario o i campi di testo per specificare l'intervallo di tempo.

> **Tip**
>
> Se il periodo di aggregazione è impostato su Automatico quando si modifica l'intervallo di tempo di un grafico, CloudWatch potrebbe modificare il periodo. Per impostare il periodo manualmente, scegli il menu a discesa Operazioni, quindi scegli Sostituzione del periodo.

Per modificare il formato di data e ora del pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Dashboards (Pannelli di controllo), quindi scegli un pannello di controllo.
3. Nell'angolo in alto a destra dello schermo del pannello di controllo, scegli il menu a discesa Custom (Personalizzato).
4. Nell'angolo in alto a destra della casella che viene visualizzata, dal menu a discesa seleziona UTC o Local time (Ora locale).

Usa i CloudWatch parametri di Amazon

I parametri sono dati riguardanti le prestazioni dei sistemi. Per impostazione predefinita, molti servizi offrono parametri gratuiti per le risorse (ad esempio istanze Amazon EC2, volumi Amazon EBS e istanze database Amazon RDS). Puoi inoltre abilitare il monitoraggio dettagliato di alcune risorse, ad esempio le istanze Amazon EC2 o pubblicare i tuoi parametri relativi alle applicazioni. Amazon CloudWatch può caricare tutte le metriche del tuo account (sia le metriche AWS delle risorse che le metriche delle applicazioni fornite) per la ricerca, la creazione di grafici e gli allarmi.

I dati metrici vengono conservati per 15 mesi, consentendoti di visualizzare sia i dati che i dati storici. up-to-the-minute

Per rappresentare graficamente le metriche nella console, puoi utilizzare CloudWatch Metrics Insights, un motore di query SQL ad alte prestazioni che puoi utilizzare per identificare tendenze e modelli all'interno di tutte le tue metriche in tempo reale.

Indice

Monitoraggio di base e monitoraggio dettagliato

CloudWatch fornisce due categorie di monitoraggio: monitoraggio di base e monitoraggio dettagliato.

Molti AWS servizi offrono un monitoraggio di base pubblicando gratuitamente un set di metriche predefinito per i clienti. CloudWatch Per impostazione predefinita, quando inizi a utilizzare uno di

questi Servizi AWS, il monitoraggio di base viene abilitato automaticamente. Per un elenco dei servizi che offrono un monitoraggio di base, consulta AWS servizi che pubblicano CloudWatch metriche.

Il monitoraggio dettagliato è offerto solo da alcuni servizi. Si tratta anche di un piano di carico. Per utilizzarlo per un AWS servizio, è necessario scegliere di attivarlo. Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Le opzioni di monitoraggio dettagliate differiscono in base ai servizi che le offrono. Ad esempio, il monitoraggio dettagliato di Amazon EC2 fornisce parametri più frequenti, pubblicati a intervalli di un minuto, anziché gli intervalli di cinque minuti utilizzati nel monitoraggio base di Amazon EC2. Monitoraggio dettagliato per Simple Storage Service (Amazon S3) e Amazon Managed Streaming for Apache Kafka significa metriche più precise.

In diversi AWS servizi, il monitoraggio dettagliato ha anche nomi diversi. Ad esempio, in Amazon EC2 si chiama monitoraggio dettagliato, in AWS Elastic Beanstalk monitoraggio avanzato e in Amazon S3 si chiama parametro di richiesta.

L'utilizzo di un monitoraggio dettagliato per Amazon EC2 aiuta a gestire meglio le risorse Amazon EC2, in modo da poter trovare tendenze e intervenire più rapidamente. Per Simple Storage Service (Amazon S3), i parametri di richiesta sono disponibili a intervalli di un minuto per aiutare a identificare rapidamente i problemi operativi e identificare rapidamente i problemi operativi. Su Amazon MSK, quando si abilita il monitoraggio del livello `PER_BROKER`, `PER_TOPIC_PER_BROKER`, oppure`PER_TOPIC_PER_PARTITION`, si ottengono ulteriori parametri che forniscono maggiore visibilità.

Nella tabella seguente sono elencati i servizi che offrono un monitoraggio dettagliato. Include inoltre collegamenti alla documentazione per quei servizi che spiegano di più sul monitoraggio dettagliato e forniscono istruzioni su come attivarlo.

Servizio	Documentazione
Amazon API Gateway	Dimensioni per i parametri di API Gateway
Amazon CloudFront	Visualizzazione di metriche CloudFront di

Servizio	Documentazione
	distribuzione aggiuntive
Amazon EC2	Abilitazione o disabilitazione del monitoraggio dettagliato per le istanze
Elastic Beanstalk	Monitoraggio e reporting dello stato avanzato
Flusso di dati Amazon Kinesis	Parametri avanzati a livello di partizione
MSK Amazon	Parametri di Amazon MSK per il monitoraggio con CloudWatch
Amazon S3	Parametri delle richieste Amazon S3 in CloudWatch

Servizio	Documentazione
Amazon SES	Raccogli metriche di monitoraggio CloudWatch dettagliate utilizzando Amazon SES Event Publishing.

Inoltre, CloudWatch offre soluzioni di out-of-the-box monitoraggio con metriche più dettagliate e dashboard preimpostati per alcuni AWS servizi, come mostrato nella tabella seguente.

Servizio	Documentazione sulle funzionalità
Lambda	Lambda Insight
Amazon ECS	Container Insights per Amazon ECS
Amazon EKS	Container Insights per Amazon EKS e Kubernetes

Interroga le tue metriche con Metrics Insights CloudWatch

CloudWatch Metrics Insights è un potente motore di query SQL ad alte prestazioni che puoi utilizzare per interrogare i tuoi parametri su larga scala. Puoi identificare tendenze e modelli all'interno di tutte le tue CloudWatch metriche in tempo reale.

Puoi inoltre impostare allarmi su qualsiasi query di Approfondimenti sulle metriche che restituisce una singola serie temporale. Ciò può essere particolarmente utile per creare allarmi che controllano le metriche aggregate su un parco istanze di infrastrutture o applicazioni. Crea l'allarme una volta e si regola dinamicamente man mano che le risorse vengono aggiunte o rimosse dal parco istanze.

Puoi eseguire una query di CloudWatch Metrics Insights nella console con l'editor di query CloudWatch Metrics Insights. Puoi anche eseguire una query di CloudWatch Metrics Insights con AWS CLI o un AWS SDK eseguendo o. `GetMetricData` `PutDashboard` Le query eseguite con l'editor di query CloudWatch Metrics Insights sono gratuite. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Con l'editor di query CloudWatch Metrics Insights, puoi scegliere tra una varietà di query di esempio predefinite e anche creare query personalizzate. Mentre crei le tue query, puoi utilizzare una vista builder per sfogliare i parametri e le dimensioni esistenti. In alternativa, puoi utilizzare una vista editor per scrivere manualmente le query.

Puoi anche utilizzare il linguaggio naturale per creare CloudWatch query Metrics Insights. Per farlo, descrivi o fai domande sui dati che stai cercando. Questa funzionalità assistita dall'intelligenza artificiale genera una query in base alla richiesta dell'utente e fornisce una line-by-line spiegazione di come funziona la query. Per ulteriori informazioni, consulta Utilizzare il linguaggio naturale per generare e aggiornare CloudWatch le query di Metrics Insights.

Con Metrics Insights puoi eseguire query su larga scala. Usando la clausola GROUP BY è possibile raggruppare i parametri in tempo reale in serie temporali separate per valore di dimensione specifico. Poiché le query di Metrics Insights includono la funzionalità ORDER BY, puoi utilizzare Metrics Insights per creare query di tipo "Top N". Ad esempio, le query di tipo "Top N" possono scansionare milioni di parametri nell'account e restituire le 10 istanze che consumano più CPU. Questo può aiutarti a individuare e risolvere i problemi di latenza delle applicazioni.

Argomenti

- Limiti di Metrics Insights
- Glossario di Metrics Insights
- Risoluzione dei problemi relativi Metrics Insights

Creazione di query

Puoi eseguire una query di CloudWatch Metrics Insights utilizzando la CloudWatch console AWS CLI, gli o gli AWS SDK. Le query eseguite nella console sono gratuite. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Per ulteriori informazioni sull'utilizzo degli AWS SDK per eseguire una query di Metrics Insights, consulta. GetMetricData

Per eseguire una query utilizzando la CloudWatch console, procedi nel seguente modo:

Effettuare una query dei parametri utilizzando Metrics Insights

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), All metrics (Tutti i parametri).
3. Scegli la scheda Queries (Query).
4. (Facoltativo) Per eseguire una query di esempio precompilata, scegli Aggiungi query e seleziona la query da eseguire. Se si è soddisfatti di questa query, è possibile ignorare il resto di questa procedura. Oppure, è possibile scegliere Editor per modificare la query di esempio e quindi scegliere Run (Esegui) per eseguire la query modificata.
5. Per creare una propria query, è possibile utilizzare la vista Builder, la vista Editor e anche utilizzare una combinazione di entrambe. È possibile passare da una visualizzazione all'altra in qualsiasi momento e visualizzare i lavori in corso in entrambe le viste.

   Nella vista Builder, è possibile sfogliare e selezionare lo spazio dei nomi dei parametri, il nome del parametro, il filtro, il gruppo e le opzioni dell'ordine. Per ognuna di queste opzioni, il generatore di query offre un elenco di possibili selezioni dal tuo ambiente tra cui scegliere.

   Nella vista Editor, è possibile iniziare a scrivere la tua query. Durante la digitazione, l'editor offre suggerimenti basati sui caratteri digitati finora.
6. Quando si è soddisfatti della query, scegli Run (Avvia).

7. (Facoltativo) Un altro modo per modificare una query creata graficamente è scegli la scheda Graphed metrics (Rappresentazione grafica di parametri) e scegli l'icona di modifica accanto alla formula di query nella colonna Details (Dettagli).
8. (Facoltativo) Per rimuovere una query dal grafico, scegli Graphed metrics (Rappresentazione grafica di parametri) e scegli l'icona X sul lato destro della riga che visualizza la query.

Componenti e sintassi della query di Metrics Insights

CloudWatch La sintassi di Metrics Insights è la seguente.

```
SELECT FUNCTION(metricName)
FROM namespace | SCHEMA(...)
[ WHERE labelKey OPERATOR labelValue [AND ... ] ]
[ GROUP BY labelKey [ , ... ] ]
[ ORDER BY FUNCTION() [ DESC | ASC ] ]
[ LIMIT number ]
```

Le possibili clausole in una query Metrics Insights sono le seguenti. Nessuna delle parole chiave è sensibile alle maiuscole e minuscole, ma gli identificatori come i nomi dei parametri, gli spazi dei nomi e le dimensioni sono sensibili alle maiuscole e minuscole.

SELECT

Obbligatorio. Specifica la funzione da utilizzare per aggregare le osservazioni in ogni bucket temporale (determinato dal periodo fornito). Specifica inoltre il nome del parametro su cui effettuare la query.

I valori validi per FUNCTION (FUNZIONE) sono `AVG`, `COUNT`, `MAX`, `MIN`, e `SUM`.

- `AVG` calcola la media delle osservazioni corrispondenti alla query.
- `COUNT` restituisce il conteggio delle osservazioni corrispondenti alla query.
- `MAX` restituisce il valore massimo delle osservazioni corrispondenti alla query.
- `MIN` restituisce il valore minimo delle osservazioni corrispondenti alla query.
- `SUM` calcola la somma delle osservazioni corrispondenti alla query.

FROM

Obbligatorio. Specifica la fonte del parametro. È possibile specificare lo spazio dei nomi del parametro che contiene il parametro che deve essere sottoposto a query o una funzione di tabella

SCHEMA. Esempi di spazio dei nomi di parametro includono `"AWS/EC2"`, `"AWS/Lambda"` e gli spazi dei nomi dei parametri che hai creato per i parametri personalizzati.

Gli spazi dei nomi del parametro che includono / o qualsiasi altro carattere che non sia una lettera, un numero o un carattere di sottolineatura devono essere racchiusi tra virgolette doppie. Per ulteriori informazioni, consulta Cosa ha bisogno di virgolette o caratteri di escape?.

SCHEMA

Una funzione di tabella opzionale che può essere utilizzata all'interno di una clausola FROM (DA). Utilizza SCHEMA per esaminare i risultati della query solo sui parametri che corrispondono esattamente a un elenco di dimensioni o ai parametri che non hanno dimensioni.

Se si utilizza una clausola SCHEMA, deve contenere almeno un argomento e questo primo argomento deve essere lo spazio dei nomi dei parametri da sottoporre a query. Se si specifica SCHEMA solo con questo argomento dello spazio dei nomi, i risultati vengono assegnati solo a parametri che non hanno dimensioni.

Se si specifica SCHEMA con argomenti aggiuntivi, gli argomenti aggiuntivi dopo l'argomento dello spazio dei nomi devono essere chiavi di etichetta. Le chiavi di etichetta devono essere nomi di dimensione. Se si specificano una o più di queste chiavi di etichetta, i risultati vengono assegnati solo ai parametri che hanno quel set esatto di dimensioni. L'ordine di queste chiavi di etichetta non è importante.

Ad esempio:

- SELECT AVG(CPUUtilization) FROM "AWS/EC2" (SELEZIONA AVG (Utilizzo della CPU) da “AWS/EC2")) corrisponde a tutti i parametri `CPUUtilization` nello spazio dei nomi `AWS/EC2`, indipendentemente dalle loro dimensioni, e restituisce una singola serie temporale aggregata.
- SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2") (SELEZIONA AVG (Utilizzo della CPU) DALLO SCHEMA ("AWS/EC2"))corrisponde solo ai parametri `CPUUtilization` nello spazio dei nomi `AWS/EC2` che non presentano alcuna dimensione definita.
- SELECT AVG (CPUUtilization) FROM SCHEMA («AWS/EC2", InstanceId) corrisponde solo alle `CPUUtilization` metriche riportate con esattamente una dimensione,. CloudWatch `InstanceId`
- SELECT SUM (RequestCount) FROM SCHEMA («AWS/ApplicationElb» LoadBalancer, AvailabilityZone) corrisponde solo alle `RequestCount` metriche riportate CloudWatch da from `AWS/ApplicationELB` con esattamente due dimensioni e. `LoadBalancer` `AvailabilityZone`

WHERE

Facoltativo. Filtra i risultati solo con i parametri che corrispondono all'espressione specificata utilizzando valori di etichetta specifici per una o più chiavi di etichetta. Ad esempio, WHERE InstanceType = 'c3.4xlarge' filtra i risultati solo in base ai tipi di istanza e WHERE! **`c3.4xlarge`** InstanceType = 'c3.4xlarge' filtra i risultati in base a tutti i tipi di istanze tranne. `c3.4xlarge`

Quando esegui una query in un account di monitoraggio, puoi utilizzare `WHERE AWS.AccountId` per limitare i risultati solo all'account specificato. Ad esempio, `WHERE AWS.AccountId=444455556666` interroga i parametri solo dall'account `444455556666`. Per limitare la tua query ai soli parametri dell'account di monitoraggio stesso, usa `WHERE AWS.AccountId=CURRENT_ACCOUNT_ID()`.

I valori delle etichette devono sempre essere racchiusi con virgolette singole.

Operatori supportati

La clausola WHERE (DOVE) supporta i seguenti operatori:

- = Il valore dell'etichetta deve corrispondere alla stringa specificata.
- ! = Il valore dell'etichetta non deve corrispondere alla stringa specificata.
- E Entrambe le condizioni specificate devono essere vere per corrispondere. È possibile utilizzare più parole chiave AND (E) per specificare due o più condizioni.

GROUP BY

Facoltativo. Raggruppa i risultati della query in più serie temporali, ognuna corrispondente a un valore diverso per la chiave o le chiavi dell'etichetta specificate. Ad esempio, l'utilizzo di `GROUP BY InstanceId` restituisce una serie temporale diversa per ogni valore di `InstanceId`. L'utilizzo di `GROUP BY ServiceName, Operation` crea una serie temporale diversa per ogni possibile combinazione dei valori di `ServiceName` e `Operation`.

Con una clausola GROUP BY (RAGGRUPPA PER), per impostazione predefinita, i risultati sono ordinati in ordine alfabetico crescente, utilizzando la sequenza di etichette specificata nella clausola GROUP BY (RAGGRUPPA PER). Per modificare l'ordine dei risultati, aggiungere una clausola ODER BY (ORDINA PER) alla tua query.

Quando esegui una query in un account di monitoraggio, puoi utilizzare `GROUP BY AWS.AccountId` per raggruppare i risultati in base agli account da cui provengono.

> **Note**
>
> Se alcune delle metriche corrispondenti non includono una chiave di etichetta specifica specificata nella clausola GROUP BY (RAGGRUPPA PER), viene restituito un gruppo nullo denominato `Other`. Ad esempio, se si specifica `GROUP BY ServiceName, Operation` e alcuni dei parametri restituiti non includono`ServiceName` come dimensione, quindi tali parametri vengono visualizzati come aventi `Other` come valore per `ServiceName`.

ORDER BY

Facoltativo. Specifica l'ordine da utilizzare per le serie temporali restituite, se la query restituisce più di una serie temporale. L'ordine è basato sui valori trovati dalla FUNCTION (FUNZIONE) specificata nella clausola ORDER BY (ORDINA PER). La FUNCTION (FUNZIONE) viene utilizzata per calcolare un singolo valore scalare da ogni serie temporale restituita e tale valore viene utilizzato per determinare l'ordine.

È inoltre necessario specificare se utilizzare l'ordine crescente ASC o decrescente DESC. Se ometti questo parametro, il valore di default è crescente ASC.

Ad esempio, aggiungendo una clausola `ORDER BY MAX() DESC` ordina i risultati in base al punto dati massimo osservato nell'intervallo di tempo, in ordine decrescente: il che significa che la serie temporale che ha il punto dati massimo viene restituita per prima.

Le funzioni valide da utilizzare all'interno di una clausola ORDER BY (ORDINA PER) sono `AVG()`, `COUNT()`, `MAX()`, `MIN()` e `SUM()`.

Se utilizzi una clausola ORDER BY (ORDINA PER) con una clausola LIMIT (LIMITE), la query risultante è una query "Top N". ORDER BY (ORDINA PER) è utile anche per le query che potrebbero restituire un numero elevato di metriche, poiché ogni query può restituire non più di 500 serie temporali. Se una query corrisponde a più di 500 serie temporali e si utilizza una clausola ORDER BY (ORDINA PER), le serie temporali vengono ordinate e quindi le 500 serie temporali che vengono prima nell'ordinamento sono quelle che vengono restituite.

LIMIT

Facoltativo. Limita il numero di serie temporali restituite dalla query al valore specificato. Il valore massimo che è possibile specificare è 500 e una query che non specifica un LIMIT (LIMITE) può anche restituire non più di 500 serie temporali.

L'utilizzo di una clausola LIMIT (LIMITE) con una clausola ORDER BY (ORDINA PER) fornisce una query "Top N".

Cosa ha bisogno di virgolette o caratteri di escape?

In una query, i valori delle etichette devono sempre essere racchiusi tra virgolette singole. Ad esempio, SELECT MAX (CPUUtilization) FROM «AWS/EC2" WHERE = ''. AutoScalingGroupName my-production-fleet

Gli spazi dei nomi dei parametri, i nomi dei paramentri e le chiavi di etichetta contenenti caratteri diversi da lettere, numeri e trattino di sottolineatura (_) devono essere racchiusi tra virgolette doppie. Ad esempio: SELECT MAX ("My.Metric").

Se uno di questi contiene una doppia virgoletta o una virgoletta singola (come `Bytes"Input"`), è necessario far precedere ogni virgoletta da una barra rovesciata, come in SELECT AVG("Bytes\"Input \"").

Se uno spazio dei nomi del parametro, un nome del parametro o una chiave etichetta contiene una parola che è una parola chiave riservata in Metrics Insights, queste devono anche essere racchiuse tra virgolette doppie. Ad esempio, se si dispone di un parametro denominato `LIMIT`, si utilizza `SELECT AVG("LIMIT")`. È inoltre valido racchiudere qualsiasi spazio dei nomi, nome del parametro o etichetta tra virgolette doppie anche se non include una parola chiave riservata.

Per un elenco completo delle parole chiave riservate, consulta Parole chiave riservate.

Creazione dettagliata di una query approfondita

Questa sezione illustra la creazione di un esempio completo che utilizza tutte le possibili clausole, in maniera dettagliata.

Iniziamo con la seguente query, che aggrega tutti i parametri di Application Load Balancer `RequestCount` raccolti con entrambe le dimensioni `LoadBalancer` e `AvailabilityZone`.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
```

Ora, se vogliamo vedere le metriche solo da uno specifico load balancer, possiamo aggiungere una clausola WHERE (DOVE) per limitare i parametri restituiti solo ai parametri in cui il valore della dimensione `LoadBalancer` è `app/load-balancer-1`.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
```

La query precedente aggrega i parametri `RequestCount` provenienti da tutte le zone di disponibilità per questo load balancer in una serie temporale. Se vogliamo vedere diverse serie temporali per ciascuna zona di disponibilità, possiamo aggiungere una clausola GROUP BY (RAGGRUPPA PER).

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
```

Successivamente, potremmo ordinare questi risultati per vedere prima i valori più alti. La seguente clausola ORDER BY (ORDINA PER) ordina le serie temporali in ordine decrescente, in base al valore massimo riportato da ogni serie temporale durante l'intervallo di tempo della query:

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
```

Infine, se siamo principalmente interessati a un tipo di query "Top N", possiamo utilizzare una clausola LIMIT (LIMITE). Questo esempio finale limita i risultati solo alle serie temporali con i cinque valori `MAX` più alti.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
LIMIT 5
```

Esempi di query tra account

Questi esempi sono validi se eseguiti in un account configurato come account di monitoraggio nell'osservabilità tra account. CloudWatch

Il seguente esempio cerca tutte le istanze Amazon EC2 nell'account di origine 123456789012 e restituisce la media.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId ='123456789012'
```

L'esempio seguente interroga il parametro `CPUUtilization` in `AWS/EC2` in tutti gli account di origine collegati e raggruppa i risultati per ID account e tipo di istanza.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
GROUP BY AWS.AccountId, InstanceType
```

L'esempio seguente interroga `CPUUtilization` nell'account di monitoraggio stesso.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId = CURRENT_ACCOUNT_ID()
```

Parole chiave riservate

Le seguenti sono parole chiave riservate in CloudWatch Metrics Insights. Se una di queste parole si trova in uno spazio dei nomi, un nome parametro o una chiave di etichetta in una query, è necessario racchiuderle tra virgolette doppie. Le parole chiave riservate non distinguono tra maiuscole

```
"ABORT" "ABORTSESSION" "ABS" "ABSOLUTE" "ACCESS" "ACCESSIBLE" "ACCESS_LOCK" "ACCOUNT"
 "ACOS" "ACOSH" "ACTION" "ADD" "ADD_MONTHS"
"ADMIN" "AFTER" "AGGREGATE" "ALIAS" "ALL" "ALLOCATE" "ALLOW" "ALTER" "ALTERAND" "AMP"
 "ANALYSE" "ANALYZE" "AND" "ANSIDATE" "ANY" "ARE" "ARRAY",
"ARRAY_AGG" "ARRAY_EXISTS" "ARRAY_MAX_CARDINALITY" "AS" "ASC" "ASENSITIVE" "ASIN"
 "ASINH" "ASSERTION" "ASSOCIATE" "ASUTIME" "ASYMMETRIC" "AT",
"ATAN" "ATAN2" "ATANH" "ATOMIC" "AUDIT" "AUTHORIZATION" "AUX" "AUXILIARY" "AVE"
 "AVERAGE" "AVG" "BACKUP" "BEFORE" "BEGIN" "BEGIN_FRAME" "BEGIN_PARTITION",
"BETWEEN" "BIGINT" "BINARY" "BIT" "BLOB" "BOOLEAN" "BOTH" "BREADTH" "BREAK" "BROWSE"
 "BT" "BUFFERPOOL" "BULK" "BUT" "BY" "BYTE" "BYTEINT" "BYTES" "CALL",
"CALLED" "CAPTURE" "CARDINALITY" "CASCADE" "CASCADED" "CASE" "CASESPECIFIC" "CASE_N"
 "CAST" "CATALOG" "CCSID" "CD" "CEIL" "CEILING" "CHANGE" "CHAR",
"CHAR2HEXINT" "CHARACTER" "CHARACTERS" "CHARACTER_LENGTH" "CHARS" "CHAR_LENGTH" "CHECK"
 "CHECKPOINT" "CLASS" "CLASSIFIER" "CLOB" "CLONE" "CLOSE" "CLUSTER",
```

```
"CLUSTERED" "CM" "COALESCE" "COLLATE" "COLLATION" "COLLECT" "COLLECTION" "COLLID"
 "COLUMN" "COLUMN_VALUE" "COMMENT" "COMMIT" "COMPLETION" "COMPRESS" "COMPUTE",
"CONCAT" "CONCURRENTLY" "CONDITION" "CONNECT" "CONNECTION" "CONSTRAINT" "CONSTRAINTS"
 "CONSTRUCTOR" "CONTAINS" "CONTAINSTABLE" "CONTENT" "CONTINUE" "CONVERT",
"CONVERT_TABLE_HEADER" "COPY" "CORR" "CORRESPONDING" "COS" "COSH" "COUNT" "COVAR_POP"
 "COVAR_SAMP" "CREATE" "CROSS" "CS" "CSUM" "CT" "CUBE" "CUME_DIST",
"CURRENT" "CURRENT_CATALOG" "CURRENT_DATE" "CURRENT_DEFAULT_TRANSFORM_GROUP"
 "CURRENT_LC_CTYPE" "CURRENT_PATH" "CURRENT_ROLE" "CURRENT_ROW" "CURRENT_SCHEMA",
"CURRENT_SERVER" "CURRENT_TIME" "CURRENT_TIMESTAMP" "CURRENT_TIMEZONE"
 "CURRENT_TRANSFORM_GROUP_FOR_TYPE" "CURRENT_USER" "CURRVAL" "CURSOR" "CV" "CYCLE"
 "DATA",
"DATABASE" "DATABASES" "DATABLOCKSIZE" "DATE" "DATEFORM" "DAY" "DAYS" "DAY_HOUR"
 "DAY_MICROSECOND" "DAY_MINUTE" "DAY_SECOND" "DBCC" "DBINFO" "DEALLOCATE" "DEC",
"DECFLOAT" "DECIMAL" "DECLARE" "DEFAULT" "DEFERRABLE" "DEFERRED" "DEFINE" "DEGREES"
 "DEL" "DELAYED" "DELETE" "DENSE_RANK" "DENY" "DEPTH" "DEREF" "DESC" "DESCRIBE",
"DESCRIPTOR" "DESTROY" "DESTRUCTOR" "DETERMINISTIC" "DIAGNOSTIC" "DIAGNOSTICS"
 "DICTIONARY" "DISABLE" "DISABLED" "DISALLOW" "DISCONNECT" "DISK" "DISTINCT",
"DISTINCTROW" "DISTRIBUTED" "DIV" "DO" "DOCUMENT" "DOMAIN" "DOUBLE" "DROP" "DSSIZE"
 "DUAL" "DUMP" "DYNAMIC" "EACH" "ECHO" "EDITPROC" "ELEMENT" "ELSE" "ELSEIF",
"EMPTY" "ENABLED" "ENCLOSED" "ENCODING" "ENCRYPTION" "END" "END-EXEC" "ENDING"
 "END_FRAME" "END_PARTITION" "EQ" "EQUALS" "ERASE" "ERRLVL" "ERROR" "ERRORFILES",
"ERRORTABLES" "ESCAPE" "ESCAPED" "ET" "EVERY" "EXCEPT" "EXCEPTION" "EXCLUSIVE" "EXEC"
 "EXECUTE" "EXISTS" "EXIT" "EXP" "EXPLAIN" "EXTERNAL" "EXTRACT" "FALLBACK
"FALSE" "FASTEXPORT" "FENCED" "FETCH" "FIELDPROC" "FILE" "FILLFACTOR" "FILTER" "FINAL"
 "FIRST" "FIRST_VALUE" "FLOAT" "FLOAT4" "FLOAT8" "FLOOR"
"FOR" "FORCE" "FOREIGN" "FORMAT" "FOUND" "FRAME_ROW" "FREE" "FREESPACE" "FREETEXT"
 "FREETEXTTABLE" "FREEZE" "FROM" "FULL" "FULLTEXT" "FUNCTION"
"FUSION" "GE" "GENERAL" "GENERATED" "GET" "GIVE" "GLOBAL" "GO" "GOTO" "GRANT" "GRAPHIC"
 "GROUP" "GROUPING" "GROUPS" "GT" "HANDLER" "HASH"
"HASHAMP" "HASHBAKAMP" "HASHBUCKET" "HASHROW" "HAVING" "HELP" "HIGH_PRIORITY" "HOLD"
 "HOLDLOCK" "HOUR" "HOURS" "HOUR_MICROSECOND" "HOUR_MINUTE"
"HOUR_SECOND" "IDENTIFIED" "IDENTITY" "IDENTITYCOL" "IDENTITY_INSERT" "IF" "IGNORE"
 "ILIKE" "IMMEDIATE" "IN" "INCLUSIVE" "INCONSISTENT" "INCREMENT"
"INDEX" "INDICATOR" "INFILE" "INHERIT" "INITIAL" "INITIALIZE" "INITIALLY" "INITIATE"
 "INNER" "INOUT" "INPUT" "INS" "INSENSITIVE" "INSERT" "INSTEAD"
"INT" "INT1" "INT2" "INT3" "INT4" "INT8" "INTEGER" "INTEGERDATE" "INTERSECT"
 "INTERSECTION" "INTERVAL" "INTO" "IO_AFTER_GTIDS" "IO_BEFORE_GTIDS"
"IS" "ISNULL" "ISOBID" "ISOLATION" "ITERATE" "JAR" "JOIN" "JOURNAL" "JSON_ARRAY"
 "JSON_ARRAYAGG" "JSON_EXISTS" "JSON_OBJECT" "JSON_OBJECTAGG"
"JSON_QUERY" "JSON_TABLE" "JSON_TABLE_PRIMITIVE" "JSON_VALUE" "KEEP" "KEY" "KEYS"
 "KILL" "KURTOSIS" "LABEL" "LAG" "LANGUAGE" "LARGE" "LAST"
"LAST_VALUE" "LATERAL" "LC_CTYPE" "LE" "LEAD" "LEADING" "LEAVE" "LEFT" "LESS" "LEVEL"
 "LIKE" "LIKE_REGEX" "LIMIT" "LINEAR" "LINENO" "LINES"
```

```
"LISTAGG" "LN" "LOAD" "LOADING" "LOCAL" "LOCALE" "LOCALTIME" "LOCALTIMESTAMP" "LOCATOR"
 "LOCATORS" "LOCK" "LOCKING" "LOCKMAX" "LOCKSIZE" "LOG"
"LOG10" "LOGGING" "LOGON" "LONG" "LONGBLOB" "LONGTEXT" "LOOP" "LOWER" "LOW_PRIORITY"
 "LT" "MACRO" "MAINTAINED" "MAP" "MASTER_BIND"
"MASTER_SSL_VERIFY_SERVER_CERT" "MATCH" "MATCHES" "MATCH_NUMBER" "MATCH_RECOGNIZE"
 "MATERIALIZED" "MAVG" "MAX" "MAXEXTENTS" "MAXIMUM" "MAXVALUE"
"MCHARACTERS" "MDIFF" "MEDIUMBLOB" "MEDIUMINT" "MEDIUMTEXT" "MEMBER" "MERGE" "METHOD"
 "MICROSECOND" "MICROSECONDS" "MIDDLEINT" "MIN" "MINDEX"
"MINIMUM" "MINUS" "MINUTE" "MINUTES" "MINUTE_MICROSECOND" "MINUTE_SECOND" "MLINREG"
 "MLOAD" "MLSLABEL" "MOD" "MODE" "MODIFIES" "MODIFY"
"MODULE" "MONITOR" "MONRESOURCE" "MONSESSION" "MONTH" "MONTHS" "MSUBSTR" "MSUM"
 "MULTISET" "NAMED" "NAMES" "NATIONAL" "NATURAL" "NCHAR" "NCLOB"
"NE" "NESTED_TABLE_ID" "NEW" "NEW_TABLE" "NEXT" "NEXTVAL" "NO" "NOAUDIT" "NOCHECK"
 "NOCOMPRESS" "NONCLUSTERED" "NONE" "NORMALIZE" "NOT" "NOTNULL"
"NOWAIT" "NO_WRITE_TO_BINLOG" "NTH_VALUE" "NTILE" "NULL" "NULLIF" "NULLIFZERO" "NULLS"
 "NUMBER" "NUMERIC" "NUMPARTS" "OBID" "OBJECT" "OBJECTS"
"OCCURRENCES_REGEX" "OCTET_LENGTH" "OF" "OFF" "OFFLINE" "OFFSET" "OFFSETS" "OLD"
 "OLD_TABLE" "OMIT" "ON" "ONE" "ONLINE" "ONLY" "OPEN" "OPENDATASOURCE"
"OPENQUERY" "OPENROWSET" "OPENXML" "OPERATION" "OPTIMIZATION" "OPTIMIZE"
 "OPTIMIZER_COSTS" "OPTION" "OPTIONALLY" "OR" "ORDER" "ORDINALITY" "ORGANIZATION"
"OUT" "OUTER" "OUTFILE" "OUTPUT" "OVER" "OVERLAPS" "OVERLAY" "OVERRIDE" "PACKAGE" "PAD"
 "PADDED" "PARAMETER" "PARAMETERS" "PART" "PARTIAL" "PARTITION"
"PARTITIONED" "PARTITIONING" "PASSWORD" "PATH" "PATTERN" "PCTFREE" "PER" "PERCENT"
 "PERCENTILE" "PERCENTILE_CONT" "PERCENTILE_DISC" "PERCENT_RANK" "PERIOD" "PERM"
"PERMANENT" "PIECESIZE" "PIVOT" "PLACING" "PLAN" "PORTION" "POSITION" "POSITION_REGEX"
 "POSTFIX" "POWER" "PRECEDES" "PRECISION" "PREFIX" "PREORDER"
"PREPARE" "PRESERVE" "PREVVAL" "PRIMARY" "PRINT" "PRIOR" "PRIQTY" "PRIVATE"
 "PRIVILEGES" "PROC" "PROCEDURE" "PROFILE" "PROGRAM" "PROPORTIONAL"
"PROTECTION" "PSID" "PTF" "PUBLIC" "PURGE" "QUALIFIED" "QUALIFY" "QUANTILE" "QUERY"
 "QUERYNO" "RADIANS" "RAISERROR" "RANDOM" "RANGE" "RANGE_N" "RANK"
"RAW" "READ" "READS" "READTEXT" "READ_WRITE" "REAL" "RECONFIGURE" "RECURSIVE" "REF"
 "REFERENCES" "REFERENCING" "REFRESH" "REGEXP" "REGR_AVGX" "REGR_AVGY"
"REGR_COUNT" "REGR_INTERCEPT" "REGR_R2" "REGR_SLOPE" "REGR_SXX" "REGR_SXY" "REGR_SYY"
 "RELATIVE" "RELEASE" "RENAME" "REPEAT" "REPLACE" "REPLICATION"
"REPOVERRIDE" "REQUEST" "REQUIRE" "RESIGNAL" "RESOURCE" "RESTART" "RESTORE" "RESTRICT"
 "RESULT" "RESULT_SET_LOCATOR" "RESUME" "RET" "RETRIEVE" "RETURN"
"RETURNING" "RETURNS" "REVALIDATE" "REVERT" "REVOKE" "RIGHT" "RIGHTS" "RLIKE" "ROLE"
 "ROLLBACK" "ROLLFORWARD" "ROLLUP" "ROUND_CEILING" "ROUND_DOWN"
"ROUND_FLOOR" "ROUND_HALF_DOWN" "ROUND_HALF_EVEN" "ROUND_HALF_UP" "ROUND_UP" "ROUTINE"
 "ROW" "ROWCOUNT" "ROWGUIDCOL" "ROWID" "ROWNUM" "ROWS" "ROWSET"
"ROW_NUMBER" "RULE" "RUN" "RUNNING" "SAMPLE" "SAMPLEID" "SAVE" "SAVEPOINT" "SCHEMA"
 "SCHEMAS" "SCOPE" "SCRATCHPAD" "SCROLL" "SEARCH" "SECOND" "SECONDS"
"SECOND_MICROSECOND" "SECQTY" "SECTION" "SECURITY" "SECURITYAUDIT" "SEEK" "SEL"
 "SELECT" "SEMANTICKEYPHRASETABLE" "SEMANTICSIMILARITYDETAILSTABLE"
```

```
"SEMANTICSIMILARITYTABLE" "SENSITIVE" "SEPARATOR" "SEQUENCE" "SESSION" "SESSION_USER"
 "SET" "SETRESRATE" "SETS" "SETSESSRATE" "SETUSER" "SHARE" "SHOW"
"SHUTDOWN" "SIGNAL" "SIMILAR" "SIMPLE" "SIN" "SINH" "SIZE" "SKEW" "SKIP" "SMALLINT"
 "SOME" "SOUNDEX" "SOURCE" "SPACE" "SPATIAL" "SPECIFIC" "SPECIFICTYPE"
"SPOOL" "SQL" "SQLEXCEPTION" "SQLSTATE" "SQLTEXT" "SQLWARNING" "SQL_BIG_RESULT"
 "SQL_CALC_FOUND_ROWS" "SQL_SMALL_RESULT" "SQRT" "SS" "SSL" "STANDARD"
"START" "STARTING" "STARTUP" "STAT" "STATE" "STATEMENT" "STATIC" "STATISTICS" "STAY"
 "STDDEV_POP" "STDDEV_SAMP" "STEPINFO" "STOGROUP" "STORED" "STORES"
"STRAIGHT_JOIN" "STRING_CS" "STRUCTURE" "STYLE" "SUBMULTISET" "SUBSCRIBER" "SUBSET"
 "SUBSTR" "SUBSTRING" "SUBSTRING_REGEX" "SUCCEEDS" "SUCCESSFUL"
"SUM" "SUMMARY" "SUSPEND" "SYMMETRIC" "SYNONYM" "SYSDATE" "SYSTEM" "SYSTEM_TIME"
 "SYSTEM_USER" "SYSTIMESTAMP" "TABLE" "TABLESAMPLE" "TABLESPACE" "TAN"
"TANH" "TBL_CS" "TEMPORARY" "TERMINATE" "TERMINATED" "TEXTSIZE" "THAN" "THEN"
 "THRESHOLD" "TIME" "TIMESTAMP" "TIMEZONE_HOUR" "TIMEZONE_MINUTE" "TINYBLOB"
"TINYINT" "TINYTEXT" "TITLE" "TO" "TOP" "TRACE" "TRAILING" "TRAN" "TRANSACTION"
 "TRANSLATE" "TRANSLATE_CHK" "TRANSLATE_REGEX" "TRANSLATION" "TREAT"
"TRIGGER" "TRIM" "TRIM_ARRAY" "TRUE" "TRUNCATE" "TRY_CONVERT" "TSEQUAL" "TYPE" "UC"
 "UESCAPE" "UID" "UNDEFINED" "UNDER" "UNDO" "UNION" "UNIQUE"
"UNKNOWN" "UNLOCK" "UNNEST" "UNPIVOT" "UNSIGNED" "UNTIL" "UPD" "UPDATE" "UPDATETEXT"
 "UPPER" "UPPERCASE" "USAGE" "USE" "USER" "USING" "UTC_DATE"
"UTC_TIME" "UTC_TIMESTAMP" "VALIDATE" "VALIDPROC" "VALUE" "VALUES" "VALUE_OF"
 "VARBINARY" "VARBYTE" "VARCHAR" "VARCHAR2" "VARCHARACTER" "VARGRAPHIC"
"VARIABLE" "VARIADIC" "VARIANT" "VARYING" "VAR_POP" "VAR_SAMP" "VCAT" "VERBOSE"
 "VERSIONING" "VIEW" "VIRTUAL" "VOLATILE" "VOLUMES" "WAIT" "WAITFOR"
"WHEN" "WHENEVER" "WHERE" "WHILE" "WIDTH_BUCKET" "WINDOW" "WITH" "WITHIN"
 "WITHIN_GROUP" "WITHOUT" "WLM" "WORK" "WRITE" "WRITETEXT" "XMLCAST" "XMLEXISTS"
"XMLNAMESPACES" "XOR" "YEAR" "YEARS" "YEAR_MONTH" "ZEROFILL" "ZEROIFNULL" "ZONE"
```

Creazione di allarmi nelle query di Approfondimenti sulle metriche

Puoi creare allarmi nelle query di Approfondimenti sulle metriche. In questo modo avrai a disposizione degli allarmi che tracciano molteplici risorse, senza dover essere aggiornati in seguito. La query rileva le nuove risorse e quelle che vengono modificate. Ad esempio, puoi creare un allarme che controlli l'utilizzo della CPU del parco istanze e valuti automaticamente le nuove istanze avviate dopo la sua creazione.

In un account di monitoraggio configurato per l'osservabilità tra più CloudWatch account, gli allarmi di Metrics Insights possono controllare le risorse negli account di origine e nell'account di monitoraggio stesso. Per ulteriori informazioni su come limitare le richieste di allarme a un account specifico o su come raggruppare i risultati in base all'ID dell'account, consulta le sezioni `WHERE` e `GROUP BY` in Componenti e sintassi della query di Metrics Insights.

Indice

Creazione di un allarme di Approfondimenti sulle metriche

Per creare un allarme su una query di Approfondimenti sulle metriche tramite la console

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nel pannello di navigazione, seleziona Metrics (Parametri), All metrics (Tutti i parametri).
3. Scegli la scheda Queries (Query).
4. (Facoltativo) Per eseguire una query di esempio precompilata, scegli Aggiungi query e seleziona la query da eseguire. Oppure, è possibile scegliere Editor per modificare la query di esempio e quindi scegliere Run (Esegui) per eseguire la query modificata.
5. Per creare una query personalizzata, puoi utilizzare la vista Builder, la vista Editor o una combinazione di entrambe. È possibile passare da una visualizzazione all'altra in qualsiasi momento e visualizzare i lavori in corso in entrambe le viste.

   Nella vista Builder, è possibile sfogliare e selezionare lo spazio dei nomi dei parametri, il nome del parametro, il filtro, il gruppo e le opzioni dell'ordine. Per ognuna di queste opzioni, il generatore di query offre un elenco di possibili selezioni dal tuo ambiente tra cui scegliere.

   Nella vista Editor, è possibile iniziare a scrivere la tua query. Durante la digitazione, l'editor offre suggerimenti basati sui caratteri digitati finora.

   > **Important**
   >
   > Per configurare un allarme su una query di Approfondimenti sulle metriche, la query deve restituire una singola serie temporale. Se contiene un'istruzione GROUP BY, quest'ultima deve essere racchiusa in un'espressione matematica del parametro che restituisca solo una serie temporale come risultato finale dell'espressione.

6. Quando si è soddisfatti della query, scegli Run (Avvia).
7. Scegli Crea allarme.
8. In Conditions (Condizioni), specifica quanto segue:

a. Per Whenever ***`metric`*** is (Ogni volta che il parametro è), specifica se il parametro deve essere maggiore di, minore di o uguale alla soglia. In than... (che...), specifica il valore di soglia.

b. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

   Per creare un allarme M di N, specifica un numero minore per il primo valore rispetto a quello specificato per il secondo valore. Per ulteriori informazioni, consulta la pagina Valutazione di un allarme.

c. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .

9. Seleziona Next (Successivo).

10. In Notification (Notifica), seleziona un argomento SNS per segnalare quando l'allarme è nello stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

    Per fare in modo che l'allarme non invii notifiche, scegli Remove (Rimuovi).

11. Per fare in modo che l'allarme esegua operazioni Auto Scaling, EC2 o Systems Manager scegli il pulsante appropriato e scegli lo stato di allarme e l'operazione da eseguire. Gli allarmi possono eseguire le operazioni Systems Manager solo quando entrano nello stato ALARM. Per ulteriori informazioni sulle azioni di Systems Manager, vedere Configurazione per la creazione CloudWatch OpsItems da allarmi e Creazione di incidenti.

    > **Note**
    >
    > Per creare un allarme che esegua un'operazione SSM Incident Manager, è necessario disporre di determinate autorizzazioni. Per ulteriori informazioni, vedere Esempi di policy basate sull'identità per AWS Systems Manager Incident Manager.

12. Al termine, scegli Apply (Applica).

13. Inserisci un nome e una descrizione per l'allarme. Il nome deve contenere solo caratteri ASCII. Quindi scegli Successivo.
14. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Per creare un allarme su una query di Metrics Insights utilizzando AWS CLI

- Utilizza il comando `put-metric-alarm` e specifica una query di Approfondimenti sulle metriche nel parametro `metrics`. Ad esempio, il comando seguente imposta un allarme che passa allo stato ALARM se una delle istanze supera il 50% di utilizzo della CPU.

```
aws cloudwatch put-metric-alarm --alarm-name Metrics-Insights-alarm --
evaluation-periods 1 --comparison-operator GreaterThanThreshold --metrics
 '[{"Id":"m1","Expression":"SELECT MAX(CPUUtilization) FROM SCHEMA(\"AWS/EC2\",
 InstanceId)", "Period":60}]' --threshold 50
```

Casi di dati parziali

Se la query di Approfondimenti sulle metriche utilizzata per l'allarme corrisponde a più di 10.000 parametri, l'allarme viene valutato in base ai primi 10.000 parametri trovati dalla query. Ciò significa che l'allarme viene valutato sulla base di dati parziali.

Per sapere se un allarme di Approfondimenti sulle metriche sta valutando il suo stato di allarme sulla base di dati parziali, puoi utilizzare i seguenti metodi:

- Nella console, quando selezioni un allarme per visualizzare la pagina Details (Dettagli), viene mostrato il messaggio Evaluation warning: Not evaluating all data (Avviso di valutazione: impossibile valutare tutti i dati).
- Il valore `PARTIAL_DATA` nel `EvaluationState` campo viene visualizzato quando si utilizza il comando describe-alarms AWS CLI o l'API. DescribeAlarms

Gli allarmi pubblicano anche eventi su Amazon EventBridge quando passa allo stato parziale dei dati, quindi puoi creare una EventBridge regola per controllare questi eventi. In questi eventi, il campo `evaluationState` presenta il valore `PARTIAL_DATA`. Di seguito è riportato un esempio.

```
{
    "version": "0",
```

```
    "id": "12345678-3bf9-6a09-dc46-12345EXAMPLE",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-11-08T11:26:05Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:my-alarm-name"
    ],
    "detail": {
        "alarmName": "my-alarm-name",
        "state": {
            "value": "ALARM",
            "reason": "Threshold Crossed: 3 out of the last 3 datapoints [20000.0
 (08/11/22 11:25:00), 20000.0 (08/11/22 11:24:00), 20000.0 (08/11/22 11:23:00)] were
 greater than the threshold (0.0) (minimum 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2022-11-08T11:26:05.399+0000\",\"startDate\":\"2022-11-08T11:23:00.000+0000\",
\"period\":60,\"recentDatapoints\":[20000.0,20000.0,20000.0],\"threshold\":0.0,
\"evaluatedDatapoints\":[{\"timestamp\":\"2022-11-08T11:25:00.000+0000\",\"value
\":20000.0}]}",
            "timestamp": "2022-11-08T11:26:05.401+0000",
            "evaluationState": "PARTIAL_DATA"
        },
        "previousState": {
            "value": "INSUFFICIENT_DATA",
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2022-11-08T11:25:51.227+0000"
        },
        "configuration": {
            "metrics": [
                {
                    "id": "m2",
                    "expression": "SELECT SUM(PartialDataTestMetric) FROM
 partial_data_test",
                    "returnData": true,
                    "period": 60
                }
            ]
        }
    }
}
```

Se la query per l'allarme include un'istruzione GROUP BY che inizialmente restituisce più di 500 serie temporali, l'allarme viene valutato in base alle prime 500 serie temporali rilevate dalla query. Tuttavia, se utilizzi una clausola ORDER BY, tutte le serie temporali rilevate dalla query vengono ordinate e le 500 serie temporali con valori più alti o più bassi in base alla clausola ORDER BY vengono utilizzate per valutare l'allarme.

Utilizzo di query di Approfondimenti sulle metriche con formule di parametri

È possibile utilizzare una query Metrics Insights come input per una funzione matematica dei parametri. Per ulteriori informazioni sulla matematica dei parametri, consulta Utilizzare la matematica dei parametri.

Una query Metrics Insights che non include una clausola GROUP BY (RAGGRUPPA PER) restituisce una singola serie temporale. Pertanto, i risultati restituiti possono essere utilizzati con qualsiasi funzione matematica del parametro che richiede una singola serie temporale come input.

Una query Metrics Insights che include una clausola GROUP BY (RAGGRUPPA PER) restituisce più serie temporali. Pertanto, i risultati restituiti possono essere utilizzati con qualsiasi funzione matematica del parametro che prende come input una serie di serie temporali.

Ad esempio, la seguente query restituisce il numero totale di byte scaricati per ciascun bucket nella Regione, come matrice di serie temporali:

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
```

In un grafico nella console o in un'GetMetricDataoperazione, i risultati di questa query sono. q1 Questa query restituisce il risultato in byte, quindi se si vuole vedere il risultato come MB, è possibile usare la seguente funzione matematica:

```
q1/1024/1024
```

Utilizza il linguaggio naturale per generare e aggiornare le query di CloudWatch Metrics Insights

Questa funzionalità è disponibile in anteprima negli Stati Uniti orientali (Virginia settentrionale), negli Stati Uniti occidentali (Oregon) e nell'Asia Pacifico (Tokyo) CloudWatch ed è soggetta a modifiche.

CloudWatch supporta una funzionalità di interrogazione in linguaggio naturale per aiutarti a generare e aggiornare le query per CloudWatch Metrics Insights e Logs Insights. CloudWatch

Con questa funzionalità, puoi porre domande o descrivere i CloudWatch dati che stai cercando in un inglese semplice. La funzionalità del linguaggio naturale genera un'interrogazione in base a un prompt immesso e fornisce una line-by-line spiegazione del funzionamento della query. Puoi anche aggiornare la tua query per analizzare ulteriormente i dati.

A seconda del tuo ambiente, puoi inserire istruzioni come “Quale istanza di Amazon Elastic Compute Cloud ha la rete in uscita più alta?” e “Mostrami le 10 migliori tabelle Amazon DynamoDB per numero di letture consumate”.

Per generare una query di CloudWatch Metrics Insights con questa funzionalità, apri l'editor di query di CloudWatch Metrics Insights nella vista del generatore o dell'editor e scegli Genera query.

Important

Per utilizzare la funzionalità di interrogazione in linguaggio naturale, è necessario utilizzare la policy CloudWatchFullAccess,, CloudWatchReadOnlyAccess, CloudWatchFullAccessV2 o. AdministratorAccessReadOnlyAccess
È inoltre possibile includere l'operazione `cloudwatch:GenerateQuery` in una policy inline o gestita dal cliente nuova o esistente.

Query di esempio

Gli esempi in questa sezione descrivono come generare e aggiornare le query utilizzando la funzionalità di linguaggio naturale.

> **Note**
>
> Per ulteriori informazioni sull'editor di query e sulla sintassi di CloudWatch Metrics Insights, consulta Componenti e sintassi delle query di CloudWatch Metrics Insights.

Esempio: generazione di una query in linguaggio naturale

Per generare un'interrogazione utilizzando il linguaggio naturale, inserisci un prompt e scegli Genera nuova query. Questo esempio mostra una query che esegue una ricerca di base.

Prompt

Di seguito è riportato un esempio di prompt che esegue la funzionalità per cercare le prime 10 tabelle DynamoDB che consumano maggiormente la capacità di lettura.

```
Show top 10 DynamoDB Tables by consumed reads
```

Query

Di seguito è riportato un esempio di query generata dalla funzionalità di linguaggio naturale in base al prompt. Nota come il prompt appare in un commento prima della query. Dopo l'esecuzione della query, puoi leggere una spiegazione del funzionamento della query.

```
# Show top 10 DynamoDB Tables by consumed reads
SELECT SUM("ConsumedReadCapacityUnits")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query selects the sum of consumed read capacity units for each DynamoDB table,
 groups the results by table name, orders the results from highest to lowest read
 capacity consumption, and limits the results to the top 10 tables.
```

> **Note**
>
> Per disattivare la visualizzazione del prompt e la spiegazione del funzionamento della query, utilizza l'icona a forma di ingranaggio nell'editor.

Esempio: aggiornamento di una query in linguaggio naturale

È possibile aggiornare una query modificando il prompt iniziale e scegliendo Aggiorna query.

Richiesta aggiornata

L'esempio seguente mostra una versione aggiornata del prompt precedente. Invece di un prompt che cerca le prime 10 tabelle DynamoDB che consumano maggiormente capacità di lettura, questo prompt ora esegue la funzionalità per ordinare i risultati in base al numero di byte restituiti.

```
Sort by bytes returned instead
```

Query aggiornata

Di seguito è riportato un esempio della query aggiornata. Nota come il prompt aggiornato appare in un commento prima della query aggiornata. Dopo l'esecuzione della query, puoi leggere una spiegazione di come la query originale è stata aggiornata.

```
# Sort by bytes returned instead
SELECT SUM("ReturnedBytes")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query modifies the original query to select the sum of returned bytes instead
 of consumed read capacity units, and orders the results from highest to lowest sum of
 returned bytes, limiting the results to the top 10 tables.
```

Rifiuto esplicito all'utilizzo dei dati volto al miglioramento del servizio

I dati prompt in linguaggio naturale forniti per addestrare il modello di intelligenza artificiale e generare query pertinenti vengono utilizzati esclusivamente per fornire e gestire il servizio. Questi dati potrebbero essere utilizzati per migliorare la qualità di Metrics Insights. CloudWatch La tua fiducia, la tua privacy e la sicurezza dei tuoi contenuti sono le nostre maggiori priorità. Per ulteriori informazioni, consulta i Termini del servizio AWS e la Policy sull'IA responsabile di AWS.

Puoi scegliere di non utilizzare i tuoi contenuti per sviluppare o migliorare la qualità delle query in linguaggio naturale creando una policy di rifiuto dei servizi di intelligenza artificiale. Per disattivare la raccolta dei dati per tutte le funzionalità di CloudWatch intelligenza artificiale, inclusa la funzionalità di generazione di query, devi creare una politica di opt-out per. CloudWatch Per ulteriori informazioni, consulta le Policy di rifiuto dei servizi di IA nella Guida per l'utente di AWS Organizations .

Inferenza SQL

CloudWatch Metrics Insights utilizza diversi meccanismi per dedurre l'intenzione di una determinata query SQL.

Argomenti

Creazione di bucket di tempo

I punti dati di serie temporali risultanti da una query vengono raggruppati in periodi fissi in base al periodo richiesto. Per aggregare i valori in SQL standard, è necessario definire una clausola RAGGRUPPA PER esplicita per raccogliere insieme tutte le osservazioni di un determinato periodo. Poiché questo è il modo standard per interrogare i dati delle serie temporali, CloudWatch Metrics Insights deduce il time bucketing senza la necessità di esprimere una clausola GROUP BY esplicita.

Ad esempio, quando viene eseguita una query con un periodo di un minuto, tutte le osservazioni appartenenti a quel minuto fino al successivo (escluso) vengono accumulate fino all'ora di inizio del bucket temporale. Ciò rende le istruzioni SQL di Metrics Insights più concise e meno prolisse.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

La query precedente restituisce una singola serie temporale (coppie valore-timestamp), che rappresenta l'utilizzo medio della CPU di tutte le istanze Amazon EC2. Supponendo che il periodo richiesto sia di un minuto, ogni punto dati restituito rappresenta la media di tutte le osservazioni misurate entro un intervallo specifico di un minuto (orario di inizio incluso, tempo finale escluso). Il timestamp relativo al punto dati specifico è l'ora di inizio del bucket

Proiezione campi

Le query Metrics Insights restituiscono sempre la proiezione del timestamp. Non è necessario specificare una colonna timestamp nella clausola SELEZIONA per ottenere il timestamp di ciascun valore corrispondente del punto dati. Per informazioni dettagliate su come viene calcolato il timestamp, consulta Creazione di bucket di tempo.

Quando si utilizza RAGGRUPPA PER, ogni nome di gruppo viene dedotto e proiettato anche nel risultato, in modo da poter raggruppare le serie temporali restituite.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
```

La query precedente restituisce una serie temporale per ogni istanza Amazon EC2. Ogni serie temporale viene etichettata dopo il valore dell'ID di istanza.

aggregazione globale ORDINA PER

Quando si utilizza ORDINA PER,FUNZIONE () determina quale funzione di aggregazione ordinare (i valori del punto dati dei parametri sottoposti a query). L'operazione di aggregazione viene eseguita su tutti i punti dati corrispondenti di ogni singola serie temporale nella finestra temporale sottoposta a query.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX()
LIMIT 10
```

La query precedente restituisce l'utilizzo della CPU per ogni istanza Amazon EC2, limitando il set di risultati a 10 voci. I risultati vengono ordinati in base al valore massimo delle singole serie temporali all'interno della finestra temporale richiesta. La clausola ORDINA PER viene applicata prima di LIMITE, in modo che l'ordine sia calcolato rispetto a più di 10 serie temporali.

Query di esempio di Metrics Insights

Questa sezione contiene esempi di utili query di CloudWatch Metrics Insights che puoi copiare e utilizzare direttamente o copiare e modificare nell'editor di query. Alcuni di questi esempi sono già disponibili nella console ed è possibile accedervi scegliendo Aggiungi query nella visualizzazione Parametri.

Esempi di Application Load Balancer

Richieste totali su tutti i bilanciatori del carico

```
SELECT SUM(RequestCount)
```

```
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
```

I 10 migliori bilanciatori del carico più attivi

```
SELECT MAX(ActiveConnectionCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
GROUP BY LoadBalancer
ORDER BY SUM() DESC
LIMIT 10
```

AWS Esempi di utilizzo delle API

Le 20 migliori AWS API in base al numero di chiamate nel tuo account

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API'
GROUP BY Service, Resource
ORDER BY COUNT() DESC
LIMIT 20
```

CloudWatch API ordinate per chiamate

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API' AND Service = 'CloudWatch'
GROUP BY Resource
ORDER BY COUNT() DESC
```

Esempi di Dynamo DB

Le 10 tabelle migliori per letture consumate

```
SELECT SUM(ProvisionedWriteCapacityUnits)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Le 10 tabelle migliori per byte restituiti

```
SELECT SUM(ReturnedBytes)
```

```
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Le 10 tabelle migliori per errori dell'utente

```
SELECT SUM(UserErrors)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Esempi Amazon Elastic Block Store

I primi 10 volumi Amazon EBS per byte scritti

```
SELECT SUM(VolumeWriteBytes)
FROM SCHEMA("AWS/EBS", VolumeId)
GROUP BY VolumeId
ORDER BY SUM() DESC
LIMIT 10
```

Tempo medio di scrittura del volume Amazon EBS

```
SELECT AVG(VolumeTotalWriteTime)
FROM SCHEMA("AWS/EBS", VolumeId)
```

Esempi di Amazon EC2

Utilizzo della CPU delle istanze EC2 ordinate dal maggiore

```
SELECT AVG(CPUUtilization)
  FROM SCHEMA("AWS/EC2", InstanceId)
  GROUP BY InstanceId
  ORDER BY AVG() DESC
```

Utilizzo medio della CPU in tutto il parco istanze

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

Le 10 istanze migliori con il massimo utilizzo della CPU

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX() DESC
LIMIT 10
```

In questo caso, l' CloudWatch agente sta raccogliendo una **`CPUUtilization`** metrica per applicazione. Questa query filtra la media di questo parametro per un nome specifico dell'applicazione.

```
SELECT AVG(CPUUtilization)
FROM "AWS/CWAgent"
WHERE ApplicationName = 'eCommerce'
```

Esempi Amazon Elastic Container Service

Utilizzo medio della CPU su tutti i cluster ECS

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
```

I 10 cluster principali per utilizzo della memoria

```
SELECT AVG(MemoryUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC
LIMIT 10
```

I 10 servizi principali per utilizzo della CPU

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

I 10 servizi principali per esecuzioneo attività (Container Insights)

```
SELECT AVG(RunningTaskCount)
FROM SCHEMA("ECS/ContainerInsights", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

Esempi Amazon Elastic Kubernetes Service Container Insights

Utilizzo medio della CPU su tutti i cluster EKS

```
SELECT AVG(pod_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
```

I 10 cluster principali per utilizzo della CPU per nodo

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

I 10 cluster principali per utilizzo della memoria pod

```
SELECT AVG(pop_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

I 10 nodi principali per utilizzo della CPU

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, NodeName)
GROUP BY ClusterName, NodeName
ORDER BY AVG() DESC LIMIT 10
```

I 10 pod principali per utilizzo della memoria

```
SELECT AVG(pod_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, PodName)
GROUP BY ClusterName, PodName
```

```
ORDER BY AVG() DESC LIMIT 10
```

EventBridge esempi

Le 10 regole principali per invocazioni

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Le 10 regole principali per invocazioni non riuscite

```
SELECT SUM(FailedInvocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Le 10 regole principali in base alle regole corrispondenti

```
SELECT SUM(MatchedEvents)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Esempi di Kinesis

I 10 flussi principali per byte scritti

```
SELECT SUM("PutRecords.Bytes")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY SUM() DESC LIMIT 10
```

I 10 flussi principali per i primi elementi dello stream

```
SELECT MAX("GetRecords.IteratorAgeMilliseconds")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY MAX() DESC LIMIT 10
```

Esempi Lambda

Funzioni Lambda ordinate per numero di invocazioni

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
```

Le prime 10 funzioni Lambda per un runtime più lungo

```
SELECT AVG(Duration)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY MAX() DESC
LIMIT 10
```

Le prime 10 funzioni Lambda per numero di errori

```
SELECT SUM(Errors)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
LIMIT 10
```

CloudWatch Esempi di log

I primi 10 gruppi di log per eventi in arrivo

```
SELECT SUM(IncomingLogEvents)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

I primi 10 gruppi di log per byte scritti

```
SELECT SUM(IncomingBytes)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Esempi di Amazon RDS

Le 10 principali istanze Amazon RDS per massimo utilizzo della CPU

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/RDS", DBInstanceIdentifier)
GROUP BY DBInstanceIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

I 10 principali cluster Amazon RDS per scrittura

```
SELECT SUM(WriteIOPS)
FROM SCHEMA("AWS/RDS", DBClusterIdentifier)
GROUP BY DBClusterIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Esempi di codice di Amazon Simple Storage Service

Latenza media per bucket

```
SELECT AVG(TotalRequestLatency)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY AVG() DESC
```

I primi 10 bucket per byte scaricati

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY SUM() DESC
LIMIT 10
```

Esempi di Amazon Simple Notification Service

Messaggi totali pubblicati per argomenti SNS

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
```

I 10 argomenti principali per messaggi pubblicati

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

I 10 argomenti principali per errori di recapito dei messaggi

```
SELECT SUM(NumberOfNotificationsFailed)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Esempi di Amazon SQS

Le prime 10 code per numero di messaggi visibili

```
SELECT AVG(ApproximateNumberOfMessagesVisible)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Le 10 code più attive

```
SELECT SUM(NumberOfMessagesSent)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY SUM() DESC
LIMIT 10
```

Le prime 10 code per età del primo messaggio

```
SELECT AVG(ApproximateAgeOfOldestMessage)
FROM SCHEMA("AWS/SQS", QueueName)
```

```
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Limiti di Metrics Insights

CloudWatch Metrics Insights ha attualmente i seguenti limiti:

- Attualmente, è possibile effettuare una query solo per le ultime tre ore di dati.
- Una singola query può elaborare non più di 10.000 parametri. Ciò significa che se le clausole SELEZIONARE, DA, e DOVE corrispondono a più di 10.000 parametri, la query elabora solo i primi 10.000 di questi parametri trovati.
- Una singola query può restituire non più di 500 serie temporali. Ciò significa che se la query restituisce più di 500 parametri, non tutti i parametri verranno restituiti nei risultati della query. Se si utilizza una clausola ORDINA PER, tutti i parametri elaborati vengono ordinati e vengono restituiti i 500 che hanno i valori più alti o più bassi in base alla clausola ORDINA PER.

  Se non si include una clausola ORDINA PER, non è possibile controllare quali dei 500 parametri corrispondenti vengono restituiti.
- Puoi avere fino a 200 allarmi Metrics Insights per regione.
- Metrics Insights non supporta i dati ad alta risoluzione, ovvero dati di parametri riportati con una granularità inferiore a un minuto. Se richiedi dati ad alta risoluzione, la richiesta non ha esito negativo, ma l'output viene aggregato con granularità di un minuto.
- Ogni GetMetricDataoperazione può avere una sola query, ma puoi avere più widget in una dashboard, ognuno dei quali include una query.

Glossario di Metrics Insights

etichetta

In Metrics Insights, un'etichetta è una coppia chiave-valore utilizzata per definire una query per restituire un determinato set di dati o per definire i criteri in base ai quali i risultati delle query devono essere separati in serie temporali separate. Una chiave di etichetta è simile al nome di una colonna in SQL. Attualmente, le etichette devono avere dimensioni CloudWatch metriche.

osservazione

Un'osservazione è un valore registrato per un determinato parametro in un determinato momento.

Risoluzione dei problemi relativi Metrics Insights

I risultati includono «Altro», ma non ho questa come dimensione

Ciò significa che la query include una clausola RAGGRUPPA PERche specifica una chiave di etichetta che non viene utilizzata in alcuno dei parametri restituiti dalla query. In questo caso, viene restituito un gruppo nullo denominato `Other`. I parametri che non includono quella chiave etichetta sono probabilmente parametri aggregati che restituiscono valori aggregati su tutti i valori di quella chiave di etichetta.

Si prenda come esempio la seguente query:

```
SELECT AVG(Faults)
FROM MyCustomNamespace
GROUP BY Operation, ServiceName
```

Se alcuni dei parametri restituiti non includono `ServiceName` come dimensione, tali parametri vengono visualizzati come aventi `Other` come valore per `ServiceName`.

Per evitare di vedere «Altro» nei risultati, utilizzare SCHEMA nella clausola DA, come nell'esempio seguente:

```
SELECT AVG(Faults)
FROM SCHEMA(MyCustomNamespace, Operation)
GROUP BY Operation, ServiceName
```

Ciò limita i risultati restituiti solo ai parametri che hanno entrambe le dimensioni `Operation` e `ServiceName`.

Il timestamp più vecchio del mio grafico ha un valore del parametro inferiore rispetto agli altri

CloudWatch Metrics Insights attualmente supporta solo le ultime tre ore di dati. Quando si crea un grafico con un periodo superiore a un minuto, potrebbero esserci casi in cui il punto dati più vecchio differisce dal valore previsto. Questo perché le query di Metrics Insights restituiscono solo le tre ore di dati più recenti. In questo caso, il punto dati più vecchio della query restituisce solo le osservazioni misurate entro il limite delle ultime tre ore invece di restituire tutte le osservazioni entro il periodo di quel punto dati.

Usa metrics explorer per monitorare le risorse in base ai tag e alle proprietà

Metrics Explorer è uno strumento basato su tag che consente di filtrare, aggregare e visualizzare i parametri in base ai tag e alle proprietà delle risorse, per migliorare l'osservabilità dei servizi. Ciò offre un'esperienza di risoluzione dei problemi flessibile e dinamica, in modo da creare più grafici contemporaneamente e utilizzarli per creare pannelli di controllo dell'integrità delle applicazioni.

Le visualizzazioni di Metrics Explorer sono dinamiche, quindi se viene creata una risorsa corrispondente dopo aver creato un widget Metrics Explorer e averlo aggiunto a una CloudWatch dashboard, la nuova risorsa viene visualizzata automaticamente nel widget Explorer.

Ad esempio, se tutte le istanze di produzione EC2 hanno il tag**`production`**, puoi utilizzare Metrics Explorer per filtrare e aggregare i parametri di tutte queste istanze per comprenderne l'integrità e le prestazioni. Se in seguito viene creata una nuova istanza con un tag corrispondente, viene aggiunta automaticamente al widget di Metrics Explorer.

> **Note**
>
> Metrics explorer offre un'esperienza. point-in-time Le risorse che sono state terminate o che non esistono più con la proprietà o il tag specificato non saranno visualizzate nella visualizzazione. Tuttavia, puoi comunque trovare le metriche per queste risorse nelle viste delle CloudWatch metriche.

Con Metrics Explorer, puoi scegliere come aggregare i parametri dalle risorse che corrispondono ai criteri e se visualizzarli tutti in un singolo grafico o in grafici diversi all'interno di un widget Metrics Explorer.

Metrics Explorer include modelli che puoi utilizzare per visualizzare grafici di visualizzazione utili con un solo clic. Puoi anche estendere questi modelli per creare widget di Metrics Explorer completamente personalizzati.

Metrics explorer supporta le metriche emesse da AWS e le metriche EC2 pubblicate dall' CloudWatch agente, incluse le metriche di memoria, disco e CPU. Per utilizzare Metrics Explorer per visualizzare le metriche pubblicate dall' CloudWatch agente, potrebbe essere necessario aggiornare il file di configurazione dell'agente. CloudWatch Per ulteriori informazioni, consulta CloudWatch configurazione dell'agente per Metrics Explorer

Per creare una visualizzazione con Metrics Explorer e facoltativamente aggiungerla a un pannello di controllo, procedi come segue.

Per creare una visualizzazione con Metrics Explorer

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Explorer.
3. Esegui una di queste operazioni:
   - Per utilizzare un modello, selezionalo nella casella che mostra Empty Explorer (Explorer vuoto).

     A seconda del modello, Explorer potrebbe visualizzare immediatamente i grafici dei parametri. In caso contrario, scegli uno o più tag o proprietà nella finestra From (Da), quindi verranno visualizzati i dati. In caso contrario, utilizza le opzioni nella parte superiore della pagina per visualizzare un intervallo di tempo più lungo nei grafici.
   - Per creare una visualizzazione personalizzata, in Metrics (Parametri) sceglie un singolo parametri o tutti i parametri disponibili da un servizio.

     Dopo aver scelto un parametro, puoi ripetere facoltativamente questo passaggio per aggiungere ulteriori parametri.
4. Per ogni metrica selezionata, CloudWatch visualizza la statistica che verrà utilizzata immediatamente dopo il nome della metrica. Per modificare il parametro, scegli il nome della statistica, quindi scegli il parametro desiderato.
5. In From (Da), scegli un tag o una proprietà della risorsa per filtrare i risultati.

   Dopo aver eseguito questa operazione, facoltativamente puoi ripetere questo passaggio per scegliere più tag o proprietà delle risorse.

   Se scegli più valori della stessa proprietà, ad esempio due tipi di istanza EC2, verranno visualizzate tutte le risorse che corrispondono a una delle proprietà selezionate. Questa operazione viene trattata come operazione OR.

   Se si scelgono proprietà o tag diversi, ad esempio il tag **`Production`** e il tipo di istanza M5, vengono visualizzate solo le risorse che corrispondono a tutte queste selezioni. Questa operazione viene trattata come operazione AND.
6. (Facoltativo) In Aggregate by (Aggrega per), scegli una statistica da utilizzare per aggregare i parametri. Quindi, accanto a for (per), scegli come aggregare il parametro dall'elenco. Puoi

aggregare tutte le risorse attualmente visualizzate oppure aggregare in base a un singolo tag o proprietà della risorsa.

A seconda di come scegli di aggregare, il risultato potrebbe essere una singola serie temporale o più serie temporali.

7. In Split by (Dividi per), puoi scegliere di dividere un singolo grafico con più serie temporali in più grafici. La divisione può essere fatta in base a una varietà di criteri che puoi scegliere in Split by (Dividi per).
8. In Graph options (Opzioni del grafico), puoi perfezionare il grafico modificando il periodo, il tipo di grafico, il posizionamento della legenda e il layout.
9. Per aggiungere questa visualizzazione come widget a una CloudWatch dashboard, scegli Aggiungi alla dashboard.

CloudWatch configurazione dell'agente per Metrics Explorer

Per consentire a Metrics Explorer di scoprire le metriche EC2 pubblicate dall' CloudWatch agente, assicurati che il file di configurazione dell' CloudWatch agente contenga i seguenti valori:

- Nella sezione `metrics`, accertati che il parametro `aggregation_dimensions` includa `["InstanceId"]`. Può anche contenere altre dimensioni.
- Nella sezione `metrics`, accertati che il parametro `append_dimensions` includa una riga `{"InstanceId":"${aws:InstanceId}"}`. Può anche includere altre righe.
- Nella sezione `metrics`, all'interno della sezione `metrics_collected`, controlla le sezioni relative a ciascun tipo di risorsa che Metrics Explorer deve individuare, ad esempio le sezioni `cpu`, `disk` e `memory`. Assicurati che ognuna di queste sezioni abbia una `"resources": [ "*"]` `line.`.
- Nella sezione `cpu` della sezione `metrics_collected`, assicurati che ci sia una riga `"totalcpu": true`.
- È necessario utilizzare lo spazio dei `CWAgent` nomi predefinito per le metriche raccolte dall' CloudWatch agente, anziché uno spazio dei nomi personalizzato.

Le impostazioni nell'elenco precedente fanno sì che l' CloudWatch agente pubblichi metriche aggregate per dischi, CPU e altre risorse che possono essere tracciate in Metrics Explorer per tutte le istanze che le utilizzano.

Queste impostazioni ripubblicheranno i parametri precedentemente impostati per la pubblicazione con più dimensioni, aggiungendo i costi dei parametri.

Per ulteriori informazioni sulla modifica del file di configurazione dell'agente, vedere. CloudWatch Crea o modifica manualmente il file di configurazione dell' CloudWatch agente

Utilizzo dei flussi di parametri

Puoi utilizzare i flussi metrici per trasmettere continuamente le CloudWatch metriche verso una destinazione di tua scelta, con consegna e bassa latenza. near-real-time Le destinazioni supportate includono AWS destinazioni come Amazon Simple Storage Service e diverse destinazioni di fornitori di servizi di terze parti.

Esistono tre scenari di utilizzo principali per i flussi CloudWatch metrici:

- Configurazione personalizzata con Firehose: crea un flusso di metriche e indirizzalo a un flusso di distribuzione di Amazon Data Firehose che distribuisca i tuoi CloudWatch parametri dove desideri che vadano. Puoi trasmetterli in streaming su un data lake come Amazon S3 o su qualsiasi destinazione o endpoint supportato da Firehose, inclusi provider di terze parti. I formati JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0 sono supportati in modo nativo oppure è possibile configurare le trasformazioni nel flusso di distribuzione di Firehose per convertire i dati in un formato diverso, ad esempio Parquet. Con un flusso di metriche, puoi aggiornare continuamente i dati di monitoraggio o combinare questi dati CloudWatch metrici con dati di fatturazione e prestazioni per creare set di dati completi. Puoi quindi utilizzare strumenti come Amazon Athena per ottenere informazioni dettagliate sull'ottimizzazione dei costi, le prestazioni delle risorse e l'utilizzo delle risorse.
- Configurazione rapida di S3: trasmetti ad Amazon Simple Storage Service con un processo di configurazione rapida. Per impostazione predefinita, CloudWatch crea le risorse necessarie per lo stream. Sono supportati i formati JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0.
- Configurazione rapida per i AWS partner: CloudWatch offre un'esperienza di configurazione rapida per alcuni partner di terze parti. È possibile utilizzare fornitori di servizi di terze parti per monitorare, risolvere i problemi e analizzare le applicazioni utilizzando i dati in streaming CloudWatch . Quando utilizzi il flusso di lavoro di configurazione rapida dei partner, devi fornire solo un URL di destinazione e una chiave API per la destinazione e CloudWatch gestire il resto della configurazione. La configurazione rapida per i partner è disponibile per i seguenti provider di terze parti:

  - Datadog

- Dynatrace
- New Relic
- Splunk Observability Cloud
- SumoLogic

Puoi trasmettere in streaming tutte le tue CloudWatch metriche o utilizzare i filtri per trasmettere solo le metriche specificate. Ogni flusso di parametri può includere fino a 1.000 filtri che includono o escludono gli spazi dei nomi dei parametri o parametri specifici. Un singolo flusso di parametri può includere o escludere solo i filtri, ma non entrambe le operazioni.

Dopo aver creato un flusso di parametri, se vengono creati nuovi parametri che corrispondono ai filtri in posizione, i nuovi parametri vengono automaticamente inclusi nel flusso.

Non esiste alcun limite al numero di flussi di parametri per account o per regione e nessun limite al numero di aggiornamenti dei parametri in streaming.

Ogni stream può utilizzare il formato JSON, il formato OpenTelemetry 1.0.0 o il formato 0.7.0. OpenTelemetry È possibile modificare il formato di output di un flusso metrico in qualsiasi momento, ad esempio per l'aggiornamento da 0.7.0 a 1.0.0. OpenTelemetry OpenTelemetry Per ulteriori informazioni sui formati di output, consulta Formati di output dei flussi di parametri.

Per i flussi di parametri negli account di monitoraggio, puoi decidere se includere i parametri dagli account di origine collegati a quell'account di monitoraggio. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

I flussi di parametri includono sempre le statistiche `Minimum`, `Maximum`, `SampleCount` e`Sum`. È possibile anche scegliere di includere statistiche aggiuntive a un costo aggiuntivo. Per ulteriori informazioni, consulta Statistiche che possono essere trasmesse.

Il prezzo dei flussi di parametri si basa sul numero di aggiornamenti dei parametri. Inoltre, Firehose addebita addebiti per il flusso di consegna utilizzato per il flusso metrico. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

Argomenti

- Impostazione di un flusso di parametri
- Statistiche che possono essere trasmesse
- Funzionamento e manutenzione del flusso di parametri
- Monitora i tuoi flussi metrici con le metriche CloudWatch

- Trust between CloudWatch e Firehose
- Formati di output dei flussi di parametri
- Risoluzione dei problemi

Impostazione di un flusso di parametri

Utilizza i passaggi descritti nelle sezioni seguenti per configurare un CloudWatch flusso di metriche.

Dopo la creazione di un flusso metrico, il tempo necessario alla visualizzazione dei dati metrici nella destinazione dipende dalle impostazioni di buffering configurate nel flusso di distribuzione di Firehose. Il buffer è espresso nella dimensione massima del payload o nel tempo di attesa massimo, a seconda di quale sia stato raggiunto per primo. Se queste sono impostate sui valori minimi (60 secondi, 1 MB), la latenza prevista è compresa tra 3 minuti se i namespace selezionati CloudWatch hanno aggiornamenti metrici attivi.

In un flusso CloudWatch metrico, i dati vengono inviati ogni minuto. I dati potrebbero arrivare alla destinazione finale fuori uso. Tutte le metriche specificate nei namespace specificati vengono inviate nel flusso di metriche, ad eccezione delle metriche con un timestamp che risale a più di due giorni.

Per ogni combinazione di nome del parametro e spazio dei nomi in streaming, vengono trasmesse tutte le combinazioni di dimensioni del nome parametro e dello spazio dei nomi.

Per i flussi di parametri negli account di monitoraggio, puoi decidere se includere i parametri dagli account di origine collegati a quell'account di monitoraggio. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Per creare e gestire i flussi di metriche, devi accedere a un account che dispone della CloudWatchFullAccesspolitica e dell'autorizzazione o a un account con il seguente elenco di `iam:PassRole` autorizzazioni:

- `iam:PassRole`
- `cloudwatch:PutMetricStream`
- `cloudwatch:DeleteMetricStream`
- `cloudwatch:GetMetricStream`
- `cloudwatch:ListMetricStreams`
- `cloudwatch:StartMetricStreams`
- `cloudwatch:StopMetricStreams`

Se hai intenzione di CloudWatch configurare il ruolo IAM necessario per i flussi di metriche, devi disporre anche delle autorizzazioni and. `iam:CreateRole` `iam:PutRolePolicy`

> **Important**
>
> Un utente con `cloudwatch:PutMetricStream` ha accesso ai dati CloudWatch metrici che vengono trasmessi in streaming, anche se non dispone dell'autorizzazione. `cloudwatch:GetMetricData`

Argomenti

Configurazione personalizzata con Firehose

Usa questo metodo per creare un flusso di metriche e indirizzarlo a un flusso di distribuzione di Amazon Data Firehose che distribuisca CloudWatch i tuoi parametri dove desideri che vadano. Puoi trasmetterli in streaming su un data lake come Amazon S3 o su qualsiasi destinazione o endpoint supportato da Firehose, inclusi provider di terze parti.

I formati JSON, OpenTelemetry 1.0.0 e OpenTelemetry 0.7.0 sono supportati in modo nativo oppure è possibile configurare le trasformazioni nel flusso di distribuzione di Firehose per convertire i dati in un formato diverso, ad esempio Parquet. Con un flusso di metriche, puoi aggiornare continuamente i dati di monitoraggio o combinare questi dati CloudWatch metrici con dati di fatturazione e prestazioni per creare set di dati completi. Puoi quindi utilizzare strumenti come Amazon Athena per ottenere informazioni dettagliate sull'ottimizzazione dei costi, le prestazioni delle risorse e l'utilizzo delle risorse.

Puoi utilizzare la CloudWatch console, il AWS CLI AWS CloudFormation, o il AWS Cloud Development Kit (AWS CDK) per configurare un flusso di metriche.

Il flusso di distribuzione Firehose utilizzato per il flusso metrico deve trovarsi nello stesso account e nella stessa regione in cui è stato impostato il flusso metrico. Per ottenere la funzionalità interregionale, è possibile configurare il flusso di distribuzione di Firehose per lo streaming verso una destinazione finale che si trova in un account diverso o in una regione diversa.

CloudWatch console

Questa sezione descrive come utilizzare la CloudWatch console per configurare un flusso metrico utilizzando Firehose.

Per configurare un flusso metrico personalizzato utilizzando Firehose

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), Streams (Flussi). Quindi scegli Create metric stream (Crea flusso parametri).
3. (Facoltativo) Se hai effettuato l'accesso a un account configurato come account di monitoraggio nell'osservabilità tra CloudWatch account, puoi scegliere se includere le metriche degli account di origine collegati in questo flusso di metriche. Per includere i parametri dagli account di origine, seleziona Include source account metrics (Includi parametri degli account di origine).
4. Scegliete Configurazione personalizzata con Firehose.
5. Per Seleziona lo stream di Kinesis Data Firehose, seleziona lo stream di distribuzione Firehose da utilizzare. Deve essere nello stesso account. Il formato predefinito per questa opzione è OpenTelemetry 0.7.0, ma è possibile modificare il formato più avanti in questa procedura.

   Quindi seleziona il flusso di distribuzione Firehose da utilizzare in Seleziona il flusso di distribuzione Firehose.
6. (Facoltativo) Puoi scegliere Seleziona il ruolo di servizio esistente per utilizzare un ruolo IAM esistente invece di doverne CloudWatch creare uno nuovo per te.
7. (Facoltativo) Per modificare il formato di output dal formato predefinito per lo scenario, scegli Change output format (Cambia formato di output). I formati supportati sono JSON, OpenTelemetry 1.0.0 e 0.7.0. OpenTelemetry
8. Per lo streaming delle metriche, scegli Tutte le metriche o Seleziona metriche.

   Se scegli Tutte le metriche, tutte le metriche di questo account verranno incluse nello stream.

   Considera attentamente se eseguire lo streaming di tutti i parametri, poiché più parametri vengono trasmessi in streaming più alti saranno gli addebiti del flusso di parametri.

   Se scegli Seleziona metriche, esegui una delle seguenti operazioni:

   - Per eseguire lo streaming della maggior parte dei namespace delle metriche, scegli Escludi e seleziona i namespace o le metriche da escludere. Quando specifichi uno spazio dei nomi in Exclude, puoi facoltativamente selezionare alcune metriche specifiche da quel

namespace da escludere. Se scegli di escludere uno spazio dei nomi ma non selezioni quindi le metriche in quel namespace, tutte le metriche di quel namespace vengono escluse.

- Per includere solo alcuni namespace o metriche nel flusso di metriche, scegli Includi e quindi seleziona i namespace o le metriche da includere. Se scegli di includere uno spazio dei nomi ma non selezioni le metriche in quel namespace, vengono incluse tutte le metriche di quel namespace.

9. (Facoltativo) Per trasmettere statistiche aggiuntive per alcune di queste metriche oltre a Minimo, Massimo e Somma, scegli Aggiungi statistiche aggiuntive. SampleCount Scegli Add recommended metrics (Aggiungi parametri consigliati) per aggiungere alcune statistiche di uso comune o selezionare manualmente lo spazio dei nomi e il nome parametro per trasmettere statistiche aggiuntive. Quindi, seleziona le statistiche aggiuntive da trasmettere.

   Per scegliere un altro gruppo di parametri per trasmettere un diverso set di statistiche aggiuntive, scegli Aggiungi altre statistiche. Ogni parametro può includere fino a 20 statistiche aggiuntive e fino a 100 parametri all'interno di un flusso di parametri possono includere statistiche aggiuntive.

   Lo streaming di statistiche aggiuntive comporta ulteriori costi. Per ulteriori informazioni, consulta Statistiche che possono essere trasmesse.

   Per le definizioni delle statistiche aggiuntive, vedereCloudWatch definizioni statistiche.
10. (Facoltativo) Personalizza il nome del nuovo flusso di parametri in Metric stream name (Nome del flusso di parametri).
11. Scegli Create metric stream (Crea filtro parametri).

AWS CLI o API AWS

Utilizza i seguenti passaggi per creare un flusso di CloudWatch metriche.

Per utilizzare l' AWS API AWS CLI o per creare un flusso di metriche

1. Se stai eseguendo lo streaming su Amazon S3, crea prima il bucket. Per ulteriori informazioni, consulta Creazione di un bucket.
2. Crea il flusso di distribuzione di Firehose. Per ulteriori informazioni, vedere Creazione di uno stream Firehose.
3. Crea un ruolo IAM che CloudWatch consenta di scrivere nel flusso di distribuzione Firehose. Per ulteriori informazioni sui contenuti di questo ruolo, consulta Trust between CloudWatch e Firehose.

4. Utilizza il comando `aws cloudwatch put-metric-stream` CLI o l'`PutMetricStream`API per creare il flusso di CloudWatch metriche.

AWS CloudFormation

Puoi usarlo AWS CloudFormation per configurare un flusso di metriche. Per ulteriori informazioni, consulta AWS::CloudWatch::MetricStream.

Da utilizzare AWS CloudFormation per creare un flusso metrico

1. Se stai eseguendo lo streaming su Amazon S3, crea prima il bucket. Per ulteriori informazioni, consulta Creazione di un bucket.
2. Crea il flusso di distribuzione di Firehose. Per ulteriori informazioni, vedere Creazione di uno stream Firehose.
3. Crea un ruolo IAM che CloudWatch consenta di scrivere nel flusso di distribuzione Firehose. Per ulteriori informazioni sui contenuti di questo ruolo, consulta Trust between CloudWatch e Firehose.
4. Crea lo stream in AWS CloudFormation. Per ulteriori informazioni, consulta AWS::CloudWatch::MetricStream.

AWS Cloud Development Kit (AWS CDK)

Puoi usarlo AWS Cloud Development Kit (AWS CDK) per configurare un flusso metrico.

Da utilizzare AWS CDK per creare un flusso metrico

1. Se stai eseguendo lo streaming su Amazon S3, crea prima il bucket. Per ulteriori informazioni, consulta Creazione di un bucket.
2. Crea il flusso di distribuzione di Firehose. Per ulteriori informazioni, consulta Creating an Amazon Data Firehose Delivery Stream.
3. Crea un ruolo IAM che CloudWatch consenta di scrivere nel flusso di distribuzione Firehose. Per ulteriori informazioni sui contenuti di questo ruolo, consulta Trust between CloudWatch e Firehose.
4. Crea il flusso di parametri. La risorsa metric stream è disponibile AWS CDK come costrutto di livello 1 (L1) denominato. `CfnMetricStream` Per ulteriori informazioni, consulta Utilizzo dei costrutti L1.

Utilizzo della configurazione rapida di Amazon S3

Il metodo Quick S3 Setup funziona bene se desideri configurare rapidamente uno stream su Amazon S3 e non hai bisogno di alcuna trasformazione di formattazione oltre ai formati JSON OpenTelemetry , 1.0.0 e 0.7.0 supportati. OpenTelemetry CloudWatch creerà tutte le risorse necessarie, incluso il flusso di distribuzione di Firehose e i ruoli IAM necessari. Il formato predefinito per questa opzione è JSON, ma puoi modificarlo durante la configurazione del flusso.

In alternativa, se desideri che il formato finale sia Parquet oppure Optimized Row Columnar (ORC), ti basta seguire la procedura descritta nella sezione Configurazione personalizzata con Firehose.

CloudWatch console

Questa sezione descrive come utilizzare la CloudWatch console per configurare un flusso di parametri Amazon S3 utilizzando la configurazione Quick S3.

Configurazione di un flusso di parametri utilizzando Configurazione rapida S3

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), Streams (Flussi). Quindi scegli Create metric stream (Crea flusso parametri).
3. (Facoltativo) Se hai effettuato l'accesso a un account configurato come account di monitoraggio nell'osservabilità tra CloudWatch account, puoi scegliere se includere le metriche degli account di origine collegati in questo flusso di metriche. Per includere i parametri dagli account di origine, seleziona Include source account metrics (Includi parametri degli account di origine).
4. Scegli Configurazione rapida S3. CloudWatch creerà tutte le risorse necessarie, incluso il flusso di distribuzione di Firehose e i ruoli IAM necessari. Il formato predefinito per questa opzione è JSON, ma è possibile modificare il formato più avanti in questa procedura.
5. (Facoltativo) Scegli Seleziona risorse esistenti per utilizzare un bucket S3 esistente o ruoli IAM esistenti invece di doverne CloudWatch creare di nuovi per te.
6. (Facoltativo) Per modificare il formato di output dal formato predefinito per lo scenario, scegli Change output format (Cambia formato di output). I formati supportati sono JSON, OpenTelemetry 1.0.0 e 0.7.0. OpenTelemetry
7. Per lo streaming delle metriche, scegli Tutte le metriche o Seleziona metriche.

   Se scegli Tutte le metriche, tutte le metriche di questo account verranno incluse nello stream.

Considera attentamente se eseguire lo streaming di tutti i parametri, poiché più parametri vengono trasmessi in streaming più alti saranno gli addebiti del flusso di parametri.

Se scegli Seleziona metriche, esegui una delle seguenti operazioni:

- Per eseguire lo streaming della maggior parte dei namespace delle metriche, scegli Escludi e seleziona i namespace o le metriche da escludere. Quando specifichi uno spazio dei nomi in Exclude, puoi facoltativamente selezionare alcune metriche specifiche da quel namespace da escludere. Se scegli di escludere uno spazio dei nomi ma non selezioni quindi le metriche in quel namespace, tutte le metriche di quel namespace vengono escluse.
- Per includere solo alcuni namespace o metriche nel flusso di metriche, scegli Includi e quindi seleziona i namespace o le metriche da includere. Se scegli di includere uno spazio dei nomi ma non selezioni le metriche in quel namespace, vengono incluse tutte le metriche di quel namespace.

8. (Facoltativo) Per trasmettere statistiche aggiuntive per alcune di queste metriche oltre a Minimo, Massimo e Somma, scegli Aggiungi statistiche aggiuntive. SampleCount Scegli Add recommended metrics (Aggiungi parametri consigliati) per aggiungere alcune statistiche di uso comune o selezionare manualmente lo spazio dei nomi e il nome parametro per trasmettere statistiche aggiuntive. Quindi, seleziona le statistiche aggiuntive da trasmettere.

   Per scegliere un altro gruppo di parametri per trasmettere un diverso set di statistiche aggiuntive, scegli Aggiungi altre statistiche. Ogni parametro può includere fino a 20 statistiche aggiuntive e fino a 100 parametri all'interno di un flusso di parametri possono includere statistiche aggiuntive.

   Lo streaming di statistiche aggiuntive comporta ulteriori costi. Per ulteriori informazioni, consulta Statistiche che possono essere trasmesse.

   Per le definizioni delle statistiche aggiuntive, vedereCloudWatch definizioni statistiche.
9. (Facoltativo) Personalizza il nome del nuovo flusso di parametri in Metric stream name (Nome del flusso di parametri).
10. Scegli Create metric stream (Crea filtro parametri).

Configurazione rapida per i partner

CloudWatch offre un'esperienza di configurazione rapida per i seguenti partner di terze parti. Per utilizzare questo flusso di lavoro, è necessario fornire solo un URL di destinazione e una chiave API

per la destinazione. CloudWatch gestisce il resto della configurazione, inclusa la creazione del flusso di distribuzione Firehose e i ruoli IAM necessari.

> **Important**
>
> Prima di utilizzare la configurazione rapida per i partner per creare un flusso di parametri, ti consigliamo vivamente di leggere la documentazione del partner, un collegamento alla quale è fornito nel seguente elenco.

- Datadog
- Dynatrace
- New Relic
- Splunk Observability Cloud
- SumoLogic

Quando configuri un flusso di parametri per uno di questi partner, il flusso viene creato con alcune impostazioni predefinite, come elencato nelle sezioni seguenti.

Argomenti

Configurazione di un flusso di parametri utilizzando la configurazione rapida per i partner

CloudWatch fornisce un'opzione di configurazione rapida per alcuni partner di terze parti. Prima di iniziare la procedura descritta in questa sezione, è necessario disporre di determinate informazioni sul partner. Queste informazioni potrebbero includere un URL di destinazione e/o una chiave API per la destinazione del partner. Inoltre, è necessario leggere la documentazione disponibile sul sito web del partner, il cui collegamento è fornito nella sezione precedente, e le impostazioni predefinite per tale partner, elencate nelle sezioni seguenti.

Per eseguire lo streaming verso una destinazione di terze parti non supportata dalla configurazione rapida, puoi seguire le istruzioni in Segui le istruzioni in Configurazione personalizzata con Firehose Per configurare uno streaming utilizzando Firehose, quindi inviare tali metriche da Firehose alla destinazione finale.

Utilizzo della configurazione rapida per i partner per creare un flusso di parametri verso un provider di terze parti

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), Streams (Flussi). Quindi scegli Create metric stream (Crea flusso parametri).
3. (Facoltativo) Se hai effettuato l'accesso a un account configurato come account di monitoraggio nell'osservabilità tra CloudWatch account, puoi scegliere se includere le metriche degli account di origine collegati in questo flusso di metriche. Per includere i parametri dagli account di origine, seleziona Include source account metrics (Includi parametri degli account di origine).
4. Scegli la configurazione rapida per i partner di Amazon Web Services
5. Seleziona il nome del partner al quale desideri trasmettere i parametri.
6. Per URL dell'endpoint, inserisci l'URL di destinazione.
7. Per Chiave di accesso o Chiave API, inserisci la chiave di accesso per il partner. Non tutti i partner richiedono una chiave di accesso.
8. Per lo streaming delle metriche, scegli Tutte le metriche o Seleziona metriche.

   Se scegli Tutte le metriche, tutte le metriche di questo account verranno incluse nello stream.

   Considera attentamente se eseguire lo streaming di tutti i parametri, poiché più parametri vengono trasmessi in streaming più alti saranno gli addebiti del flusso di parametri.

   Se scegli Seleziona metriche, esegui una delle seguenti operazioni:

   - Per eseguire lo streaming della maggior parte dei namespace delle metriche, scegli Escludi e seleziona i namespace o le metriche da escludere. Quando specifichi uno spazio dei nomi in Exclude, puoi facoltativamente selezionare alcune metriche specifiche da quel namespace da escludere. Se scegli di escludere uno spazio dei nomi ma non selezioni quindi le metriche in quel namespace, tutte le metriche di quel namespace vengono escluse.
   - Per includere solo alcuni namespace o metriche nel flusso di metriche, scegli Includi e quindi seleziona i namespace o le metriche da includere. Se scegli di includere uno spazio dei

nomi ma non selezioni le metriche in quel namespace, vengono incluse tutte le metriche di quel namespace.

9. (Facoltativo) Per trasmettere statistiche aggiuntive per alcune di queste metriche oltre a Minimo, Massimo e Somma, scegli Aggiungi statistiche aggiuntive. SampleCount Scegli Add recommended metrics (Aggiungi parametri consigliati) per aggiungere alcune statistiche di uso comune o selezionare manualmente lo spazio dei nomi e il nome parametro per trasmettere statistiche aggiuntive. Quindi, seleziona le statistiche aggiuntive da trasmettere.

   Per scegliere un altro gruppo di parametri per trasmettere un diverso set di statistiche aggiuntive, scegli Aggiungi altre statistiche. Ogni parametro può includere fino a 20 statistiche aggiuntive e fino a 100 parametri all'interno di un flusso di parametri possono includere statistiche aggiuntive.

   Lo streaming di statistiche aggiuntive comporta ulteriori costi. Per ulteriori informazioni, consulta Statistiche che possono essere trasmesse.

   Per le definizioni delle statistiche aggiuntive, vedereCloudWatch definizioni statistiche.
10. (Facoltativo) Personalizza il nome del nuovo flusso di parametri in Metric stream name (Nome del flusso di parametri).
11. Scegli Create metric stream (Crea filtro parametri).

Impostazioni predefinite dei flussi Datadog

La configurazione rapida per i partner dei flussi verso Datadog utilizza le seguenti impostazioni predefinite:

- Formato di output: 0.7.0 OpenTelemetry
- Codifica dei contenuti in streaming Firehose in GZIP
- Opzioni di buffering dello stream Firehose Intervallo di 60 secondi, dimensione di 4 MB/s
- Opzione Firehose Stream Retry Durata di 60 secondi

Quando utilizzi la configurazione rapida per i partner per creare un flusso di parametri verso Datadog e trasmetti determinati parametri, per impostazione predefinita essi includono alcune statistiche aggiuntive. La trasmissione di statistiche aggiuntive può comportare costi aggiuntivi. Per ulteriori informazioni sulle statistiche e i rispettivi costi, consulta la pagina Statistiche che possono essere trasmesse.

L'elenco seguente mostra i parametri per i quali vengono trasmesse statistiche aggiuntive per impostazione predefinita, se scegli di trasmettere tali parametri. Puoi scegliere di deselezionare queste statistiche aggiuntive prima di avviare il flusso.

- **`Duration`** in **`AWS/Lambda`**: p50, p80, p95, p99, p99.9
- **`PostRuntimeExtensionDuration`** in **`AWS/Lambda`**: p50, p99
- **`FirstByteLatency`** e **`TotalRequestLatency`**in **`AWS/S3`**: p50, p90, p95, p99, p99.9
- **`ResponseLatency`** in **`AWS/Polly`** e **`TargetResponseTime`** in AWS/ApplicationELB: p50, p90, p95, p99
- **`Latency`** e **`IntegrationLatency`** in **`AWS/ApiGateway`**: p90, p95, p99
- **`Latency`** e **`TargetResponseTime`** in **`AWS/ELB`**: p95, p99
- **`RequestLatency`** in **`AWS/AppRunner`**: p50, p95, p99
- **`ActivityTime`**, **`ExecutionTime`**, **`LambdaFunctionRunTime`**, **`LambdaFunctionScheduleTime`**, **`LambdaFunctionTime`**, **`ActivityRunTime`** e **`ActivityScheduleTime`** in **`AWS/States`**: p95, p99
- **`EncoderBitRate`**, **`ConfiguredBitRate`** e **`ConfiguredBitRateAvailable`** in **`AWS/MediaLive`**: p90
- **`Latency`** in **`AWS/AppSync`**: p90

Impostazioni predefinite dei flussi Dynatrace

La configurazione rapida per i partner dei flussi verso Dynatrace utilizza le seguenti impostazioni predefinite:

- Formato di output: 0.7.0 OpenTelemetry
- Codifica dei contenuti in streaming Firehose in GZIP
- Opzioni di buffering dello stream Firehose Intervallo di 60 secondi, dimensione di 5 MB/s
- Opzione Firehose Stream Retry Durata di 600 secondi

Impostazioni predefinite dei flussi New Relic

La configurazione rapida per i partner dei flussi verso New Relic utilizza le seguenti impostazioni predefinite:

- Formato di output: 0.7.0 OpenTelemetry

- Codifica dei contenuti in streaming Firehose in GZIP
- Opzioni di buffering dello stream Firehose Intervallo di 60 secondi, dimensione di 1 MB
- Opzione Firehose Stream Retry Durata di 60 secondi

Impostazioni predefinite dei flussi Splunk Observability Cloud

La configurazione rapida per i partner dei flussi verso Splunk Observability Cloud utilizza le seguenti impostazioni predefinite:

- Formato di output: 0.7.0 OpenTelemetry
- Codifica dei contenuti in streaming Firehose in GZIP
- Opzioni di buffering dello stream Firehose Intervallo di 60 secondi, dimensione di 1 MB
- Opzione Firehose Stream Retry Durata di 300 secondi

Impostazioni predefinite dei flussi Sumo Logic

La configurazione rapida per i partner dei flussi verso Sumo Logic utilizza le seguenti impostazioni predefinite:

- Formato di output: 0.7.0 OpenTelemetry
- Codifica dei contenuti in streaming Firehose in GZIP
- Opzioni di buffering dello stream Firehose Intervallo di 60 secondi, dimensione di 1 MB
- Opzione Firehose Stream Retry Durata di 60 secondi

Statistiche che possono essere trasmesse

I flussi di parametri includono le seguenti statistiche: `Minimum`, `Maximum`, `SampleCount` e `Sum`. È inoltre possibile scegliere di includere le seguenti statistiche aggiuntive in un flusso di parametri. Questa scelta è basata su base di parametri. Per ulteriori informazioni sulle statistiche, consulta CloudWatch definizioni statistiche.

- Valori percentili come p95 o p99 (per stream con JSON o formato) OpenTelemetry
- Media troncata (solo per flussi con formato JSON)
- Media winsorizzata (solo per flussi con formato JSON)
- Conteggio troncato (solo per flussi con formato JSON)

- Somma troncata (solo per flussi con formato JSON)
- Grado percentile (solo per flussi con formato JSON)
- Media Interquartile (solo per flussi con formato JSON)

Lo streaming di statistiche aggiuntive comporta costi aggiuntivi. Lo streaming tra uno e cinque di queste statistiche aggiuntive per un determinato parametri viene fatturato come aggiornamento aggiuntivo del parametro. Successivamente, ogni set aggiuntivo di un massimo di cinque di queste statistiche viene fatturato come un altro aggiornamento dei parametri.

Ad esempio, supponiamo che per un parametro vengano trasmesse le seguenti sei statistiche aggiuntive: p95, p99, p99.9, media troncata, media winsorizzata e Somma troncata. Ogni aggiornamento di questo parametro viene fatturato come tre aggiornamenti dei parametri: uno per l'aggiornamento del parametro che include le statistiche predefinite, uno per le prime cinque statistiche aggiuntive e uno per la sesta statistica aggiuntiva. L'aggiunta di altre quattro statistiche aggiuntive per un totale di dieci non aumenterebbe la fatturazione, ma un'undicesima statistica aggiuntiva lo farebbe.

Quando si specifica una combinazione di nome parametro e spazio dei nomi per eseguire lo streaming di statistiche aggiuntive, tutte le combinazioni di dimensioni del nome del parametro e dello spazio dei nomi vengono trasmesse in streaming con le statistiche aggiuntive.

CloudWatch metric streams pubblica una nuova metrica`TotalMetricUpdate`, che riflette il numero base di aggiornamenti delle metriche più gli aggiornamenti aggiuntivi delle metriche derivanti dallo streaming di statistiche aggiuntive. Per ulteriori informazioni, consulta Monitora i tuoi flussi metrici con le metriche CloudWatch .

Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

> **Note**
>
> Alcuni parametri non supportano i percentili. Le statistiche percentili per queste metriche sono escluse dal flusso e non comportano addebiti per il flusso di parametri. Un esempio di queste statistiche che non supportano i percentili sono alcuni parametri nello `AWS/ECS`spazio dei nomi.

Le statistiche aggiuntive configurate vengono trasmesse in streaming solo se corrispondono ai filtri per lo streaming. Ad esempio, se un flusso viene creato solo`EC2`e`RDS`nei filtri di

inclusione e quindi negli elenchi di configurazione delle statisticheEC2eLambda, quindi il flusso includeEC2parametri con statistiche aggiuntive,RDSparametri con solo le statistiche predefinite e non includonoLambdastatistiche a tutti.

Funzionamento e manutenzione del flusso di parametri

I flussi di parametri sono sempre in uno dei due stati, Running (In esecuzione) o Stopped (Arrestato).

- Running (In esecuzione): il flusso di parametri viene eseguito correttamente. Potrebbe non esserci alcun dato di parametri trasmesso alla destinazione a causa dei filtri nel flusso.
- Stopped (Arrestato): il flusso di parametri è stato interrotto esplicitamente da qualcuno e non a causa di un errore. Potrebbe essere utile interrompere il flusso per sospendere temporaneamente lo streaming dei dati senza eliminare il flusso.

Se interrompi e riavvii un flusso di metriche, i dati delle metriche pubblicati CloudWatch durante l'interruzione del flusso di metriche non vengono ripristinati nel flusso di metriche.

Se si modifica il formato di output di un flusso di parametri, in alcuni casi è possibile che venga visualizzata una piccola quantità di dati di parametri scritti nella destinazione sia nel vecchio formato che nel nuovo formato. Per evitare questa situazione, è possibile creare un nuovo flusso di distribuzione Firehose con la stessa configurazione di quello attuale, quindi passare al nuovo flusso di distribuzione Firehose e modificare contemporaneamente il formato di output. In questo modo, i record Kinesis con diversi formati di output vengono memorizzati su Amazon S3 in oggetti separati. Successivamente, puoi reindirizzare il traffico al flusso di distribuzione Firehose originale ed eliminare il secondo flusso di distribuzione.

Per visualizzare, modificare, arrestare e avviare i flussi di parametri

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), Streams (Flussi).

   Viene visualizzato l'elenco dei flussi e la colonna Status (Stato) visualizza se ogni flusso è in esecuzione o arrestato.
3. Per arrestare o avviare un flusso di parametri, seleziona il flusso e scegli Stop (Arresta) o Start (Avvia).
4. Per visualizzare i dettagli su un flusso di parametri, seleziona il flusso e scegli View details (Visualizza dettagli).

5. Per modificare il formato di output dello stream, i filtri, lo stream Firehose di destinazione o i ruoli, scegliete Modifica e apportate le modifiche desiderate.

   Se modifichi i filtri, potrebbero verificarsi alcune lacune nei dati dei parametri durante la transizione.

Monitora i tuoi flussi metrici con le metriche CloudWatch

I flussi metrici emettono CloudWatch metriche sul loro stato e sul loro funzionamento nel namespace. `AWS/CloudWatch/MetricStreams` Vengono emessi i seguenti parametri. Questi parametri vengono emessi con una dimensione `MetricStreamName` e senza dimensione. Puoi utilizzare i parametri senza dimensioni per visualizzare i parametri aggregati per tutti i flussi di parametri. Puoi utilizzare i parametri con la dimensione `MetricStreamName` per visualizzare i parametri relativi solo a quel flusso di parametri.

Per tutti questi parametri, i valori vengono emessi solo per i flussi di parametri che si trovano nello stato Running (In esecuzione).

Parametro	Descrizione
`MetricUpdate`	Gli aggiornamenti dei parametri numerici inviati al flusso di parametri. Se durante un periodo di tempo non vengono trasmessi aggiornamenti dei parametri, questo parametro non viene emesso. Se interrompi il flusso del parametro, questo parametro interrompe l'emissione fino a quando il flusso del parametro non viene riavviato. Statistiche valide: `Sum` Unità: nessuna
`TotalMetricUpdate`	Viene calcolato come MetricUpdate + un numero in base alle statistiche aggiuntive trasmesse in streaming. Per ogni combinazione univoca dello spazio dei nomi e del nome parametro, lo streaming di 1-5 statistiche aggiuntive aggiunge 1 al`TotalMetricUpdate` , lo streaming 6-10 statistiche aggiuntive aggiunge 2 a`TotalMetricUpdate` , e così via.

Parametro	Descrizione
	Statistiche valide: `Sum` Unità: nessuna
`PublishErrorRate`	Il numero di errori irreversibili che si verificano durante l'inserimento dei dati nel flusso di distribuzione di Firehose. Se durante un periodo di tempo non si verificano errori, questo parametro non viene emesso. Se interrompi il flusso del parametro, questo parametro interrompe l'emissione fino a quando il flusso del parametro non viene riavviato. Statistiche valide: `Average` per vedere la frequenza degli aggiornamenti dei parametri che non possono essere scritti. Questo valore sarà compreso tra 0,0 e 1,0. Unità: nessuna

Trust between CloudWatch e Firehose

Il flusso di distribuzione Firehose deve essere affidabile CloudWatch tramite un ruolo IAM con autorizzazioni di scrittura per Firehose. Queste autorizzazioni possono essere limitate al singolo flusso di distribuzione Firehose utilizzato CloudWatch dal flusso metrico. Il ruolo IAM deve considerare attendibile il principale di servizio `streams.metrics.cloudwatch.amazonaws.com`.

Se utilizzi la CloudWatch console per creare un flusso di metriche, puoi CloudWatch creare il ruolo con le autorizzazioni corrette. Se utilizzi un altro metodo per creare un flusso di parametri o desideri creare il ruolo IAM stesso, è necessario che contenga le policy di autorizzazione e le policy di attendibilità riportate di seguito.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Effect": "Allow",
```

```
            "Resource": "arn:aws:firehose:region:account-id:deliverystream/*"
        }
    ]
}
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "streams.metrics.cloudwatch.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

I dati metrici vengono trasmessi in streaming CloudWatch al flusso di distribuzione Firehose di destinazione per conto della fonte proprietaria della risorsa metric stream.

Formati di output dei flussi di parametri

I dati in un flusso CloudWatch metrico possono essere nel formato JSON o nel formato. OpenTelemetry Attualmente sono supportati entrambi i formati OpenTelemetry 1.0.0 e 0.7.0.

Indice

Formato JSON

In un flusso CloudWatch metrico che utilizza il formato JSON, ogni record Firehose contiene più oggetti JSON separati da un carattere di nuova riga (\n). Ogni oggetto include un singolo punto dati di un singolo parametro.

Il formato JSON utilizzato è completamente compatibile con AWS Glue e con Amazon Athena. Se disponi di un flusso di distribuzione Firehose e di una AWS Glue tabella formattata correttamente, il formato può essere trasformato automaticamente in formato Parquet o in formato Orc (Optimized Row Columnar) prima di essere archiviato in S3. Per ulteriori informazioni sulla trasformazione del formato, vedere Conversione del formato di registrazione di input in Firehose. Per ulteriori informazioni sul formato corretto per AWS Glue, vedere. Quale schema AWS Glue dovrei usare per il formato di output JSON?

Nel formato JSON, i valori validi per `unit` sono gli stessi del valore di `unit` nella struttura dell'API `MetricDatum`. Per ulteriori informazioni, vedere MetricDatum. Il valore per il campo `timestamp` è in millisecondi di epoca, come `1616004674229`.

Di seguito è riportato un esempio del formato. In questo esempio, JSON è formattato per una facile lettura, ma in pratica l'intero formato si trova su un'unica riga.

```
{
    "metric_stream_name": "MyMetricStream",
    "account_id": "1234567890",
    "region": "us-east-1",
    "namespace": "AWS/EC2",
    "metric_name": "DiskWriteOps",
    "dimensions": {
        "InstanceId": "i-123456789012"
    },
    "timestamp": 1611929698000,
    "value": {
        "count": 3.0,
        "sum": 20.0,
        "max": 18.0,
        "min": 0.0,
        "p99": 17.56,
        "p99.9": 17.8764,
        "TM(25%:75%)": 16.43
    },
    "unit": "Seconds"
```

```
}
```

Quale schema AWS Glue dovrei usare per il formato di output JSON?

Di seguito è riportato un esempio di rappresentazione JSON della AWS Glue tabella `StorageDescriptor` for an, che verrebbe quindi utilizzata da Firehose. Per ulteriori informazioni su`StorageDescriptor`, vedere. StorageDescriptor

```
{
  "Columns": [
    {
      "Name": "metric_stream_name",
      "Type": "string"
    },
    {
      "Name": "account_id",
      "Type": "string"
    },
    {
      "Name": "region",
      "Type": "string"
    },
    {
      "Name": "namespace",
      "Type": "string"
    },
    {
      "Name": "metric_name",
      "Type": "string"
    },
    {
      "Name": "timestamp",
      "Type": "timestamp"
    },
    {
      "Name": "dimensions",
      "Type": "map<string,string>"
    },
    {
      "Name": "value",
      "Type":
 "struct<min:double,max:double,count:double,sum:double,p99:double,p99.9:double>"
    },
```

```
    {
      "Name": "unit",
      "Type": "string"
    }
  ],
  "Location": "s3://my-s3-bucket/",
  "InputFormat": "org.apache.hadoop.mapred.TextInputFormat",
  "OutputFormat": "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
  "SerdeInfo": {
    "SerializationLibrary": "org.apache.hive.hcatalog.data.JsonSerDe"
  },
  "Parameters": {
    "classification": "json"
  }
}
```

L'esempio precedente riguarda i dati scritti su Amazon S3 in formato JSON. Sostituisci i valori nei campi seguenti con i valori indicati per memorizzare i dati in formato Parquet o in formato ORC (Optimized Row Columnar).

- Parquet:
  - Formato di input: org.apache.hadoop.hive.ql.io.parquet. MapredParquetInputFormat
  - Formato di output: org.apache.hadoop.hive.ql.io.parquet. MapredParquetOutputFormat
  - SerDeInfo.serializationLib: org.apache.hadoop.hive.ql.io.parquet.serde. ParquetHiveSerDe
  - parameters.classification: parquet
- ORC:
  - Formato di input: org.apache.hadoop.hive.ql.io.orc. OrcInputFormat
  - Formato di output: org.apache.hadoop.hive.ql.io.orc. OrcOutputFormat
  - SerDeInfo.serializationLib: org.apache.hadoop.hive.ql.io.orc. OrcSerde
  - parameters.classification: orc

OpenTelemetry Formato 1.0.0

> **Note**
>
> Con il formato OpenTelemetry 1.0.0, gli attributi delle metriche vengono codificati come un elenco di `KeyValue` oggetti anziché il `StringKeyValue` tipo utilizzato nel formato 0.7.0. Come consumatore, questa è l'unica modifica importante tra i formati 0.7.0 e 1.0.0. Un parser

generato dai file proto 0.7.0 non analizza gli attributi dei parametri codificati nel formato 1.0.0. Lo stesso vale al contrario: un parser generato dai file proto 1.0.0 non analizza gli attributi metrici codificati nel formato 0.7.0.

OpenTelemetry è una raccolta di strumenti, API e SDK. Puoi utilizzarlo per strumentare, generare, raccogliere ed esportare dati di telemetria (metriche, log e tracce) per l'analisi. OpenTelemetry fa parte della Cloud Native Computing Foundation. Per ulteriori informazioni, vedere OpenTelemetry.

Per informazioni sulla specifica OpenTelemetry 1.0.0 completa, vedere la versione di rilascio 1.0.0.

Un record Kinesis può contenere una o più strutture di `ExportMetricsServiceRequest` OpenTelemetry dati. Ogni struttura dati inizia con un'intestazione con `UnsignedVarInt32` che indica la lunghezza del record in byte. Ciascuna `ExportMetricsServiceRequest` può contenere dati provenienti da più parametri contemporaneamente.

Di seguito è riportata una rappresentazione in formato stringa del messaggio della struttura `ExportMetricsServiceRequest` OpenTelemetry dati. OpenTelemetry serializza il protocollo binario di Google Protocol Buffers e questo non è leggibile dall'uomo.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "123456789012"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
    }
    attributes {
      key: "aws.exporter.arn"
```

```
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  scope_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "NoneTranslated"
      summary {
        data_points {
          start_time_unix_nano: 60000000000
          time_unix_nano: 120000000000
          count: 1
          sum: 1.0
          quantile_values {
            value: 1.0
          }
          quantile_values {
            quantile: 0.95
            value: 1.0
          }
          quantile_values {
            quantile: 0.99
            value: 1.0
          }
          quantile_values {
            quantile: 1.0
            value: 1.0
          }
          attributes {
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
            key: "MetricName"
            value {
              string_value: "ConsumedReadCapacityUnits"
            }
          }
          attributes {
```

```
            key: "Dimensions"
            value {
              kvlist_value {
                values {
                  key: "TableName"
                  value {
                    string_value: "MyTable"
                  }
                }
              }
            }
          }
        }
        data_points {
          start_time_unix_nano: 70000000000
          time_unix_nano: 130000000000
          count: 2
          sum: 5.0
          quantile_values {
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
          attributes {
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
            key: "MetricName"
            value {
              string_value: "ConsumedReadCapacityUnits"
            }
          }
          attributes {
            key: "Dimensions"
            value {
              kvlist_value {
                values {
                  key: "TableName"
                  value {
```

```
                    string_value: "MyTable"
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Oggetto di primo livello per serializzare i dati metrici OpenTelemetry

`ExportMetricsServiceRequest`è il wrapper di primo livello per serializzare un payload di esportazione. OpenTelemetry Contiene uno o più `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics`è l'oggetto di primo livello per rappresentare gli oggetti. MetricData

```
// A collection of ScopeMetrics from a Resource.
message ResourceMetrics {
  reserved 1000;

  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated ScopeMetrics scope_metrics = 2;

  // This schema_url applies to the data in the "resource" field. It does not apply
  // to the data in the "scope_metrics" field which have their own schema_url field.
  string schema_url = 3;
```

```
}
```

Oggetto della risorsa

Un oggetto `Resource` è un oggetto coppia di valori che contiene alcune informazioni sulla risorsa che ha generato i parametri. Per i parametri creati da AWS, la struttura dati contiene l'ARN (Amazon Resource Name) della risorsa correlata al parametro, ad esempio un'istanza EC2 o un bucket S3.

L'oggetto `Resource` contiene un attributo denominato `attributes`, che memorizza un elenco di coppie chiave-valore.

- `cloud.account.id` contiene l'ID dell'account
- `cloud.region` contiene la regione
- `aws.exporter.arn` contiene l'ARN del flusso di parametri
- `cloud.provider` è sempre `aws`.

```
// Resource information.
message Resource {
  // Set of attributes that describe the resource.
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
 then
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

L'oggetto ScopeMetrics

Il campo `scope` non verrà compilato. Compiliamo solo il campo dei parametri che stiamo esportando.

```
// A collection of Metrics produced by an Scope.
message ScopeMetrics {
  // The instrumentation scope information for the metrics in this message.
  // Semantically when InstrumentationScope isn't set, it is equivalent with
  // an empty instrumentation scope name (unknown).
  opentelemetry.proto.common.v1.InstrumentationScope scope = 1;
```

```
  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;

  // This schema_url applies to all metrics in the "metrics" field.
  string schema_url = 3;
}
```

L'oggetto Metric

L'oggetto Metric contiene alcuni metadati e un campo dati `Summary` che contiene un elenco di `SummaryDataPoint`.

Per i flussi di parametri, i metadati sono i seguenti:

- `name` sarà `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`*
- `description` sarà vuoto
- `unit` verrà compilato mappando l'unità del dato del parametro alla variante con distinzione tra maiuscole e minuscole del codice unificato per le unità di misura. Per ulteriori informazioni, consulta Traduzioni con formato OpenTelemetry 1.0.0 and Il codice unificato per le unità di misura.
- `type` sarà `SUMMARY`

```
message Metric {
  reserved 4, 6, 8;

  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  // Data determines the aggregation type (if any) of the metric, what is the
  // reported value type for the data points, as well as the relatationship to
  // the time interval over which they are reported.
  oneof data {
    Gauge gauge = 5;
    Sum sum = 7;
    Histogram histogram = 9;
```

```
    ExponentialHistogram exponential_histogram = 10;
    Summary summary = 11;
  }
}

message Summary {
  repeated SummaryDataPoint data_points = 1;
}
```

L' SummaryDataPoint oggetto

L' SummaryDataPoint oggetto contiene il valore di un singolo punto dati in una serie temporale in una DoubleSummary metrica.

```
// SummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message SummaryDataPoint {
  reserved 1;

  // The set of key/value pairs that uniquely identify the timeseries from
  // where this point belongs. The list may be empty (may contain 0 elements).
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 7;

  // StartTimeUnixNano is optional but strongly encouraged, see the
  // the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 start_time_unix_nano = 2;

  // TimeUnixNano is required, see the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;

  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
```

```
  //
  // Note: Sum should only be filled out when measuring non-negative discrete
  // events, and is assumed to be monotonic over the values of these events.
  // Negative events *can* be recorded, but sum should not be filled out when
  // doing so.  This is specifically to enforce compatibility w/ OpenMetrics,
  // see: https://github.com/OpenObservability/OpenMetrics/blob/main/specification/
OpenMetrics.md#summary
  double sum = 5;

  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  //
  // See the following issue for more context:
  // https://github.com/open-telemetry/opentelemetry-proto/issues/125
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    //
    // Quantile values must NOT be negative.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;

  // Flags that apply to this specific data point.  See DataPointFlags
  // for the available flags and their meaning.
  uint32 flags = 8;
}
```

Per ulteriori informazioni, consulta Traduzioni con formato OpenTelemetry 1.0.0.

Traduzioni con formato OpenTelemetry 1.0.0

CloudWatch esegue alcune trasformazioni per formattare CloudWatch i dati. OpenTelemetry

Traduzione dello spazio dei nomi, del nome parametro e delle dimensioni

Questi attributi sono coppie chiave-valore codificate nella mappatura.

- Un attributo ha la chiave `Namespace` e il suo valore è lo spazio dei nomi del parametro
- Un attributo ha la chiave `MetricName` e il suo valore è il nome del parametro
- Una coppia ha la chiave `Dimensions` e il suo valore è un elenco nidificato di coppie chiave/valore. Ogni coppia in questo elenco è mappata a una dimensione CloudWatch metrica, in cui la chiave della coppia è il nome della dimensione e il suo valore è il valore della dimensione.

Traduzione di media, somma SampleCount, minimo e massimo

Il datapoint di riepilogo consente di CloudWatch esportare tutte queste statistiche utilizzando un punto dati.

- `startTimeUnixNano`contiene il CloudWatch `startTime`
- `timeUnixNano`contiene il CloudWatch `endTime`
- `sum` contiene la statistica Sum (Somma).
- `count`contiene la SampleCount statistica.
- `quantile_values` contiene due oggetti `valueAtQuantile.value`:
  - `valueAtQuantile.quantile = 0.0` con `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` con `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` con `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` con `valueAtQuantile.value = Max value`

Le risorse che utilizzano il flusso metrico possono calcolare la statistica media come Somma/. SampleCount

Unità di traduzione

CloudWatch le unità sono mappate alla variante con distinzione tra maiuscole e minuscole del codice unificato per le unità di misura, come illustrato nella tabella seguente. Per ulteriori informazioni, consulta The Unified Code For Units of Measure.

CloudWatch	OpenTelemetry
Secondo	s

CloudWatch	OpenTelemetry
Secondo o secondi	s
Microsecondi	us
Millisecondi	ms
Byte	By
Kilobyte	kBy
Megabyte	MBy
Gigabyte	GBy
Terabyte	TBy
Bit	bit
Kilobit	kbit
Megabit	MBit
Gigabit	GBit
Terabit	Tbit
Percentuale	%
Conteggio	{Count}
Nessuno	1

Le unità combinate con una barra vengono mappate applicando la conversione di entrambe le OpenTelemetry unità. Ad esempio, Byte/secondo è mappato a By/s.

Come analizzare i messaggi 1.0.0 OpenTelemetry

Questa sezione fornisce informazioni per aiutarti a iniziare con l'analisi della 1.0.0. OpenTelemetry

Innanzitutto, dovresti ottenere associazioni specifiche per la lingua, che ti consentano di analizzare i messaggi 1.0.0 nella tua lingua preferita. OpenTelemetry

Per ottenere associazioni specifiche della lingua

- I passaggi dipendono dalla lingua preferita.
  - Per usare Java, aggiungi la seguente dipendenza Maven al tuo progetto Java: Java >> 0.14.1. OpenTelemetry
  - Per utilizzare qualsiasi altra lingua, procedi come segue:

    a. Assicurati che la lingua in uso sia supportata controllando l'elenco in Generating Your Classes (Generazione delle classi).

    b. Installa il compilatore Protobuf seguendo i passaggi descritti in Download Protocol Buffers (Scarica buffer protocolli).

    c. Scarica le definizioni OpenTelemetry 0.7.0 nella versione di rilascio 1.0.0 ProtoBuf .

    d. Verificate di trovarvi nella cartella principale delle definizioni 0.7.0 scaricate. OpenTelemetry ProtoBuf Quindi crea un cartella `src` e quindi esegui il comando per generare associazioni specifiche della lingua. Per ulteriori informazioni, consulta Generating Your Classes.

    Di seguito è riportato un esempio di come generare associazioni Javascript.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

Nella sezione seguente sono inclusi esempi di utilizzo delle associazioni specifiche della lingua che puoi creare utilizzando le istruzioni precedenti.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
```

```
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }

        return result;
    }
}
```

Javascript

Questo esempio presuppone che la cartella principale con le associazioni generate sia `./`

L'argomento dati della funzione `parseRecord` può essere uno dei seguenti tipi:

- `Uint8Array` è ottimale
- `Buffer` ottimale sotto il nodo
- `Array.`*`number`* numero intero a 8 bit

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')
```

```
function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)

        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

È necessario leggere i delimitatori `var-int` in autonomia o utilizzare i metodi interni `_VarintBytes(size)` e `_DecodeVarint32(buffer, position)`. Questi restituiscono la posizione nel buffer subito dopo i byte di dimensione. Il lato lettura costruisce un nuovo buffer che è limitato alla lettura solo dei byte del messaggio.

```
size = my_metric.ByteSize()
```

```
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Utilizza `Buffer.DecodeMessage()`.

C#

Utilizza `CodedInputStream`. Questa classe può leggere messaggi delimitati da dimensioni.

C++

Le funzioni descritte in `google/protobuf/util/delimited_message_util.h` possono leggere messaggi delimitati dalle dimensioni.

Altre lingue

Per le altre lingue, consulta Download Protocol Buffers.

Nell'implementare il parser, considera che un record Kinesis può contenere più messaggi di buffer di protocollo `ExportMetricsServiceRequest`, ognuno dei quali inizia con un'intestazione con un `UnsignedVarInt32` che indica la lunghezza del record in byte.

OpenTelemetry Formato 0.7.0

OpenTelemetry è una raccolta di strumenti, API e SDK. Puoi utilizzarlo per strumentare, generare, raccogliere ed esportare dati di telemetria (metriche, log e tracce) per l'analisi. OpenTelemetry fa parte della Cloud Native Computing Foundation. Per ulteriori informazioni, vedere OpenTelemetry.

Per informazioni sulla specifica OpenTelemetry 0.7.0 completa, vedere la versione v0.7.0.

Un record Kinesis può contenere una o più strutture di `ExportMetricsServiceRequest` OpenTelemetry dati. Ogni struttura dati inizia con un'intestazione con `UnsignedVarInt32` che indica la lunghezza del record in byte. Ciascuna `ExportMetricsServiceRequest` può contenere dati provenienti da più parametri contemporaneamente.

Di seguito è riportata una rappresentazione in formato stringa del messaggio della struttura `ExportMetricsServiceRequest` OpenTelemetry dati. OpenTelemetry serializza il protocollo binario di Google Protocol Buffers e questo non è leggibile dall'uomo.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "2345678901"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
    }
    attributes {
      key: "aws.exporter.arn"
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  instrumentation_library_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "1"
      double_summary {
        data_points {
          labels {
            key: "Namespace"
            value: "AWS/DynamoDB"
          }
          labels {
```

```
          key: "MetricName"
          value: "ConsumedReadCapacityUnits"
        }
        labels {
          key: "TableName"
          value: "MyTable"
        }
        start_time_unix_nano: 1604948400000000000
        time_unix_nano: 1604948460000000000
        count: 1
        sum: 1.0
        quantile_values {
          quantile: 0.0
          value: 1.0
        }
        quantile_values {
          quantile: 0.95
          value: 1.0
        }
        quantile_values {
          quantile: 0.99
          value: 1.0
        }
        quantile_values {
          quantile: 1.0
          value: 1.0
        }
      }
      data_points {
        labels {
          key: "Namespace"
          value: "AWS/DynamoDB"
        }
        labels {
          key: "MetricName"
          value: "ConsumedReadCapacityUnits"
        }
        labels {
          key: "TableName"
          value: "MyTable"
        }
        start_time_unix_nano: 1604948460000000000
        time_unix_nano: 1604948520000000000
        count: 2
```

```
          sum: 5.0
          quantile_values {
            quantile: 0.0
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
        }
      }
    }
  }
}
```

Oggetto di primo livello per serializzare i dati metrici OpenTelemetry

`ExportMetricsServiceRequest`è il wrapper di primo livello per serializzare un payload di esportazione. OpenTelemetry Contiene uno o più `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics`è l'oggetto di primo livello per rappresentare gli oggetti. MetricData

```
// A collection of InstrumentationLibraryMetrics from a Resource.
message ResourceMetrics {
  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated InstrumentationLibraryMetrics instrumentation_library_metrics = 2;
}
```

Oggetto della risorsa

Un oggetto `Resource` è un oggetto coppia di valori che contiene alcune informazioni sulla risorsa che ha generato i parametri. Per i parametri creati da AWS, la struttura dati contiene l'ARN (Amazon Resource Name) della risorsa correlata al parametro, ad esempio un'istanza EC2 o un bucket S3.

L'oggetto `Resource` contiene un attributo denominato `attributes`, che memorizza un elenco di coppie chiave-valore.

- `cloud.account.id` contiene l'ID dell'account
- `cloud.region` contiene la regione
- `aws.exporter.arn` contiene l'ARN del flusso di parametri
- `cloud.provider` è sempre `aws`.

```
// Resource information.
message Resource {
  // Set of labels that describe the resource.
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

L'oggetto InstrumentationLibraryMetrics

Il campo instrumentation_library non verrà compilato. Compileremo solo il campo dei parametri che stiamo esportando.

```
// A collection of Metrics produced by an InstrumentationLibrary.
message InstrumentationLibraryMetrics {
  // The instrumentation library information for the metrics in this message.
  // If this field is not set then no library info is known.
  opentelemetry.proto.common.v1.InstrumentationLibrary instrumentation_library = 1;
  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;
}
```

L'oggetto Metric

L'oggetto Metric contiene un campo dati `DoubleSummary` che contiene un elenco di `DoubleSummaryDataPoint`.

```
message Metric {
  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  oneof data {
    IntGauge int_gauge = 4;
    DoubleGauge double_gauge = 5;
    IntSum int_sum = 6;
    DoubleSum double_sum = 7;
    IntHistogram int_histogram = 8;
    DoubleHistogram double_histogram = 9;
    DoubleSummary double_summary = 11;
  }
}

message DoubleSummary {
  repeated DoubleSummaryDataPoint data_points = 1;
}
```

L' MetricDescriptor oggetto

L' MetricDescriptor oggetto contiene metadati. Per ulteriori informazioni, vedete metrics.proto on. GitHub

Per gli stream metrici, ha i seguenti contenuti: MetricDescriptor

- `name` sarà `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`*
- `description` sarà vuoto.
- `unit` verrà compilato mappando l'unità del dato del parametro alla variante con distinzione tra maiuscole e minuscole del codice unificato per le unità di misura. Per ulteriori informazioni, consulta Traduzioni in formato OpenTelemetry 0.7.0 and Il codice unificato per le unità di misura.
- `type` sarà `SUMMARY`.

L'oggetto DoubleSummaryDataPoint

L' DoubleSummaryDataPoint oggetto contiene il valore di un singolo punto dati in una serie temporale in una DoubleSummary metrica.

```
// DoubleSummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message DoubleSummaryDataPoint {
  // The set of labels that uniquely identify this timeseries.
  repeated opentelemetry.proto.common.v1.StringKeyValue labels = 1;

  // start_time_unix_nano is the last time when the aggregation value was reset
  // to "zero". For some metric types this is ignored, see data types for more
  // details.
  //
  // The aggregation value is over the time interval (start_time_unix_nano,
  // time_unix_nano].
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  //
  // Value of 0 indicates that the timestamp is unspecified. In that case the
  // timestamp may be decided by the backend.
  fixed64 start_time_unix_nano = 2;

  // time_unix_nano is the moment when this aggregation value was reported.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;

  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
  double sum = 5;

  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  message ValueAtQuantile {
```

```
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;
}
```

Per ulteriori informazioni, consulta Traduzioni in formato OpenTelemetry 0.7.0.

Traduzioni in formato OpenTelemetry 0.7.0

CloudWatch esegue alcune trasformazioni per formattare CloudWatch i dati. OpenTelemetry

Traduzione dello spazio dei nomi, del nome parametro e delle dimensioni

Questi attributi sono coppie chiave-valore codificate nella mappatura.

- Una coppia contiene lo spazio dei nomi del parametro
- Una coppia contiene il nome del parametro
- Per ogni dimensione, CloudWatch memorizza la coppia seguente: `metricDatum.Dimensions[i].`*`Name`*`, metricDatum.Dimensions[i].`*`Value`*

Media di traduzione, somma SampleCount, minimo e massimo

Il datapoint di riepilogo consente di CloudWatch esportare tutte queste statistiche utilizzando un punto dati.

- `startTimeUnixNano`contiene il CloudWatch `startTime`
- `timeUnixNano`contiene il CloudWatch `endTime`
- `sum` contiene la statistica Sum (Somma).
- `count`contiene la SampleCount statistica.
- `quantile_values` contiene due oggetti `valueAtQuantile.value`:
  - `valueAtQuantile.quantile = 0.0` con `valueAtQuantile.value =` *`Min value`*

- `valueAtQuantile.quantile = 0.99` con `valueAtQuantile.value = p99 value`
- `valueAtQuantile.quantile = 0.999` con `valueAtQuantile.value = p99.9 value`
- `valueAtQuantile.quantile = 1.0` con `valueAtQuantile.value = Max value`

Le risorse che utilizzano il flusso metrico possono calcolare la statistica media come Somma/. SampleCount

Unità di traduzione

CloudWatch le unità sono mappate alla variante con distinzione tra maiuscole e minuscole del codice unificato per le unità di misura, come illustrato nella tabella seguente. Per ulteriori informazioni, consulta The Unified Code For Units of Measure.

CloudWatch	OpenTelemetry
Secondo	s
Secondo o secondi	s
Microsecondo	us
Millisecondi	ms
Byte	By
Kilobyte	kBy
Megabyte	MBy
Gigabyte	GBy
Terabyte	TBy
Bit	bit
Kilobit	kbit
Megabit	MBit
Gigabit	GBit

CloudWatch	OpenTelemetry
Terabit	Tbit
Percentuale	%
Conteggio	{Count}
Nessuno	1

Le unità combinate con una barra vengono mappate applicando la conversione di entrambe le OpenTelemetry unità. Ad esempio, Byte/secondo è mappato a By/s.

Come analizzare i messaggi 0.7.0 OpenTelemetry

Questa sezione fornisce informazioni per aiutarti a iniziare con l'analisi della versione 0.7.0. OpenTelemetry

Innanzitutto, dovresti ottenere collegamenti specifici per la lingua, che ti consentano di analizzare i messaggi 0.7.0 nella tua lingua preferita. OpenTelemetry

Per ottenere associazioni specifiche della lingua

- I passaggi dipendono dalla lingua preferita.
  - Per usare Java, aggiungi la seguente dipendenza Maven al tuo progetto Java: Java >> 0.14.1. OpenTelemetry
  - Per utilizzare qualsiasi altra lingua, procedi come segue:
    a. Assicurati che la lingua in uso sia supportata controllando l'elenco in Generating Your Classes (Generazione delle classi).
    b. Installa il compilatore Protobuf seguendo i passaggi descritti in Download Protocol Buffers (Scarica buffer protocolli).
    c. Scarica le definizioni OpenTelemetry 0.7.0 nella versione v0.7.0 ProtoBuf .
    d. Conferma di trovarti nella cartella principale delle definizioni 0.7.0 scaricate. OpenTelemetry ProtoBuf Quindi crea un cartella `src` e quindi esegui il comando per generare associazioni specifiche della lingua. Per ulteriori informazioni, consulta Generating Your Classes.

Di seguito è riportato un esempio di come generare associazioni Javascript.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

Nella sezione seguente sono inclusi esempi di utilizzo delle associazioni specifiche della lingua che puoi creare utilizzando le istruzioni precedenti.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }
```

```
        return result;
    }
}
```

Javascript

Questo esempio presuppone che la cartella principale con le associazioni generate sia `./`

L'argomento dati della funzione `parseRecord` può essere uno dei seguenti tipi:

- `Uint8Array` è ottimale
- `Buffer` ottimale sotto il nodo
- `Array.number` numero intero a 8 bit

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')

function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)
```

```
        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

È necessario leggere i delimitatori `var-int` in autonomia o utilizzare i metodi interni `_VarintBytes(size)` e `_DecodeVarint32(buffer, position)`. Questi restituiscono la posizione nel buffer subito dopo i byte di dimensione. Il lato lettura costruisce un nuovo buffer che è limitato alla lettura solo dei byte del messaggio.

```
size = my_metric.ByteSize()
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Utilizza `Buffer.DecodeMessage()`.

C#

Utilizza `CodedInputStream`. Questa classe può leggere messaggi delimitati da dimensioni.

C++

Le funzioni descritte in `google/protobuf/util/delimited_message_util.h` possono leggere messaggi delimitati dalle dimensioni.

Altre lingue

Per le altre lingue, consulta Download Protocol Buffers.

Nell'implementare il parser, considera che un record Kinesis può contenere più messaggi di buffer di protocollo `ExportMetricsServiceRequest`, ognuno dei quali inizia con un'intestazione con un `UnsignedVarInt32` che indica la lunghezza del record in byte.

Risoluzione dei problemi

Se non visualizzi i dati dei parametri nella destinazione finale, controlla quanto segue:

- Verifica che il flusso di parametri sia in esecuzione. Per istruzioni su come utilizzare la CloudWatch console a tale scopo, consulta. Funzionamento e manutenzione del flusso di parametri
- Le metriche pubblicate più di due giorni negli ultimi due giorni non vengono trasmesse in streaming. Per determinare se una determinata metrica verrà trasmessa in streaming, crea un grafico della metrica nella CloudWatch console e controlla quanti anni ha l'ultimo datapoint visibile. Se sono trascorsi più di due giorni, non verrà rilevata dai flussi metrici.
- Verifica i parametri emessi dal flusso di parametri. Nella CloudWatch console, in Metriche, MetricStreams esamina lo spazio dei nomi AWS/CloudWatch/per le MetricUpdatemetriche, e. TotalMetricUpdatePublishErrorRate
- Se la PublishErrorRatemetrica è alta, verificate che la destinazione utilizzata dal flusso di distribuzione di Firehose esista e che il ruolo IAM specificato nella configurazione del flusso di metriche conceda al servizio principale `CloudWatch` le autorizzazioni di scrittura su di essa. Per ulteriori informazioni, consulta Trust between CloudWatch e Firehose.
- Verificate che lo stream di distribuzione di Firehose sia autorizzato a scrivere nella destinazione finale.
- Nella console Firehose, visualizzate il flusso di distribuzione Firehose utilizzato per il flusso metrico e controllate la scheda Monitoring per vedere se il flusso di distribuzione Firehose sta ricevendo dati.
- Conferma di aver configurato il flusso di distribuzione di Firehose con i dettagli corretti.
- Controllate tutti i log o le metriche disponibili per la destinazione finale in cui il flusso di distribuzione di Firehose scrive.
- Per ottenere informazioni più dettagliate, abilitate la registrazione CloudWatch degli errori dei log nel flusso di distribuzione di Firehose. Per ulteriori informazioni, consulta Monitoring Amazon Data Firehose tramite CloudWatch log.

Visualizzazione di parametri disponibili

I parametri sono raggruppati in primo luogo in base allo spazio dei nomi e in secondo luogo in base alle diverse combinazioni di dimensioni all'interno di ciascuno spazio dei nomi. Ad esempio, puoi visualizzare tutti i parametri EC2, i parametri EC2 raggruppati per istanza o i parametri EC2 raggruppati per gruppo Auto Scaling.

Solo i AWS servizi che utilizzi inviano i parametri ad Amazon CloudWatch.

Per un elenco di AWS servizi a cui inviano metriche CloudWatch, consulta. AWS servizi che pubblicano CloudWatch metriche Da questa pagina è inoltre possibile visualizzare i parametri e le dimensioni pubblicate da ciascuno di questi servizi.

> **Note**
>
> I parametri che non hanno ricevuto nuovi punti di dati nelle ultime due settimane non vengono visualizzati nella console. Inoltre, non vengono visualizzati quando digiti il nome del parametro o i nomi delle dimensioni nella casella di ricerca della scheda Tutti i parametri della console e non vengono restituiti nei risultati del comando list-metrics . Il modo migliore per recuperare queste metriche è utilizzare get-metric-statisticsi comandi get-metric-datao in. AWS CLI
>
> Se il parametro precedente che vuoi visualizzare ha un parametro corrente con dimensioni simili, puoi visualizzare questo parametro corrente simile, quindi scegli la scheda Origine e modifichi il nome del parametro e i campi di dimensione come desideri, nonché modificare l'intervallo di tempo sull'ora in cui il parametro è stato restituito.

I seguenti passaggi aiutano a sfogliare gli spazi dei nomi dei parametri per trovare e visualizzare i parametri. È inoltre possibile ricercare parametri utilizzando termini di ricerca mirati. Per ulteriori informazioni, consulta Ricerca di parametri disponibili.

Se stai navigando in un account configurato come account di monitoraggio in modalità osservabile su CloudWatch più account, puoi visualizzare le metriche degli account di origine collegati a questo account di monitoraggio. Quando vengono visualizzati le metriche degli account di origine, viene visualizzato anche l'ID o l'etichetta dell'account da cui provengono. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Visualizzazione dei parametri disponibili per spazio dei nomi e dimensione tramite la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), All metrics (Tutti i parametri).
3. Seleziona uno spazio dei nomi del parametro (ad esempio, EC2 o Lambda).
4. Seleziona una dimensione del parametro, ad esempio, Per-Instance Metrics (Parametri per istanza) o By Function Name (Per nome della funzione).
5. La scheda Browse (Sfoglia) mostra tutti i parametri per tale dimensione nello spazio dei nomi. Accanto al nome di ogni parametro è presente un pulsante informativo che consente di visualizzare un popup con la definizione del parametro.

   Se si tratta di un account di monitoraggio in CloudWatch modalità osservabile su più account, vengono visualizzate anche le metriche degli account di origine collegati a questo account di monitoraggio. Le colonne Account label (Etichetta dell'account) e Account id (ID account) nella tabella mostrano l'account di provenienza di ogni parametro.

   Puoi eseguire le operazioni indicate di seguito:

   a. Per ordinare la tabella, utilizza l'intestazione della colonna.
   b. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.
   c. Per filtrare per account, scegli l'etichetta o l'ID dell'account, quindi scegli Add to search (Aggiungi alla ricerca).
   d. Per filtrare per risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).
   e. Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).
6. (Facoltativo) Per aggiungere questo grafico a una CloudWatch dashboard, scegli Azioni, Aggiungi alla dashboard.

Per visualizzare le metriche disponibili in base allo spazio dei nomi, alla dimensione o alla metrica dell'account, utilizzando il AWS CLI

Utilizza il comando list-metrics per elencare le metriche. CloudWatch Per un elenco degli spazi dei nomi, dei parametri e delle dimensioni di tutti i servizi che pubblicano parametri, consulta AWS servizi che pubblicano CloudWatch metriche.

Il comando di esempio seguente elenca tutte le metriche per Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

Di seguito è riportato un output di esempio.

```
{
  "Metrics" : [
    ...
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkOut"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "CPUUtilization"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkIn"
    },
    ...
  ]
}
```

Elenco di tutti i parametri disponibili per una risorsa specificata

L'esempio seguente specifica lo spazio dei nomi `AWS/EC2` e la dimensione `InstanceId` per visualizzare i risultati solo per l'istanza specificata.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
 Name=InstanceId,Value=i-1234567890abcdef0
```

Elenco di un parametro per tutte le risorse

L'esempio seguente specifica lo spazio dei nomi `AWS/EC2` e un nome parametro per visualizzare i risultati solo per il parametro specificato.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Per recuperare le metriche dagli account di origine collegati in modo osservabile su più account CloudWatch

L'esempio seguente viene eseguito in un account di monitoraggio per recuperare le metriche sia dall'account di monitoraggio che da tutti gli account di origine collegati. Se non aggiungi `--include-linked-accounts`, il comando restituisce solo le metriche dell'account di monitoraggio.

```
aws cloudwatch list-metrics --include-linked-accounts
```

Per recuperare le metriche da un account di origine in modalità osservabilità tra account CloudWatch

L'esempio seguente viene eseguito in un account di monitoraggio per recuperare le metriche dall'account di origine con l'ID 111122223333.

```
aws cloudwatch list-metrics --include-linked-accounts --owning-account "111122223333"
```

Ricerca di parametri disponibili

Puoi effettuare una ricerca all'interno di tutti i parametri nell'account utilizzando termini di ricerca mirati. Vengono restituiti parametri aventi risultati corrispondenti all'interno del proprio spazio dei nomi, nome parametro o dimensioni.

Se si tratta di un account di monitoraggio in modalità CloudWatch osservabile su più account, cerca anche le metriche dagli account di origine collegati a questo account di monitoraggio.

> **Note**
>
> I parametri che non hanno ricevuto nuovi punti di dati nelle ultime due settimane non vengono visualizzati nella console. Inoltre, non vengono visualizzati quando digiti il nome del parametro o i nomi delle dimensioni nella casella di ricerca della scheda Tutti i parametri della console e non vengono restituiti nei risultati del comando list-metrics . Il modo migliore per recuperare queste metriche è utilizzare i comandi o in. get-metric-dataget-metric-statistics AWS CLI

Per cercare le metriche disponibili in CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nel campo di ricerca della scheda All metrics (Tutti i parametri), immetti un termine di ricerca, ad esempio un nome parametro, uno spazio dei nomi, un ID account, un'etichetta dell'account, un nome o un valore della dimensione oppure un nome di risorsa. Verranno visualizzati tutti gli spazi dei nomi aventi parametri con tale termine di ricerca.

   Ad esempio, eseguendo una ricerca di **`volume`**, verranno visualizzati gli spazi dei nomi contenenti parametri con questo termine nel nome.

   Per ulteriori informazioni sulla ricerca, consulta Utilizzo delle espressioni di ricerca nei grafici
4. Per visualizzare in un grafico tutti i risultati di ricerca, scegli Graph search (Ricerca del grafico)

   oppure

   Seleziona uno spazio dei nomi per visualizzarne i relativi parametri. A questo punto puoi effettuare le seguenti operazioni:

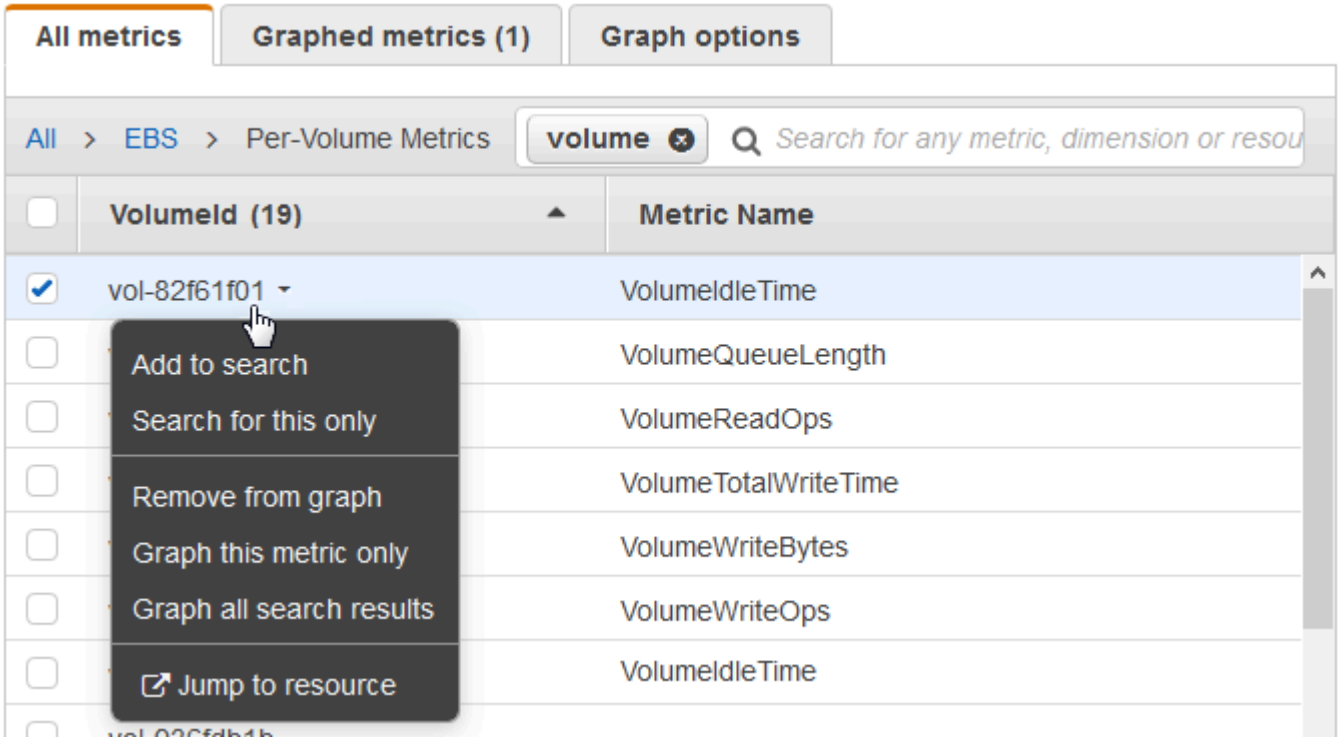
   a. Per rappresentare graficamente uno o più parametri, seleziona la casella di controllo accanto a ciascun parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.

   b. Per delimitare la ricerca, passa il mouse su un nome parametro e scegli Add to search (Aggiungi alla ricerca) o Search for this only (Cerca solo questo).

   c. Per visualizzare una delle risorse nella relativa console, scegli l'ID risorsa, quindi Jump to resource (Salta alla risorsa).

d. Per visualizzare la guida di un parametro, seleziona il nome parametro, quindi seleziona What is this? (Che cos'è questo?).

I parametri selezionati vengono visualizzati nel grafico.



5. (Opzionale) Seleziona uno dei pulsanti nella barra di ricerca per modificare tale parte del termine di ricerca.

Rappresentazione grafica dei parametri

Usa la CloudWatch console per rappresentare graficamente i dati metrici generati da altri AWS servizi. In questo modo si osserva con maggiore efficienza l'attività dei parametri nei servizi. Le seguenti procedure descrivono come rappresentare graficamente le metriche in. CloudWatch

Indice

- Modifica dell'asse Y di un grafico
- Creazione di un allarme a partire da un parametro in un grafico

Rappresentazione grafica di un parametro

Puoi selezionare le metriche e creare grafici dei dati metrici utilizzando la console. CloudWatch

CloudWatch supporta le seguenti statistiche sulle metriche:`Average`,,`Minimum`, `Maximum` e. `Sum` `SampleCount` Per ulteriori informazioni, consulta Statistiche.

Puoi visualizzare i dati a diversi livelli di dettaglio. Ad esempio, puoi scegliere la visualizzazione di un minuto, che può essere utile durante la risoluzione dei problemi. In alternativa, scegli la visualizzazione meno dettagliata di un'ora. Può essere utile quando si visualizza un intervallo di tempo più ampio (ad esempio 3 giorni) in modo da poter vedere le tendenze nel tempo. Per ulteriori informazioni, consulta Periodi.

Se utilizzi un account configurato come account di monitoraggio in modalità osservabile CloudWatch tra più account, puoi rappresentare graficamente le metriche degli account di origine collegati a questo account di monitoraggio. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Creazione di un grafico

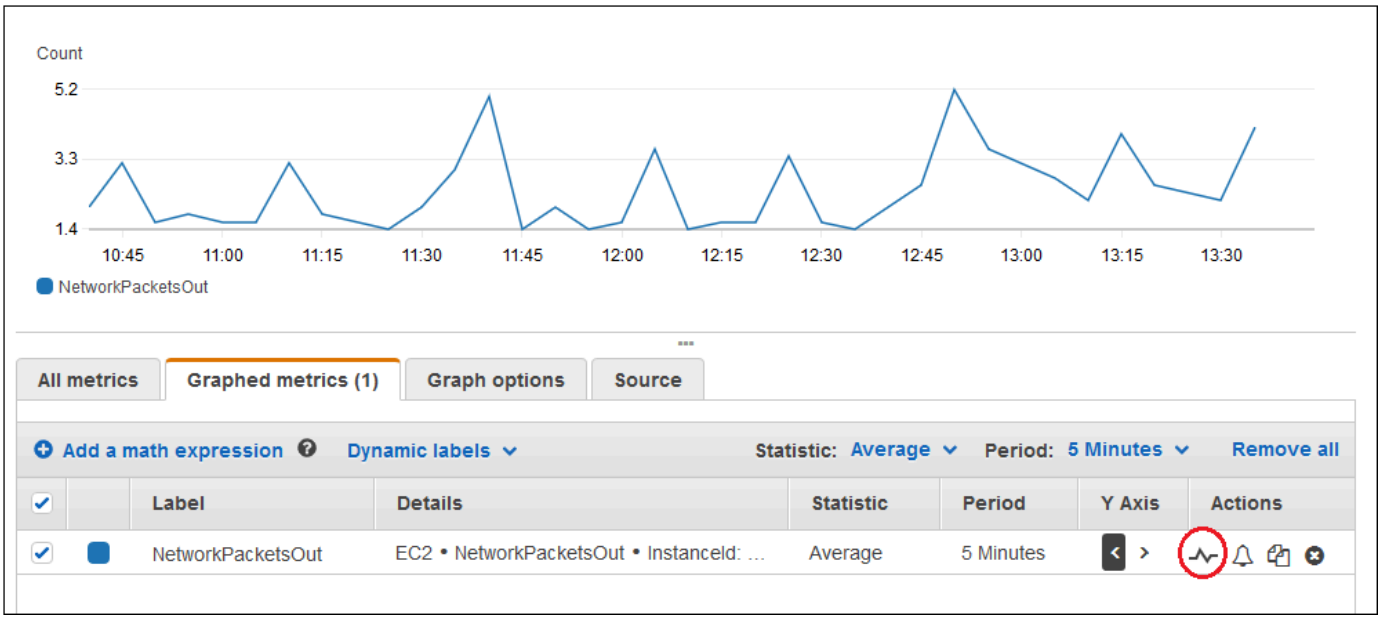
Rappresentazione grafica di un parametro

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), All metrics (Tutti i parametri).
3. Nella scheda Tutti i parametri, inserisci un termine nel campo di ricerca, ad esempio un nome del parametro, un ID account o un nome della risorsa.

   Ad esempio, se si ricerca il parametro `CPUUtilization`, verranno visualizzati gli spazi dei nomi e le dimensioni con questo parametro.
4. Seleziona uno dei risultati della ricerca per visualizzare i parametri.
5. Per rappresentare graficamente uno o più parametri, seleziona la casella di controllo accanto a ciascun parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.

6. (Facoltativo) Per modificare il tipo di grafico, scegli la scheda Opzioni. È quindi possibile scegliere tra un grafico a linee, un grafico ad area in pila, una visualizzazione numerica, un indicatore, un grafico a barre o un grafico a torta.
7. Seleziona la scheda Graphed metrics (Parametri nel grafico).
8. (Facoltativo) Per modificare la statistica utilizzata nel grafico, scegli la nuova statistica nella colonna Statistic (Statistica) accanto al nome del parametro.

   Per ulteriori informazioni sulle CloudWatch statistiche, vedereCloudWatch definizioni statistiche. Per ulteriori informazioni sulle statistiche percentili pxx, consulta Percentili.
9. (Facoltativo) Per aggiungere un intervallo per il rilevamento delle anomalie contenente i valori previsti per il parametro, scegli l'icona relativa alla funzionalità di rilevamento di anomalie in Actions (Operazioni) accanto al parametro.



   CloudWatch utilizza fino a due settimane dei dati storici recenti della metrica per calcolare un modello per i valori previsti. Quindi visualizza questo intervallo di valori attesi come una banda sul grafico. CloudWatch aggiunge una nuova riga sotto la metrica per visualizzare l'espressione matematica della banda di rilevamento delle anomalie, denominata ANOMALY_DETECTION_BAND. Se esistono dati storici recenti, puoi visualizzare immediatamente una banda di rilevamento di anomalie di anteprima, che è un'approssimazione della banda di rilevamento di anomalie generata dal modello. Sono necessari fino a 15 minuti per visualizzare la banda di rilevamento di anomalie effettiva.

Per impostazione predefinita, CloudWatch crea i limiti superiore e inferiore della banda di valori previsti con un valore predefinito di 2 per la soglia di banda. Per modificare questo numero, modificare il valore alla fine della formula in Details (Dettagli) per l'intervallo specifico.

- (Facoltativo) Scegli Edit model (Modifica modello) per modificare la modalità di calcolo del modello di rilevamento delle anomalie. È possibile escludere l'utilizzo di periodi di tempo passati e futuri nella formazione per il calcolo del modello. È fondamentale escludere dai dati di formazione eventi insoliti quali interruzione del sistema, distribuzioni e festività. È inoltre possibile specificare il fuso orario da utilizzare per il modello per le modifiche all'ora legale.

Per ulteriori informazioni, consulta Utilizzo del CloudWatch rilevamento delle anomalie.

Per nascondere il modello dal grafico, rimuovi il segno di spunta dalla riga con la funzione `ANOMALY_DETECTION_BAND` o scegli l'icona `X`. Per eliminare completamente il modello, scegli Edit model (Modifica modello), Delete model (Elimina modello).

10. (Facoltativo) Quando si scelgono i parametri da riprodurre in un grafico, specificare che venga visualizzata un'etichetta dinamica sulla legenda del grafico per ogni parametro. Le etichette dinamiche visualizzano una statistica sul parametro e vengono aggiornate automaticamente quando il pannello di controllo o il grafico viene aggiornato. Per aggiungere un'etichetta dinamica, scegli Parametri definiti, Etichette dinamiche.

    Per impostazione predefinita, i valori dinamici aggiunti all'etichetta vengono visualizzati all'inizio dell'etichetta. Puoi quindi fare clic sul valore Label (Etichetta) per il parametro per modificare l'etichetta. Per ulteriori informazioni, consulta Utilizzo di etichette dinamiche.

11. Per visualizzare ulteriori informazioni sul parametro rappresentato graficamente, passare il mouse sulla legenda.

12. Annotazioni orizzontali possono essere utili agli utenti per visualizzare in modo più efficiente quando un parametro è aumentato fino a un determinato livello o se il parametro si trova all'interno di un intervallo predefinito. Per aggiungere un'annotazione orizzontale, scegli Opzioni del grafico, Aggiungi annotazione orizzontale:

    a. In Label (Etichetta), immetti un'etichetta per l'annotazione.

    b. In Value (Valore), immetti un valore del parametro in cui è visualizzata l'annotazione orizzontale.

    c. In Fill (Riempi), specifica se utilizzare un'ombreggiatura di riempimento con questa annotazione. Ad esempio, seleziona `Above` o `Below` per l'area corrispondente da riempire.

Se specifichi `Between`, viene visualizzato un altro campo `Value` e viene riempita l'area del grafico tra i due valori.

d. Se il grafico include più parametri, in Axis (Asse), specifica se i numeri di `Value` fanno riferimento al parametro associato all'asse Y sinistro o all'asse Y destro.

Puoi modificare il colore di riempimento di un'annotazione selezionando il quadrato colori nella colonna sinistra dell'annotazione.

Ripeti queste fasi per aggiungere più annotazioni orizzontali allo stesso grafico.

Per nascondere un'annotazione, deseleziona la casella di controllo nella colonna sinistra per tale annotazione.

Per cancellare un'annotazione, seleziona x nella colonna Actions (Operazioni).

13. Per ottenere un URL del grafico, seleziona Actions (Operazioni), Share (Condividi). Copiare l'URL da salvare o condividere.
14. Per aggiungere il grafico a un pannello di controllo, seleziona Actions (Operazioni), Add to dashboard (Aggiungi a pannello di controllo).

Creazione di un grafico dei parametri da un'altra origine dati

È possibile creare un grafico che mostri risorse provenienti da fonti di dati diverse CloudWatch da. Per ulteriori informazioni sulla creazione di connessioni a queste altre origini dati, consulta Recupero dei parametri da altre origini dati.

Per creare un grafico dei parametri da un'altra origine dati

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), All metrics (Tutti i parametri).
3. Scegli la scheda Query da più origini.
4. Per Origine dati, seleziona l'origine dati che desideri utilizzare.

Se non hai già creato una connessione all'origine dati che desideri, seleziona Crea e gestisci di origini dati, quindi scegli Crea e gestisci origini dati. Per informazioni sul resto della procedura per la creazione dell'origine dati, consulta Connessione a un'origine dati predefinita con una procedura guidata.

5. La procedura guidata o l'editor di query richiede le informazioni necessarie per la query. Il flusso di lavoro è diverso per ogni origine dati ed è personalizzato in base a essa. Ad esempio, per le origini dati Amazon Managed Service for Prometheus e le origini dati Prometheus, viene visualizzata una casella di editor di query PromQL con una guida alle query.
6. Quando hai completato la creazione della query, scegli Query a grafo.

   Il grafico viene compilato con i parametri della query.
7. (Facoltativo) Annotazioni orizzontali possono essere utili agli utenti per visualizzare in modo più efficiente quando un parametro è aumentato fino a un determinato livello o se il parametro si trova all'interno di un intervallo predefinito. Per aggiungere un'annotazione orizzontale, scegli Opzioni del grafico, Aggiungi annotazione orizzontale:

   a. In Label (Etichetta), immetti un'etichetta per l'annotazione.
   b. In Value (Valore), immetti un valore del parametro in cui è visualizzata l'annotazione orizzontale.
   c. In Fill (Riempi), specifica se utilizzare un'ombreggiatura di riempimento con questa annotazione. Ad esempio, seleziona `Above` o `Below` per l'area corrispondente da riempire. Se specifichi `Between`, viene visualizzato un altro campo `Value` e viene riempita l'area del grafico tra i due valori.
   d. Se il grafico include più parametri, in Axis (Asse), specifica se i numeri di `Value` fanno riferimento al parametro associato all'asse Y sinistro o all'asse Y destro.

      Puoi modificare il colore di riempimento di un'annotazione selezionando il quadrato colori nella colonna sinistra dell'annotazione.

   Ripeti queste fasi per aggiungere più annotazioni orizzontali allo stesso grafico.

   Per nascondere un'annotazione, deseleziona la casella di controllo nella colonna sinistra per tale annotazione.

   Per cancellare un'annotazione, seleziona x nella colonna Actions (Operazioni).
8. (Facoltativo) Per aggiungere il grafico a un pannello di controllo, seleziona Operazioni, Aggiungi a pannello di controllo.

Aggiornamento di un grafico

Aggiornamento del tuo grafico

1. Per modificare il nome del grafico, seleziona l'icona a forma di matita.
2. Per modificare l'intervallo di tempo, seleziona uno dei valori predefiniti o scegli custom (personalizzato). Per ulteriori informazioni, consulta Modifica dell'intervallo di tempo o del formato del fuso orario di un grafico.
3. Per modificare la statistica, seleziona la scheda Graphed metrics (Parametri nel grafico). Scegli l'intestazione di colonna o un valore singolo, quindi seleziona una delle statistiche o dei percentili predefiniti oppure specifica un percentile personalizzato (ad esempio, **`p95.45`**).
4. Per modificare il periodo, seleziona la scheda Graphed metrics (Parametri nel grafico). Seleziona l'intestazione di colonna o un singolo valore, quindi scegli un valore diverso.
5. Per aggiungere un'annotazione orizzontale, scegli Graph options (Opzioni del grafico), Add horizontal annotation (Aggiungi annotazione orizzontale):

   a. In Label (Etichetta), immetti un'etichetta per l'annotazione.

   b. In Value (Valore), immetti un valore del parametro in cui è visualizzata l'annotazione orizzontale.

   c. In Fill (Riempi), specifica se utilizzare un'ombreggiatura di riempimento con questa annotazione. Ad esempio, seleziona `Above` o `Below` per l'area corrispondente da riempire. Se specifichi `Between`, viene visualizzato un altro campo `Value` e viene riempita l'area del grafico tra i due valori.

   d. Se il grafico include più parametri, in Axis (Asse), specificare se i numeri di `Value` fanno riferimento al parametro associato all'asse y sinistro o all'asse y destro.

      Puoi modificare il colore di riempimento di un'annotazione selezionando il quadrato colori nella colonna sinistra dell'annotazione.

   Ripeti queste fasi per aggiungere più annotazioni orizzontali allo stesso grafico.

   Per nascondere un'annotazione, deseleziona la casella di controllo nella colonna sinistra per tale annotazione.

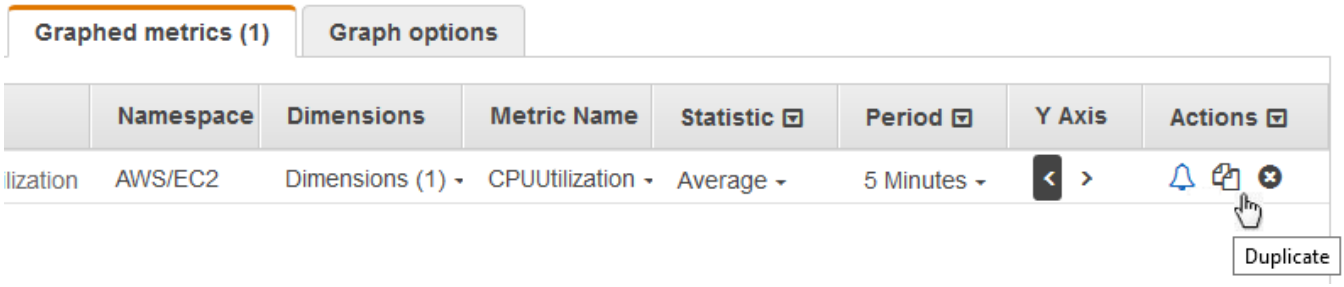
   Per cancellare un'annotazione, seleziona x nella colonna Actions (Operazioni).

6. Per modificare l'intervallo di aggiornamento, scegli Refresh options (Opzioni di aggiornamento), quindi Auto refresh (Aggiornamento automatico) oppure scegli 1 Minute (1 minuto), 2 Minutes (2 minuti), 5 Minutes (5 minuti) o 15 Minutes (15 minuti).

Duplicazione di un parametro

Duplicazione di un parametro

1. Seleziona la scheda Graphed metrics (Parametri nel grafico).
2. In Actions (Operazioni), seleziona l'icona Duplicate (Duplica).



3. Aggiorna il parametro duplicato in base alle esigenze.

Unisci due grafici in uno

Puoi unire due grafici diversi in uno solo, dove il grafico risultante mostrerà entrambi i parametri. Ciò può essere utile se hai già diversi parametri visualizzati in grafici diversi e desideri combinarli oppure se desideri creare facilmente un unico grafico con parametri di diverse regioni.

Per unire un grafico in un altro, utilizza l'URL o l'origine JSON del grafico in cui desideri unirli.

Unione di due grafici in uno

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Apri il grafico da unire in un altro grafico. A tale scopo, puoi scegliere Parametri, Tutti i parametri, quindi scegliere un parametro da rappresentare graficamente. Oppure puoi aprire un pannello di controllo e quindi aprire uno dei grafici nel pannello selezionando il grafico e scegliendo Apri nei parametri dal menu in alto a destra del grafico.
3. Dopo aver aperto un grafico, completa una delle seguenti operazioni:.
   - Copia l'URL dalla barra del browser.
   - Scegli la scheda Origine, quindi seleziona Copia.

4. Apri il grafico in cui unire il grafico precedente.
5. Con il secondo grafico aperto nella vista Parametri, scegli Operazioni, Unisci grafico.
6. Inserisci l'URL o il codice JSON che hai copiato in precedenza e scegli Unisci.
7. Vengono visualizzati i grafici uniti. L'asse y a sinistra è per il grafico originale e l'asse y a destra è per il grafico che vi è stato unito.

> **Note**
>
> Se il grafico in cui hai effettuato l'unione utilizza la funzione METRICS(), i parametri del grafico che sono stati uniti non sono inclusi nel calcolo METRICS() nel grafico unito.

8. Per salvare il grafico unito a un pannello di controllo, seleziona Operazioni, Aggiungi a pannello di controllo.

Utilizzo di etichette dinamiche

Puoi utilizzare etichette dinamiche con i tuoi grafici. Le etichette dinamiche aggiungono un valore aggiornato in modo dinamico all'etichetta per il parametro selezionato. Puoi aggiungere un ampio intervallo di valori alle etichette, come mostrato nelle tabelle seguenti.

Il valore dinamico mostrato nell'etichetta è derivato dall'intervallo di tempo attualmente visualizzato sul grafico. La parte dinamica dell'etichetta viene aggiornata automaticamente quando il pannello di controllo o il grafico viene aggiornato.

Se utilizzi un'etichetta dinamica con un'espressione di ricerca, l'etichetta dinamica si applica a ogni parametro restituito dalla ricerca.

È possibile utilizzare la CloudWatch console per aggiungere un valore dinamico a un'etichetta, modificare l'etichetta, modificare la posizione del valore dinamico all'interno della colonna dell'etichetta e apportare altre personalizzazioni.

Etichette dinamiche

All'interno di un'etichetta dinamica, è possibile utilizzare i seguenti valori relativi alle proprietà del parametro:

Valore dinamico live dell'etic hetta	Descrizione
${AVG}	La media dei valori nell'intervallo di tempo attualmente mostrato nel grafico.
${DATAPOINT_COUNT}	Il numero di punti dati nell'intervallo di tempo attualmente mostrato nel grafico.
${FIRST}	Il più vecchio dei valori del parametro nell'intervallo di tempo che è attualmente mostrato nel grafico.
${FIRST_LAST_RANGE}	Differenza tra i valori dei parametri dei punti dati più vecchi e quelli più recenti attualmente visualizzati nel grafico.
${FIRST_LAST_TIME_ RANGE}	L'intervallo di tempo assoluto tra i punti dati più vecchi e quelli più recenti attualmente visualizzati nel grafico.
${FIRST_TIME}	Il timestamp del punto dati meno recente nell'intervallo di tempo attualmente mostrato nel grafico.
${FIRST_TIME_RELATIVE}	La differenza temporale assoluta tra ora e timestamp del punto dati più vecchio nell'intervallo di tempo attualmente visualizzato nel grafico.
${LABEL}	Rappresentazione dell'etichetta predefinita per un parametro.
${LAST}	Il più recente dei valori del parametro nell'intervallo di tempo che è attualmente mostrato nel grafico.
${LAST_TIME}	Il timestamp del punto dati più recente nell'intervallo di tempo attualmente mostrato nel grafico.
${LAST_TIME_RELATIVE}	La differenza temporale assoluta tra ora e timestamp del punto dati più nuovo nell'intervallo di tempo attualmente visualizzato nel grafico.
${MAX}	Il massimo dei valori nell'intervallo di tempo attualmente mostrato nel grafico.

Valore dinamico live dell'etic hetta	Descrizione
$ {MAX_TIME}	Timestamp del punto dati con il valore del parametro più alto, dei punti dati attualmente visualizzati nel grafico.
${MAX_TIME_RELATIVE}	La differenza assoluta di tempo tra ora e timestamp del punto dati con il valore più alto, di quei punti dati attualmente mostrati nel grafico.
${MIN}	Il minimo dei valori nell'intervallo di tempo attualmente mostrato nel grafico.
${MIN_MAX_RANGE}	Differenza nei valori dei parametri tra i punti dati con i valori del parametro più alti e più bassi, dei punti dati attualmente visualizz ati nel grafico.
${MIN_MAX_TIME_RANGE}	Intervallo di tempo assoluto tra i punti dati con i valori del parametro più alti e più bassi, dei punti dati attualmente visualizz ati nel grafico.
${MIN_TIME}	Timestamp del punto dati con il valore del parametro più basso, dei punti dati attualmente visualizzati nel grafico.
${MIN_TIME_RELATIVE}	La differenza assoluta di tempo tra ora e timestamp del punto dati con il valore più basso, di quei punti dati attualmente mostrati nel grafico.
$ {PROP ('AccountId')}	L'ID AWS dell'account della metrica.
$ {PROP ('AccountLabel')}	L'etichetta specificata per l'account di origine che possiede questa metrica, in termini di osservabilità CloudWatch tra account.
${PROP('Dim.*dimension_name* ')}	Il valore della dimensione specificata. Sostituisci *dimension_name* con il nome della dimensione, distinguendo tra maiuscole e minuscole.
$ {PROP ('')} MetricName	Nome del parametro.

Valore dinamico live dell'etic hetta	Descrizione
${PROP('Namespace')}	Namespace del parametro.
${PROP('Period')}	Il periodo del parametro in secondi.
${PROP('Region')}	La AWS regione in cui viene pubblicata la metrica.
${PROP('Stat')}	Statistica del parametro rappresentata graficamente.
${SUM}	La somma dei valori nell'intervallo di tempo attualmente mostrato nel grafico.

Ad esempio, supponiamo di disporre di un'espressione di ricerca **`SEARCH(' {AWS/Lambda, FunctionName} Errors ', 'Sum')`** che individua la `Errors` per ogni funzione Lambda. Se imposti l'etichetta su `[max: ${MAX} Errors for Function Name ${LABEL}]`, l'etichetta per ogni parametro è [max: ***number*** Errors for Function Name ***Name***].

È possibile aggiungere un massimo di sei valori dinamici a un'etichetta. Puoi utilizzare il segnaposto `${LABEL}` solo una volta all'interno di ogni etichetta.

Modifica dell'intervallo di tempo o del formato del fuso orario di un grafico

Questa sezione descrive come modificare il formato di data, ora e fuso orario su un grafico CloudWatch delle metriche. Descrive inoltre come eseguire lo zoom avanti su un grafico per applicare un intervallo di tempo specifico. Per informazioni sulla creazione di un grafico, consulta Rappresentazione grafica di un parametro.

> **Note**
>
> Se l'intervallo di tempo di un pannello di controllo è più breve del periodo utilizzato per un grafico sul pannello di controllo, accade quanto segue:
>
> - Il grafico viene modificato per visualizzare la quantità di dati corrispondente a un periodo completo per quel widget, anche se questo è più lungo dell'intervallo di tempo del pannello di controllo. Ciò garantisce che nel grafico sia presente almeno un punto dati.

- L'ora di inizio del periodo per questo punto dati viene regolata all'indietro per garantire che sia possibile visualizzare almeno un punto dati.

Impostazione di un intervallo di tempo relativo

New interface

Specificazione di un intervallo di tempo relativo per un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri. Nell'angolo in alto a destra dello schermo, è possibile selezionare uno degli intervalli di tempo predefiniti, che partono da 1 ora fino a 1 settimana (1h, 3h, 12h, 1d, 3d, oppure 1w). In alternativa, è possibile scegliere Personalizza per impostare il tuo intervallo di tempo.
3. Scegli Personalizza e quindi seleziona la sceda Relativo nell'angolo in alto a sinistra del box. È possibile specificare un intervallo di tempo in Minuti, Ore, Giorni, Settimane, Mesi.
4. Dopo aver specificato un intervallo di tempo, scegli Applica.

Original interface

Specificazione di un intervallo di tempo relativo per un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri. Nell'angolo in alto a destra dello schermo, è possibile selezionare uno degli intervalli di tempo predefiniti, che partono da 1 ora fino a 1 settimana (1h, 3h, 12h, 1d, 3d, oppure 1w). In alternativa, è possibile scegliere personalizza per impostare il tuo intervallo di tempo.
3. Scegli Personalizza e quindi seleziona Relativo nell'angolo in alto a sinistra del box. È possibile specificare un intervallo di tempo in Minuti, Ore, Giorni, Settimane, Mesi.

Impostazione di un intervallo di tempo assoluto

New interface

Specificazione di un intervallo di tempo assoluto per un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri. Nell'angolo in alto a destra dello schermo, è possibile selezionare uno degli intervalli di tempo predefiniti, che partono da 1 ora fino a 1 settimana (1h, 3h, 12h, 1d, 3d, oppure 1w). In alternativa, è possibile scegliere Personalizza per impostare il tuo intervallo di tempo.
3. Scegli Personalizza e quindi seleziona la scheda Assoluto nell'angolo in alto a sinistra del box. Utilizza la selezione calendario o i campi di testo per specificare l'intervallo di tempo.
4. Dopo aver specificato un intervallo di tempo, scegli Applica.

Original interface

Specificazione di un intervallo di tempo assoluto per un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri. Nell'angolo in alto a destra dello schermo, è possibile selezionare uno degli intervalli di tempo predefiniti, che partono da 1 ora fino a 1 settimana (1h, 3h, 12h, 1d, 3d, oppure 1w). In alternativa, è possibile scegliere personalizza per impostare il tuo intervallo di tempo.
3. Scegli personalizza e quindi seleziona Assoluto nell'angolo in alto a sinistra del box. Utilizza la selezione calendario o i campi di testo per specificare l'intervallo di tempo.
4. Dopo aver specificato un intervallo di tempo, scegli Applica.

Impostazione del formato del fuso orario

New interface

Specificazione del fuso orario di un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri. Nell'angolo in alto a destra dello schermo, è possibile selezionare uno degli intervalli di tempo predefiniti,

che partono da 1 ora fino a 1 settimana (1h, 3h, 12h, 1d, 3d, oppure 1w). In alternativa, è possibile scegliere Personalizza per impostare il tuo intervallo di tempo.

3. Scegli Personalizza quindi scegli il menu a discesa nell'angolo in alto a destra della casella. È possibile modificare il fuso orario in UTC o Fuso orario locale.
4. Dopo aver scelto i filtri, scegli Applica.

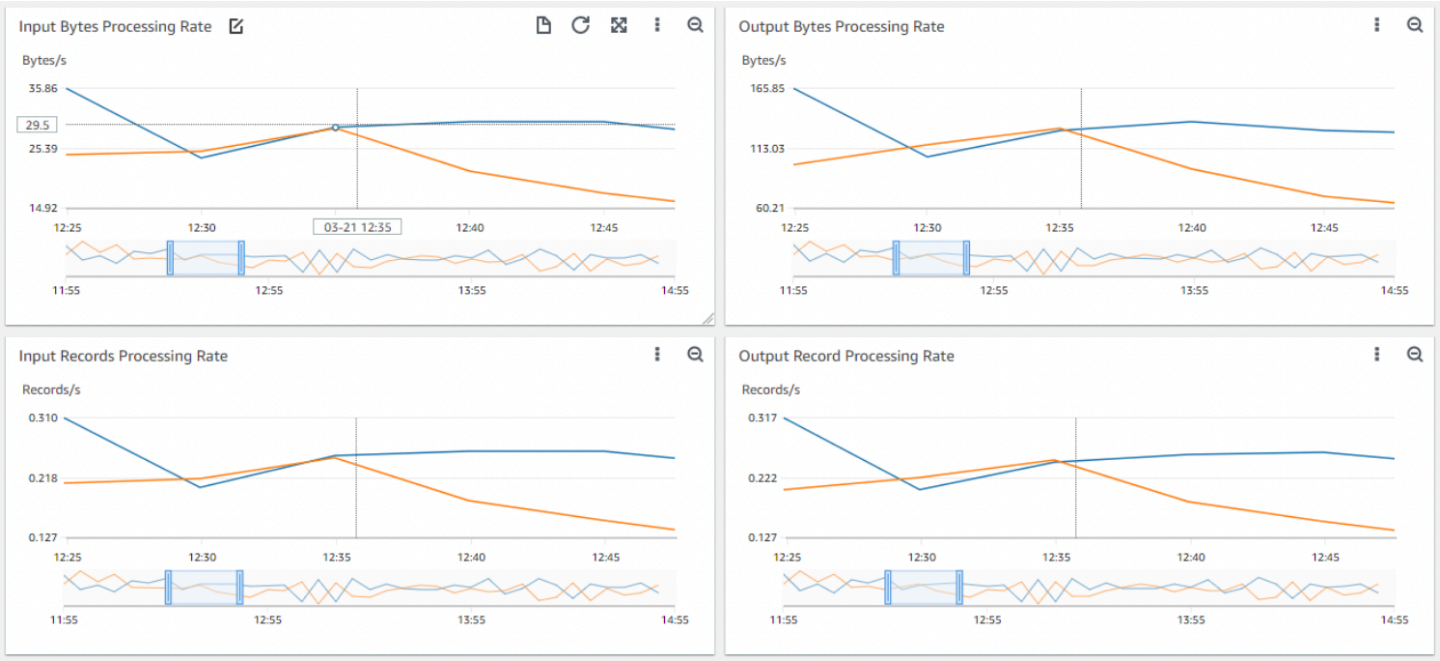
Original interface

Specificazione del fuso orario di un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri. Nell'angolo in alto a destra dello schermo, è possibile selezionare uno degli intervalli di tempo predefiniti, che partono da 1 ora fino a 1 settimana (1h, 3h, 12h, 1d, 3d, oppure 1w). In alternativa, è possibile scegliere personalizza per impostare il tuo intervallo di tempo.
3. Scegli personalizza quindi scegli il menu a discesa nell'angolo in alto a destra della casella. È possibile modificare il fuso orario in UTC o Fuso orario locale.

Ingrandimento di un grafico a linee o di un grafico ad aree in pila

Nella CloudWatch console, puoi utilizzare la funzione di zoom minimappa per concentrarti su sezioni di grafici a linee e grafici ad area impilata senza passare dalla visualizzazione ingrandita a quella ingrandita. Ad esempio, puoi utilizzare la funzione di zoom mini-mappa per concentrarti su un picco di un grafico lineare, in modo da poter confrontare il picco con altri parametri del pannello di controllo per lo stesso intervallo temporale. Le procedure in questa sezione descrivono come utilizzare la funzione di zoom.



Nell'immagine precedente, la funzione di zoom si concentra su un picco di un grafico lineare correlato alla velocità di elaborazione dei byte di input e mostra anche altri grafici lineari nel pannello di controllo che si concentrano su altre sezioni dello stesso intervallo temporale.

New interface

Ingrandimento di un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri.
3. Scegli Browse (Sfoglia), quindi seleziona uno o più parametri da rappresentare nel grafico.
4. Scegli Options (Opzioni), quindi seleziona Line (Linea) per Widget type (Tipo di widget).
5. Seleziona e trascina l'area del grafico che desideri ingrandire, quindi rilasciala.
6. Per ripristinare lo zoom, seleziona l'icona Reset zoom, a forma di lente d'ingrandimento con un simbolo meno (-) all'interno.

Original interface

Ingrandimento di un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione, scegli Metrics (Parametri), quindi scegli All metrics (Tutti i parametri).
3. Scegli All metrics (Tutti i parametri), quindi seleziona un parametro da rappresentare nel grafico.
4. Scegli Graph options (Opzioni del grafico). In Widget type (Tipo di widget), seleziona Line (Linea).
5. Seleziona e trascina l'area del grafico che desideri ingrandire, quindi rilasciala.
6. Per ripristinare lo zoom, seleziona l'icona Reset zoom, a forma di lente d'ingrandimento con un simbolo meno (-) all'interno.

> **Tip**
>
> Se hai già creato un pannello di controllo contenente un grafico lineare o ad aree in pila, puoi visitare il pannello di controllo e iniziare a utilizzare la funzione di zoom.

Modifica dell'asse Y di un grafico

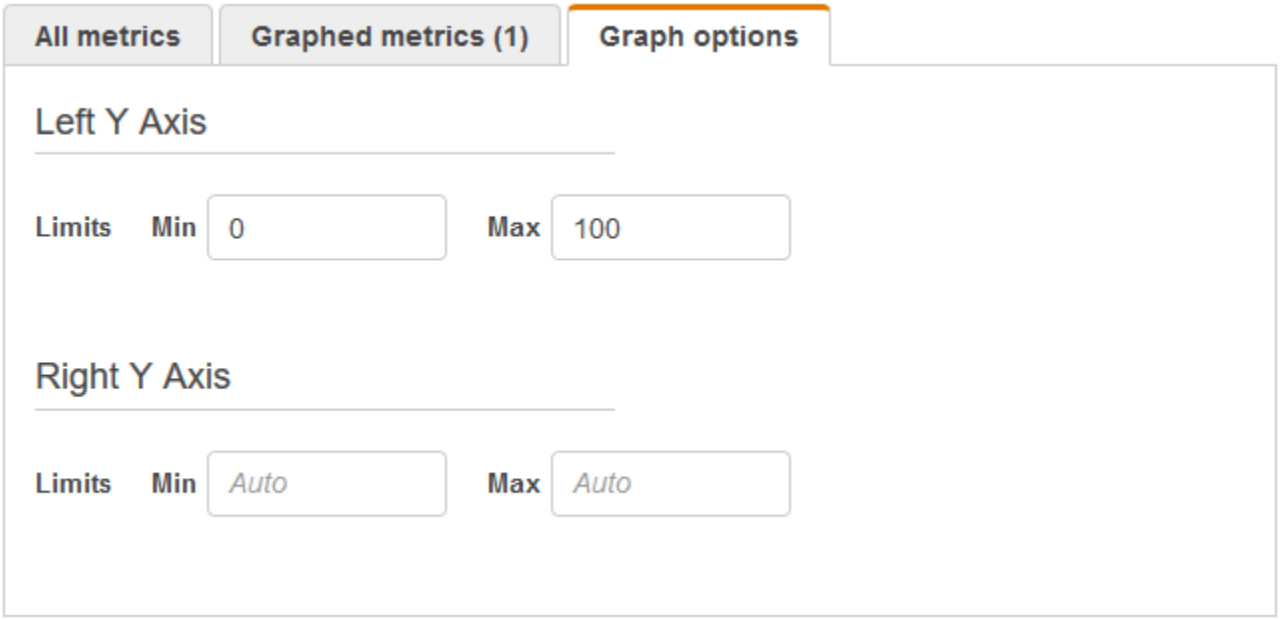
Puoi impostare limiti personalizzati per l'asse y su un grafico per visualizzare meglio i dati. Ad esempio, puoi modificare i limiti su un grafico `CPUUtilization` al 100%, in modo che risulti più facile vedere se la CPU è bassa (la riga tracciata si trova vicino alla parte inferiore del grafico) o elevata (la riga tracciata si trova nella parte superiore del grafico).

Puoi alternare due diversi assi y del grafico. Questa funzione è utile se il grafico contiene parametri aventi unità differenti o che differiscono notevolmente nei relativi intervalli di valori.

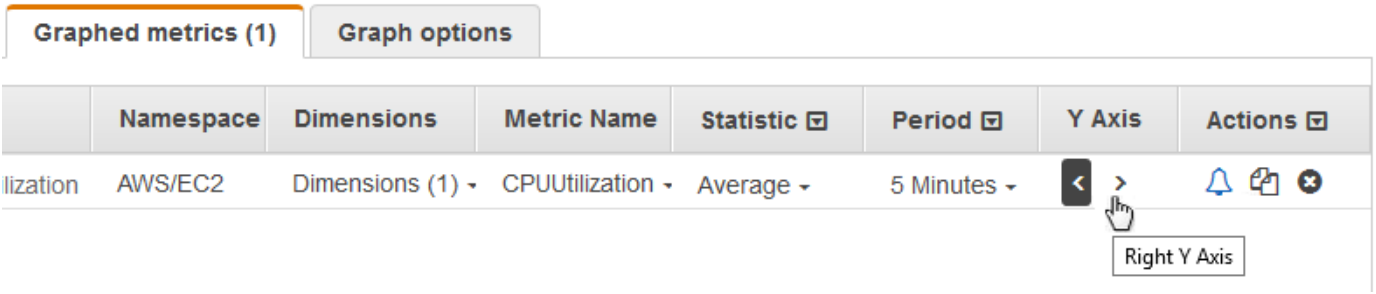
Per modificare l'asse y di un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Seleziona uno spazio dei nomi parametro (ad esempio, EC2) e una dimensione del parametro (ad esempio, Per-Instance Metrics (Parametri per istanza)).
4. La scheda All metrics (Tutti i parametri) visualizza tutti i parametri per tale dimensione in tale spazio dei nomi. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro.

5. Nella scheda Graph options (Opzioni del grafico), specifica i valori Min e Max per Left Y Axis (Asse Y sinistro). Il valore di Min non può essere maggiore di Max.



6. Per creare un secondo asse y, specificare i valori Min e Max di Right Y Axis (Asse Y destro).
7. Per alternare i due assi y, scegli la scheda Graphed metrics (Parametri nel grafico). In Y Axis (Asse Y), seleziona Left Y Axis (Asse Y sinistro) o Right Y Axis (Asse Y destro).



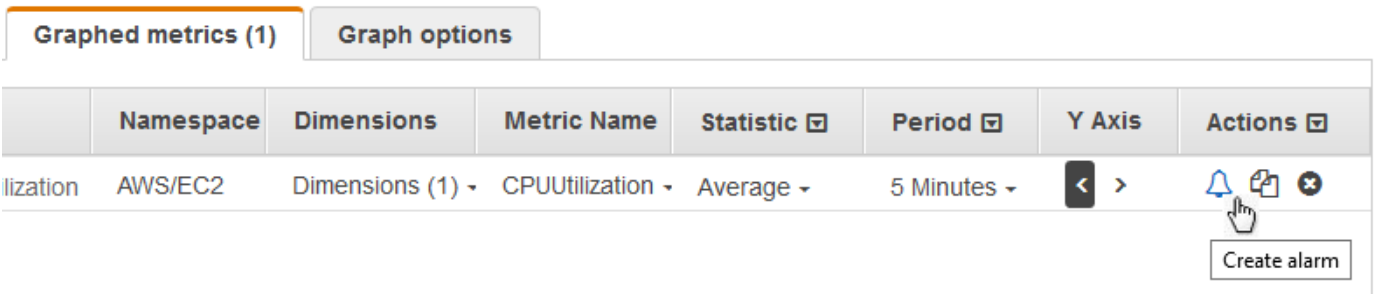
Creazione di un allarme a partire da un parametro in un grafico

Puoi creare il grafico di un parametro e un allarme a partire dal parametro nel grafico, offrendoti il vantaggio di popolare molti dei campi di allarme al tuo posto.

Creazione di un allarme a partire da un parametro in un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.

3. Seleziona uno spazio dei nomi parametro (ad esempio, EC2) e una dimensione del parametro (ad esempio, Per-Instance Metrics (Parametri per istanza)).
4. La scheda All metrics (Tutti i parametri) visualizza tutti i parametri per tale dimensione in tale spazio dei nomi. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro.
5. Per creare un allarme per il parametro, seleziona la scheda Graphed metrics (Parametri nel grafico). In Actions (Operazioni), seleziona l'icona dell'allarme.



6. In Conditions (Condizioni), scegli Static (Statico) o Anomaly detection (Rilevamento delle anomalie) per specificare se utilizzare una soglia statica o un modello di rilevamento delle anomalie per l'allarme.

   A seconda della scelta, inserire il resto dei dati per le condizioni di allarme.
7. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

   Per creare un allarme M di N, specifica un numero minore per il primo valore rispetto a quello specificato per il secondo valore. Per ulteriori informazioni, consulta la pagina Valutazione di un allarme.
8. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .
9. Seleziona Next (Successivo).
10. In Notification (Notifica), seleziona un argomento SNS per segnalare quando l'allarme è nello stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

Per fare in modo che l'allarme non invii notifiche, scegli Remove (Rimuovi).

11. Per fare in modo che l'allarme esegua operazioni Auto Scaling o EC2, scegli il pulsante appropriato e scegli lo stato di allarme e l'operazione da eseguire.
12. Al termine, scegli Apply (Applica).
13. Inserisci un nome e una descrizione per l'allarme. Il nome deve contenere solo caratteri ASCII. Quindi scegli Successivo.
14. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Utilizzo del CloudWatch rilevamento delle anomalie

Quando abiliti il rilevamento delle anomalie per una metrica, CloudWatch applica algoritmi statistici e di apprendimento automatico. Questi algoritmi analizzano continuamente i parametri di sistemi e applicazioni, determinano le normali linee di base e le anomalie superficiali con un intervento minimo dell'utente.

Gli algoritmi generano un modello di rilevamento delle anomalie. Il modello genera un intervallo di valori previsti che rappresentano il normale comportamento del parametro.

Puoi abilitare il rilevamento delle anomalie utilizzando l' AWS Management Console, l'o l' AWS CLI SDK AWS CloudFormation. AWS Puoi abilitare il rilevamento delle anomalie sulle metriche fornite da AWS e anche sulle metriche personalizzate. In un account configurato come account di monitoraggio per l'osservabilità tra più CloudWatch account, puoi creare rilevatori di anomalie sulle metriche degli account di origine oltre alle metriche dell'account di monitoraggio.

Puoi utilizzare il modello di valori previsti in due modi:

- Crea allarmi basati sul rilevamento delle anomalie, ovvero allarmi basati sul valore previsto di un parametro. Questi tipi di allarmi non hanno una soglia statica per determinare lo stato dell'allarme. Confrontano invece il valore del parametro con il valore previsto, in base al modello di rilevamento delle anomalie.

  Puoi decidere se l'allarme viene attivato quando il valore del parametro è al di sopra dell'intervallo di valori previsti, si trova al di sotto di tale intervallo oppure in entrambi i casi.

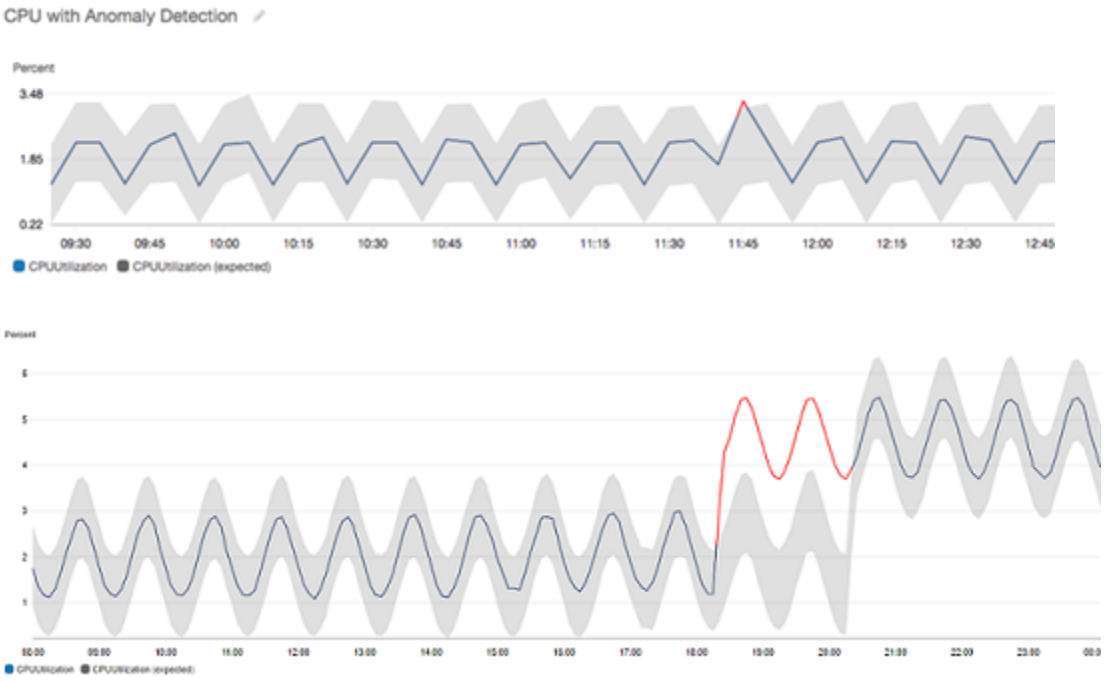
  Per ulteriori informazioni, consulta Crea un allarme basato sul rilevamento delle CloudWatch anomalie.

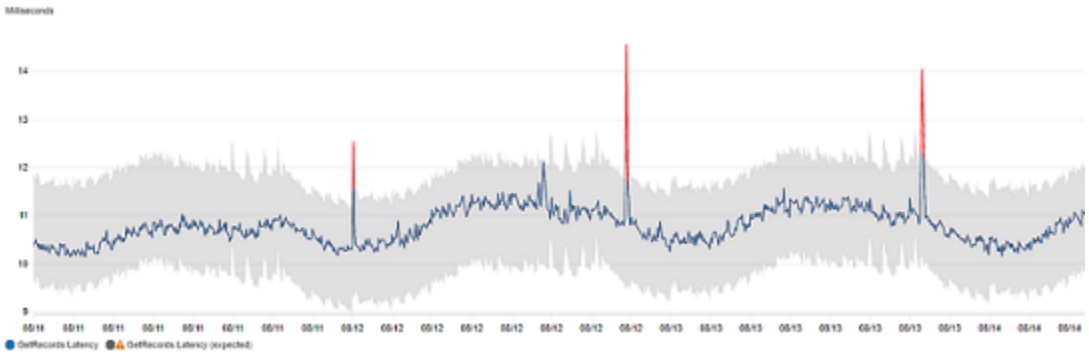
- Quando si visualizza un grafico di dati di parametri, i valori previsti vengono sovrapposti nel grafico sotto forma di intervallo. Ciò rende visivamente chiaro quali valori nel grafico non sono compresi nell'intervallo normale. Per ulteriori informazioni, consulta Creazione di un grafico.

  Puoi anche recuperare i valori superiore e inferiore dell'intervallo del modello mediante la richiesta API `GetMetricData` con la funzione matematica `ANOMALY_DETECTION_BAND` del parametro. Per ulteriori informazioni, consulta. GetMetricData

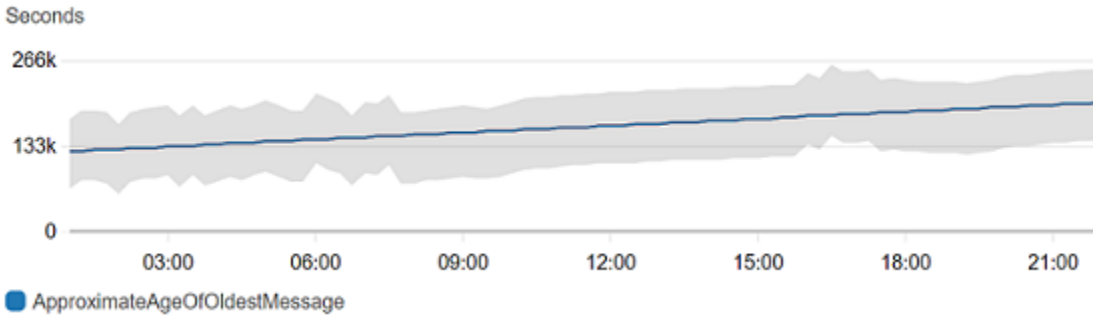
In un grafico con rilevamento di anomalie, l'intervallo di valori previsto viene mostrato come un intervallo grigio. Se il valore effettivo del parametro va oltre questo intervallo, viene visualizzato in rosso per tale periodo.

Gli algoritmi di rilevamento delle anomalie tengono conto delle variazioni di stagionalità e di tendenza dei parametri. Le variazioni di stagionalità potrebbero essere orarie, giornaliere o settimanali, come mostrato negli esempi seguenti.

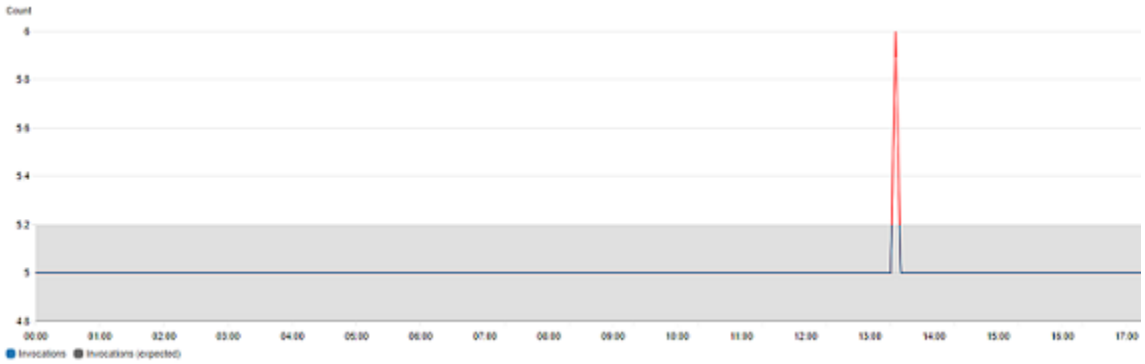




Le tendenze con intervallo più lungo possono essere al ribasso o al rialzo.



I rilevamenti delle anomalie funzionano bene anche con i parametri con pattern normali.



Come funziona il rilevamento delle CloudWatch anomalie

Quando abiliti il rilevamento delle anomalie per una metrica, CloudWatch applica algoritmi di apprendimento automatico ai dati passati della metrica per creare un modello dei valori previsti della metrica. Il modello valuta sia le tendenze che i pattern orari, giornalieri e settimanali del parametro. L'algoritmo esegue l'apprendimento in base a due settimane di dati del parametro, ma puoi abilitare il rilevamento di anomalie per un parametro anche se il parametro non dispone di tale intervallo di dati.

Specificate un valore per la soglia di rilevamento delle anomalie che CloudWatch viene utilizzato insieme al modello per determinare l'intervallo di valori «normale» per la metrica. Un valore più alto per la soglia del rilevamento delle anomalie produce un intervallo più ampio di valori "normali".

Il modello di machine learning è specifico di un parametro o di una funzione statistica. Ad esempio, se abiliti il rilevamento di anomalie per un parametro mediante la funzione statistica `AVG`, il modello fa riferimento specifico alla funzione statistica `AVG`.

Quando CloudWatch crea un modello per molte metriche comuni a partire dai AWS servizi, assicura che la banda non si estenda al di fuori dei valori logici. Ad esempio, la banda `MemoryUtilization` di un'istanza EC2 rimarrà compresa tra 0 e 100 e il tracciamento delle bande, che non può essere negativo CloudFront `Requests`, non si estenderà mai al di sotto dello zero.

Dopo aver creato un modello, il rilevamento delle CloudWatch anomalie valuta continuamente il modello e lo aggiusta per garantire che sia il più preciso possibile. Ciò include riaddestrare il modello per regolare se i valori dei parametri si evolvono nel tempo o presentano cambiamenti improvvisi, e include anche i predittori per migliorare i modelli di parametri stagionali, variabili o sparse.

Dopo aver abilitato il rilevamento di anomalie per un parametro, per l'apprendimento automatico del modello puoi scegliere di escludere l'uso di periodi di tempo specifici relativi al parametro in questione. In questo modo, puoi escludere l'uso di distribuzioni o di altri eventi insoliti durante l'apprendimento automatico del modello in modo da garantire la creazione di un modello più preciso.

L'utilizzo di modelli di rilevamento delle anomalie per gli allarmi comporta addebiti sull'account. AWS Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

Rilevamento di anomalie sulla matematica del parametro

Il rilevamento di anomalie sulla matematica del parametro è una funzione che è possibile utilizzare per creare allarmi di rilevamento anomalie sull'output di espressioni matematiche dei parametri. È possibile utilizzare queste espressioni per creare grafici che visualizzano le bande di rilevamento delle anomalie. La funzione supporta funzioni aritmetiche di base, operatori logici e confronto e la maggior parte delle altre funzioni. Per informazioni sulle funzioni che non sono supportate, consulta Using metric Math nella Amazon CloudWatch User Guide.

È possibile creare modelli di rilevamento anomalie basati su espressioni matematiche dei parametri simili a come si creano già modelli di rilevamento delle anomalie. Dalla CloudWatch console, puoi applicare il rilevamento delle anomalie alle espressioni matematiche metriche e selezionare il rilevamento delle anomalie come tipo di soglia per queste espressioni.

> **Note**
>
> Il rilevamento delle anomalie solo sulla matematica dei parametri può essere abilitato e modificato nell'ultima versione dell'interfaccia utente dei parametri. Quando si creano rilevatori di anomalie basati su espressioni matematiche dei parametri nella nuova versione dell'interfaccia, è possibile visualizzarli nella versione precedente, ma non modificarli.

Per informazioni su come creare allarmi e modelli per il rilevamento delle anomalie e la matematica dei parametri, vedere le seguenti sezioni:

- Creazione di un CloudWatch allarme basato sul rilevamento delle anomalie
- Creazione di un CloudWatch allarme basato su un'espressione matematica metrica

Puoi anche creare, eliminare e scoprire modelli di rilevamento delle anomalie basati su espressioni matematiche metriche utilizzando l' CloudWatch API con, e. `PutAnomalyDetector` `DeleteAnomalyDetector` `DescribeAnomalyDetectors` Per informazioni su queste azioni API, consulta le seguenti sezioni in Amazon CloudWatch API Reference.

- PutAnomalyDetector
- DeleteAnomalyDetector
- DescribeAnomalyDetectors

Per informazioni sui prezzi degli allarmi di rilevamento delle anomalie, consulta i prezzi di Amazon. CloudWatch

Utilizzare la matematica dei parametri

La matematica metrica consente di interrogare più CloudWatch metriche e utilizzare espressioni matematiche per creare nuove serie temporali basate su queste metriche. Puoi visualizzare le serie temporali risultanti sulla CloudWatch console e aggiungerle ai dashboard. Utilizzando le AWS Lambda metriche come esempio, puoi dividere la `Errors` metrica per la metrica per ottenere un `Invocations` tasso di errore. Quindi aggiungi le serie temporali risultanti a un grafico sulla dashboard. CloudWatch

È inoltre possibile eseguire calcoli matematici dei parametri a livello di codice, utilizzando l'operazione API `GetMetricData`. Per ulteriori informazioni, consulta GetMetricData.

Aggiungere un'espressione matematica a un grafico CloudWatch

Puoi aggiungere un'espressione matematica a un grafico sulla dashboard CloudWatch . Ogni grafico è limitato all'utilizzo di un massimo di 500 parametri ed espressioni, perciò è possibile aggiungere un'espressione matematica solo se il grafico ha al massimo 499 parametri. Ciò vale anche se non tutti i parametri vengono visualizzati sul grafico.

Per aggiungere un'espressione matematica a un grafico

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Creare o modificare un grafico. Ci deve essere almeno una metrica nel grafico.
3. Seleziona Graphed metrics (Parametri nel grafico).
4. Scegli Math expression (Espressione matematica), Start with an empty expression (Inizia con un'espressione vuota). Viene visualizzata una nuova riga per l'espressione.
5. Nella nuova riga, nella colonna Dettagli immetti l'espressione matematica. Le tabelle della sezione Sintassi e funzioni della formula di parametri elencano le funzioni che è possibile utilizzare nell'espressione.

   Per usare un parametro o il risultato di un'altra espressione come parte della formula di questa espressione, utilizzare il valore illustrato nella colonna id: ad esempio m1+m2 o e1-MIN(e1).

   Puoi modificare il valore di Id. È possibile includere numeri, lettere un carattere di sottolineatura e devono iniziare con una lettera minuscola. Modificando il valore di Id con un nome più significativo, inoltre, è possibile ottenete un grafico più comprensibile: ad esempio, si può modificare m1 e m2 con errors e requests.

   > **Tip**
   >
   > Scegli la freccia verso il basso accanto a Math Expression (Espressione matematica) per visualizzare un elenco di funzioni supportate, che è possibile utilizzare durante la creazione dell'espressione.

6. Nella colonna Label (Etichetta) dell'espressione, immetti un nome che descriva ciò che l'espressione sta calcolando.

   Se il risultato di un'espressione è una gamma di serie temporali, ognuna di queste serie temporali viene visualizzata nel grafico con una linea separata, con colori diversi. Immediatamente sotto il grafico c'è una legenda per ogni riga nel grafico. Per una singola

espressione che produce più serie temporali, le didascalie della legenda per quelle serie temporali sono nel formato ***`Expression-Label Metric-Label`***. Ad esempio, se il grafico include un parametro con un'etichetta di Errors(Errori) e un'espressione FILL (METRICS (), 0), che ha un'etichetta di Filled With 0: (Completo con 0:), una linea nella legenda sarebbe Filled With 0: Errors (Completo con 0: Errori). Impostare *`Expression-Label`* in modo che sia vuota e fare in modo che la legenda mostri solo le etichette originali del parametro.

Quando un'espressione produce un array di serie temporali nel grafico, non è possibile modificare i colori utilizzati per ciascuna di tali serie temporali.

7. Dopo aver aggiunto le espressioni desiderate, è possibile semplificare il grafico nascondendo alcuni dei parametri originali. Per nascondere un parametro o un'espressione, deseleziona la casella di controllo a sinistra del campo Id.

Sintassi e funzioni della matematica dei parametri

Le seguenti sezioni spiegano le funzioni disponibili per la matematica dei parametri. Tutte le funzioni devono essere scritte in lettere maiuscole (ad esempio AVG), mentre il campo Id di tutti i parametri e le espressioni matematiche deve iniziare con una lettera minuscola.

Il risultato finale di qualsiasi espressione matematica deve essere una singola serie temporale o un array di serie temporali. Alcune funzioni producono un numero scalare. È possibile utilizzare queste funzioni all'interno di una funzione più grande che produce alla fine una serie temporale. Ad esempio, l'AVG di una singola serie temporale produce un numero scalare, per cui non può essere il risultato finale dell'espressione. Tuttavia, è possibile utilizzarlo nella funzione m1-AVG(m1) per visualizzare una serie temporale della differenza tra ogni singolo punto di dati e il valore medio nella serie temporale.

Abbreviazioni dei tipi di dati

Alcune funzioni sono valide solo per determinati tipi di dati. Le abbreviazioni dell'elenco seguente vengono utilizzate in tabelle di funzioni per rappresentare i tipi di dati supportati per ciascuna funzione:

- S rappresenta un numero scalare, ad esempio 2, -5 o 50.25.
- TS è una serie temporale (una serie di valori per una singola CloudWatch metrica nel tempo): ad esempio, la `CPUUtilization` metrica relativa agli `i-1234567890abcdef0` ultimi tre giorni.
- TS [] è una gamma di serie temporali, ad esempio le serie temporali per più parametri.

- String[] è un array di stringhe.

La funzione METRICS()

La funzione METRICS() restituisce tutti i parametri della richiesta. Le espressioni matematiche non sono incluse.

È possibile utilizzare METRICS() all'interno di un'espressione maggiore che produce una singola serie temporale o un array di serie temporali. Ad esempio, l'espressione SUM(METRICS()) restituisce una serie temporale (TS) che è la somma dei valori di tutti i parametri nel grafico. METRICS()/100 restituisce un array di serie temporali, ciascuna delle quali è una serie temporale che mostra ciascun punto dati di uno dei parametri diviso per 100.

È possibile utilizzare la funzione METRICS() con una stringa per restituire solo i parametri contenuti nel grafico che contengono tale stringa nel campo Id. Ad esempio, l'espressione SUM(METRICS("errors")) restituisce una serie temporale che è la somma dei valori di tutti i parametri nel grafico che contengono "errori" nel campo Id. È inoltre possibile utilizzare SUM([METRICS("4xx"), METRICS("5xx")]) per abbinare più stringhe.

Funzioni aritmetiche di base

La tabella seguente elenca le funzioni aritmetiche di base supportate. I valori mancanti in una serie temporale vengono trattati come 0. Se il valore di un punto dati porta una funzione a tentare di dividere per zero, il punto dati viene eliminato.

Operazione	Argomenti	Esempi
Operatori aritmetici: + - * / ^	S, S	PERIOD(m1)/60
	S, TS	5 * m1
	TS, TS	m1 - m2
	S, TS[]	SUM(100/[m1, m2])
	TS, TS[]	AVG(METRICS())
		METRICS()*100
Unary subtraction -	S	-5*m1

Operazione	Argomenti	Esempi
	TS TS[]	-m1 SUM(-[m1, m2])

Confronto e operatori logici

È possibile utilizzare operatori di confronto e logici con una coppia di serie temporali o una coppia di singoli valori scalari. Quando si utilizza un operatore di confronto con una coppia di serie temporali, gli operatori restituiscono una serie temporale in cui ogni punto dati è 0 (false) o 1 (true). Se si utilizza un operatore di confronto su una coppia di valori scalari, viene restituito un singolo valore scalare, 0 o 1.

Quando gli operatori di confronto vengono utilizzati tra due serie temporali e solo una delle serie temporali ha un valore per un determinato timestamp, la funzione considera il valore mancante nell'altra serie temporale come 0.

È possibile utilizzare gli operatori logici in combinazione con gli operatori di confronto, per creare funzioni più complesse.

Nella tabella seguente sono elencati gli operatori supportati.

Tipo di operatore	Operatori supportati
Operatori di confronto	== != <= >= < >
Operatori logici	AND o && OR o ||

Per vedere come vengono utilizzati questi operatori, supponiamo di avere due serie temporali: metric1 ha valori di `[30, 20, 0, 0]` e metric2 ha valori di `[20, -, 20, -]` dove - indica che non c'è alcun valore per quella data e ora.

Expression	Output
(metric1 < metric2)	0, 0, 1, 0
(metric1 >;= 30)	1, 0, 0, 0
(metric1 > 15 AND metric2 > 15)	1, 0, 0, 0

Funzioni supportate per la matematica dei parametri

La tabella riportata di seguito descrive le funzioni che è possibile utilizzare in espressioni matematiche. Immetti tutte le funzioni in lettere maiuscole.

Il risultato finale di qualsiasi espressione matematica deve essere una singola serie temporale o un array di serie temporali. Alcune funzioni nelle tabelle nelle seguenti sezioni producono un numero scalare. È possibile utilizzare queste funzioni all'interno di una funzione più grande che produce alla fine una serie temporale. Ad esempio, l'AVG di una singola serie temporale produce un numero scalare, per cui non può essere il risultato finale dell'espressione. Tuttavia, è possibile utilizzarlo nella funzione m1-AVG(m1) per visualizzare una serie temporale della differenza tra ogni singolo punto di dati e il valore medio di quel punto di dati.

Nella tabella riportata di seguito, ogni esempio nella colonna Examples (Esempi) è un'espressione che consente di ottenere una singola serie temporale o un array di serie temporali. Questo esempio mostra come le funzioni che restituiscono numeri scalari possono essere utilizzate come parte di un'espressione valida che produce una singola serie temporale.

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
ABS	TS TS[]	TS TS[]	Restituisce il valore assoluto di ogni punto dati.	ABS(m1-m2) MIN(ABS([m1, m2])) ABS(METRICS())	✓
ANOMALY_DETECTION_BAND	TS TS, S	TS[]	Restituisce un intervallo di rilevamento delle anomalie per il parametro specificato. L'intervallo è composto da due serie temporali, una che rappresenta il limite superiore del valore previsto "normale" del parametro e l'altra che rappresenta il limite inferiore. La funzione può richiedere due argomenti. Il primo è l'ID del parametro per il quale creare l'intervallo. Il secondo argomento è il numero di deviazioni standard da usare per l'intervallo. Se non si specifica questo argomento, viene utilizzato il valore predefinito 2. Per ulteriori informazi	ANOMALY_DETECTION_BAND(m1) ANOMALY_DETECTION_BAND(m1,4)	

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			oni, consulta Utilizzo del CloudWatch rilevamento delle anomalie.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
AVG	TS TS[]	S TS	L'AVG di una singola serie temporale restituisce un valore scalare con la media di tutti i punti di dati del parametro . L'AVG di un array di serie temporali restituisce una singola serie temporale. I valori mancanti vengono trattati come 0. **Note** Si consiglia di non utilizzare questa funzione negli CloudWatch allarmi se si desidera che la funzione restituisca uno scalare. Ad esempio, `AVG(m2)`. Ogni volta che un allarme valuta se cambiare stato, CloudWatch tenta di	SUM([m1,m2])/AVG(m2) AVG(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			recuperare un numero maggiore di punti dati rispetto al numero specificato come Periodi di valutazione. Questa funzione agisce in modo diverso quando vengono richiesti dati aggiuntivi. Per utilizzare questa funzione con gli allarmi, in particolare gli allarmi con operazioni di dimensionamento automatico, si consiglia di impostare l'allarme in modo che utilizzi M punti dati su N, dove M < N.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
CEIL	TS TS[]	TS TS[]	Restituisce il valore limite ogni parametro. Il valore limite è il numero intero più piccolo maggiore o uguale a ciascun valore.	CEIL(m1) CEIL(METRICS()) SUM(CEIL(METRICS()))	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
DATAPOINT_COUNT	TS TS[]	S TS	Restituisce un conteggio dei punti dati che hanno segnalato valori. È utile per calcolare le medie di parametri sparse. **Note** Si consiglia di non utilizzare questa funzione negli CloudWatch allarmi. Ogni volta che un allarme valuta se cambiare stato, CloudWatch tenta di recuperare un numero maggiore di punti dati rispetto al numero specificato come Periodi di valutazione. Questa funzione agisce in modo diverso	SUM(m1) / DATAPOINT_COUNT(m1) DATAPOINT_COUNT(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			quando vengono richiesti dati aggiuntivi.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
DB_PERF_INSIGHTS	Stringa, stringa, stringa String, String, String[]	TS (se viene fornita una singola stringa) TS[] (se viene fornito un array di stringhe	Restituisce i parametri del contatore di Approfondimenti sulle prestazioni per database come Amazon Relational Database Service e Amazon DocumentDB (compatibile con MongoDB). Questa funzione restituisce la stessa quantità di dati che è possibile ottenere interrogando direttamente le API di Approfondimenti sulle prestazioni. È possibile utilizzare queste metriche CloudWatch per rappresentare graficamente e creare allarmi. **Important** Quando usi questa funzione, devi specificare l'ID univoco della risorsa del database. È diverso dall'iden	DB_PERF_INSIGHTS('RDS', 'db-ABCDEFGHIJKLMNOPQRSTUVWXY1', 'os.cpuUtilization.user.avg') DB_PERF_INSIGHTS('DOCDB, 'db-ABCDEFGHIJKLMNOPQRSTUVWXY1', ['os.cpuUtilization.idle.avg', 'os.cpuUtilization.user.max'])	

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			tificatore del database. Per trovare l'ID della risorsa del database nella console Amazon RDS, scegli l'istanza database per vederne i dettagli. Quindi seleziona la scheda Configurazione. ID risorsa è visualizzato nella sezione Configurazione. DB_PERF_INSIGHTS introduce il parametro DBLoad anche a intervalli inferiori al minuto. Le metriche di Performance Insights recuperate con questa funzione non vengono archiviate in. CloudWatc		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			h Pertanto, alcune CloudWatch funzionalità come l'osservabilità tra account, il rilevamento delle anomalie, i flussi di metriche, il metrics explorer e Metric Insights non funzionano con le metriche di Performance Insights recuperate con DB_PERF_INSIGHTS. Una singola richiesta che utilizza la funzione DB_PERF_INSIGHTS può recuperare il seguente numero di punti dati. • 1080 punti dati per periodi ad alta risoluzione (1 s, 10 s, 30 s) • 1440 punti dati per periodi di risoluzione standard (1 m, 5 m, 1 ora, 1d) La funzione DB_PERF_INSIGHTS supporta solo		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			le seguenti lunghezze di periodo: • 1 secondo • 10 secondi • 30 secondi • 1 minuto • 5 minuti • 1 ora • 1 giorno Per ulteriori informazioni su Approfondimenti sulle prestazioni di Amazon RDS, consulta Approfondimenti sulle prestazioni per i parametri del contatore. Per ulteriori informazioni su Approfondimenti sulle prestazioni di Amazon DocumentDB, consulta Approfondimenti sulle prestazioni per i parametri del contatore.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			Note I parametri ad alta risoluzione con granularità inferiore al minuto recuperati da DB_PERF_INSIGHTS sono applicabili solo al parametro DBLoad o ai parametri del sistema operativo se è stato abilitato il monitoraggio avanzato a una risoluzione più elevata. Per ulteriori informazioni sul monitoraggio avanzato di Amazon RDS, consulta Monitoraggio dei parametri del sistema operativo con		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			il monitoraggio avanzato. È possibile creare un allarme ad alta risoluzione utilizzando la funzione DB_PERF_INSIGHTS per un intervallo di tempo massimo di tre ore. È possibile utilizzare la CloudWatch console per rappresentare graficamente le metriche recuperate con la funzione DB_PERF_INSIGHTS per qualsiasi intervallo di tempo.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
DIFF	TS TS[]	TS TS[]	Restituisce la differenza tra ogni valore della serie temporale e il valore precedente di quella serie temporale.	DIFF(m1)	✓
DIFF_TIME	TS TS[]	TS TS[]	Restituisce la differenza in secondi tra il timestamp di ogni valore della serie temporale e il timestamp del valore precedente di quella serie temporale.	DIFF_TIME(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
FILL	TS, [S | REPEA | LINEAR TS[], [TS | S | REPEA | LINEAR	TS TS[]	Riempie i valori mancanti di una serie temporale. Ci sono diverse opzioni per i valori da utilizzare come filler per i valori mancanti: • È possibile specificare un valore da utilizzare come valore filler. • È possibile specifica re un parametro da utilizzare come valore filler. • Puoi utilizzare la parola chiave REPEAT per riempire i valori mancanti con il valore effettivo più recente del parametro prima del valore mancante. • Puoi utilizzare la parola chiave LINEAR per riempire i valori mancanti con valori che creano un'interp olazione lineare tra i	FILL(m1,10) FILL(METRICS(), 0) FILL(METRICS(), m1) FILL(m1, MIN(m1)) FILL(m1, REPEAT) FILL(METRICS(), LINEAR)	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supporto per multi-account?
			valori all'inizio e alla fine dello spazio. **Note** Quando usi questa funzione in un avviso, puoi riscontrare un problema se i tuoi parametri vengono pubblicati con un leggero ritardo e il minuto più recente non ha mai dati. In questo caso, FILL sostituisce il punto dati mancante con il valore richiesto. Ciò fa sì che il punto dati più recente per il parametro sia sempre il valore FILL, che può bloccare		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			l'allarme nello stato OK o ALARM. Puoi aggirare questo problema usando un allarme M di N. Per ulteriori informazioni, consulta Valutazione di un allarme.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
FIRST LAST	TS[]	TS	Restituisce la prima o l'ultima serie temporale da un array di serie temporali. Questo è utile quando viene utilizzato con la funzione SORT. Può anche essere utilizzato per ottenere le soglie minime e massime dalla funzione ANOMALY_DETECTION_BAND.	IF(FIRST(SORT(METRICS(), AVG, DESC))>100, 1, 0) esamina il parametro principale da un array, che viene ordinato per AVG. Restituisce quindi un 1 o uno 0 per ogni punto di dati, a seconda se il valore del punto di dati è superiore a 100. LAST(ANOMALY_DETECTION_BAND(m1)) restituisce il limite superiore della banda di previsione dell'anomalia.	✓
FLOOR	TS TS[]	TS TS[]	Restituisce il valore minimo di ciascun parametro. Il valore minimo è il numero intero più grande minore o uguale a ciascun valore.	FLOOR(m1) FLOOR(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
IF	Espressione IF	TS	Utilizzare IF insieme a un operatore di confronto per filtrare i punti dati da una serie temporale o creare una serie temporale mista composta da più serie temporali raccolte. Per ulteriori informazioni, consulta Utilizzo delle espressioni IF.	Per alcuni esempi, consulta Utilizzo delle espressioni IF.	✓
INSIGHT_RULE_METRIC	INSIGHTULE_MIC(ruleNe, metricNe)	TS	Utilizza INSIGHT_RULE_METRIC per estrarre le statistiche da una regola in Contributor Insights. Per ulteriori informazioni, consulta Rappresentazione grafica dei parametri generati dalle regole.		
LAMBDA	LAMBD([, opziona arg] *) LambdactionNa	TS TS{}	Richiama una funzione Lambda per interrogare le metriche da un'origine dati che non lo è. CloudWatch Per ulteriori informazioni, consulta Come passare argomenti alla funzione Lambda.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
LOG	TS TS[]	TS TS[]	Il LOG di una serie temporale restituisce il valore logaritmico naturale di ogni valore della serie temporale.	LOG(METRICS())	✓
LOG10	TS TS[]	TS TS[]	Il LOG10 di una serie temporale restituisce il valore logaritmico in base 10 di ogni valore della serie temporale.	LOG10(m1)	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
MAX	TS TS[]	S TS	Il MAX di una singola serie temporale restituisce un valore scalare con il valore massimo di tutti i punti di dati del parametro. Se si immette una matrice di serie temporali, la funzione MAX crea e restituisce una serie temporale costituita dal valore più alto per ogni punto dati, tra le serie temporali utilizzate come input. **Note** Si consiglia di non utilizzare questa funzione negli CloudWatch allarmi se si desidera che la funzione restituisca uno scalare. Ad esempio, `MAX(m2)` ogni volta che	MAX(m1)/m1 MAX(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			un allarme valuta se cambiare stato, CloudWatch tenta di recuperare un numero maggiore di punti dati rispetto al numero specificato come Periodi di valutazione. Questa funzione agisce in modo diverso quando vengono richiesti dati aggiuntivi.		
METRIC_COUNT	TS[]	S	Restituisce il numero di parametri nell'array di serie temporali.	m1/METRIC_COUNT(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
PARAMETRI	null string	TS[]	La funzione METRICS() restituisce tutte le CloudWatch metriche della richiesta. Le espressioni matematiche non sono incluse. È possibile utilizzare METRICS() all'interno di un'espressione maggiore che produce una singola serie temporale o un array di serie temporali. È possibile utilizzare la funzione METRICS() con una stringa per restituire solo i parametri contenuti nel grafico che contengono tale stringa nel campo Id. Ad esempio, l'espressione SUM(METRICS("errors")) restituisce una serie temporale che è la somma dei valori di tutti i parametri nel grafico che contengono "errori" nel campo Id. È inoltre possibile utilizzar	AVG(METRICS()) SUM(METRICS("errors"))	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			e SUM([METRICS("4xx"), METRICS("5xx")]) per abbinare più stringhe.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
MIN	TS TS[]	S TS	Il MIN di una singola serie temporale restituisce un valore scalare con il valore minimo di tutti i punti di dati del parametro. Se si immette una matrice di serie temporali, la funzione MIN crea e restituisce una serie temporale costituita dal valore più basso per ogni punto dati, tra le serie temporali utilizzate come input. Se si immette una matrice di serie temporali, la funzione MIN crea e restituisce una serie temporale costituita dal valore più alto per ogni punto dati, tra le serie temporali utilizzate come input.	m1-MIN(m1) MIN(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			Note Si consiglia di non utilizzare questa funzione negli CloudWatch allarmi se si desidera che la funzione restituisca uno scalare. Ad esempio, `MIN(m2)` ogni volta che un allarme valuta se cambiare stato, CloudWatch tenta di recuperare un numero maggiore di punti dati rispetto al numero specificato come Periodi di valutazione. Questa funzione agisce in modo diverso		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			quando vengono richiesti dati aggiuntivi.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?

MINUTO
ORA
GIORNO
DATA
MESE
ANNO
EPOCA
	TS	TS	Queste funzioni prendono il periodo e l'intervallo della serie temporale e restituiscono una nuova serie temporale non sparse in cui ogni valore è basato sul relativo timestamp.
			• MINUTE restituisce una serie temporale non sparse di numeri interi compresi tra 0 e 59 che rappresentano il minuto UTC di ogni timestamp nella serie temporale originale.
			• HOUR restituisce una serie temporale non sparse di numeri interi compresi tra 0 e 23 che rappresentano l'ora UTC di ogni timestamp nella serie temporale originale.
			• DAY restituisce una serie temporale non sparse di numeri interi tra 1 e 7 che rappresentano il
				MINUTE(m1)
				IF(DAY(m1)<6,m1) restituisce i parametri solo dai giorni feriali, dal lunedì al venerdì.
				IF(MONTH(m1) == 4,m1) restituisce solo i parametri pubblicati ad aprile.
					✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			giorno UTC della settimana di ogni timestamp nella serie temporale originale. 1 rappresenta il lunedì e 7 la domenica. • DATE restituisce una serie temporale non sparse di numeri interi tra 1 e 31 che rappresentano il giorno UTC del mese di ogni timestamp nella serie temporale originale. • MONTH restituisce una serie temporale non sparse di numeri interi tra 1 e 12 che rappresentano il mese UTC di ogni timestamp nella serie temporale originale. 1 rappresenta gennaio e 12 rappresenta dicembre. • YEAR restituisce una serie temporale non sparse di numeri interi che rappresentano		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			l'anno UTC di ogni timestamp nella serie temporale originale. • EPOCH restituisce una serie temporale non sparsa di interi che rappresentano il tempo UTC in secondi dall'epoca di ogni timestamp nella serie temporale originale. L'epoca è il 1° gennaio 1970.		
PERIODO	TS	S	Restituisce il periodo di un parametro in secondi. L'input valido è un parametro, non i risultati di altre espressioni.	m1/PERIOD(m1)	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
TARIFFA	TS TS[]	TS TS[]	Restituisce la percentuale di variazione del parametro al secondo. Questo viene calcolato come la differenza tra il valore dell'ultimo punto dati e il valore precedente, diviso per la differenza temporale in secondi tra i due valori. **Important** L'impostazione di allarmi su espressioni che utilizzano la funzione RATE su metriche con dati sparsi può comportarsi in modo imprevedibile, poiché l'intervallo di punti dati recuperato durante la valutazione dell'allarme può variare in base all'ultima	RATE(m1) RATE(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			pubblicazione dei punti dati.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
REMOVE_EMPTY	TS[]	TS[]	Rimuove tutte le serie temporali che non dispongono di punti di dati da una gamma di serie temporali. Il risultato è un gamma di serie temporali in cui ogni serie temporale contiene almeno un punto di dati. **Note** Ti consigliamo di non utilizzare questa funzione negli allarmi. CloudWatch Ogni volta che un allarme valuta se cambiare stato, CloudWatch tenta di recuperare un numero maggiore di punti dati rispetto al numero specificato	REMOVE_EMPTY (METRICS ())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			come Periodi di valutazione. Questa funzione agisce in modo diverso quando vengono richiesti dati aggiuntivi.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
RUNNING_SUM	TS TS[]	TS TS[]	Restituisce una serie temporale con la somma parziale dei valori della serie temporale originale . **Note** Si consiglia di non utilizzare questa funzione negli CloudWatch allarmi. Ogni volta che un allarme valuta se cambiare stato, CloudWatch tenta di recuperare un numero maggiore di punti dati rispetto al numero specificato come Periodi di valutazione. Questa funzione agisce in modo diverso	RUNNING_SUM([m1,m2])	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			quando vengono richiesti dati aggiuntivi.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
SEARCH	Espressione di ricerca	Uno o più TS	Restituisce una o più serie temporali che corrispondono a un criterio di ricerca specificato. La funzione SEARCH consente di aggiungere più serie temporali correlate a un grafico con un'espressione. Il grafico viene aggiornato in modo dinamico per includere nuovi parametri, che vengono aggiunti successivamente e soddisfano i criteri di ricerca. Per ulteriori informazioni, consulta Utilizzo delle espressioni di ricerca nei grafici. Non puoi creare un allarme in base a un'espressione SEARCH. Questo perché le espressioni di ricerca restituiscono più serie temporali e un allarme basato su un'espressione matematica può		✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			osservare solo una serie temporale. Se hai effettuato l'accesso a un account di monitoraggio in modalità osservabile CloudWatch su più account, la funzione RICERCA trova le metriche negli account di origine e nell'account di monitoraggio.		
SERVICE_QUOTA	TS che è un parame di utilizzo	TS	Restituisce la quota di servizio per il parametro di utilizzo specificato. Puoi utilizzare questa opzione per visualizzare il modo in cui l'utilizzo corrente si rapporta alla quota e per impostare gli allarmi che ti avvisano quando ti avvicini alla quota. Per ulteriori informazioni, consulta AWS metriche di utilizzo.		✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
SEZIONE	(TS[], S, S) or (TS[], S)	TS[] TS	Recupera parte di un array di serie temporali. Ciò è particolarmente utile se combinato con SORT. Ad esempio, è possibile escludere il risultato principale da un array di serie temporali. È possibile utilizzare due argomenti scalari per definire il set di serie temporali che si desidera ottenere. I due scalari definiscono l'inizio (inclusivo) e la fine (esclusivo) dell'array da ottenere. L'array è a indice zero, quindi la prima serie temporale nell'array è la serie temporale 0. In alternativa, puoi specificare un solo valore e CloudWatch restituire tutte le serie temporali che iniziano con quel valore.	SLICE(SORT(METRICS(), SUM, DESC), 0, 10) restituisce i dieci parametri dall'array dei parametri nella richiesta che hanno il valore SUM più alto. SLICE(SORT(METRICS(), AVG, ASC), 5) ordina l'array dei parametri in base alla statistica AVG, quindi restituisce tutte le serie temporali ad eccezione delle cinque con AVG più basso.	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
SORT	(TS[], FUNCT SORT_(R) (TS[], FUNCT SORT_(R, S)	TS[]	Ordina un array di serie temporali in base alla funzione specificata. La funzione utilizzata può essere AVG, MIN, MAX o SUM. L'ordinamento può essere ASC per crescente (valori più bassi prima) o DESC per ordinare prima i valori più alti. È possibile specificare facoltativamente un numero dopo l'ordinamento che funga da limite. Ad esempio, specificando un limite di 5 vengono restituite solo le prime 5 serie temporali dall'ordinamento. Quando questa funzione matematica viene visualizzata su un grafico, anche le etichette di ogni parametro nel grafico vengono ordinate e numerate.	SORT(METRICS(), AVG, DESC, 10) calcola il valore medio di ogni serie temporale, ordina le serie temporali con i valori più alti all'inizio dell'ordinamento e restituisce solo le 10 serie temporali con le medie più alte. SORT(METRICS(), MAX, ASC) ordina l'array dei parametri in base alla statistica MAX, quindi le restituisce tutte in ordine crescente.	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
STDDEV	TS TS[]	S TS	Lo STDDEV di una singola serie temporale restituisce un valore scalare che rappresenta la deviazione standard di tutti i punti di dati del parametro. Lo STDDEV di una gamma di serie temporali restituisce una singola serie temporale. **Note** Si consiglia di non utilizzare questa funzione negli CloudWatch allarmi se si desidera che la funzione restituisca uno scalare. Ad esempio, `STDDEV(m2)` ogni volta che un allarme valuta se cambiare stato, CloudWatch tenta di recuperare	m1/STDDEV(m1) STDDEV(METRICS())	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			un numero maggiore di punti dati rispetto al numero specificato come Periodi di valutazione. Questa funzione agisce in modo diverso quando vengono richiesti dati aggiuntivi.		

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
SUM	TS TS[]	S TS	La SUM di una singola serie temporale restituisce un valore scalare che rappresenta la somma dei valori di tutti i punti di dati del parametro. La funzione SUM di un array di serie temporali restituisce una singola serie temporale. **Note** Si consiglia di non utilizzare questa funzione negli CloudWatch allarmi se si desidera che la funzione restituisca uno scalare. Ad esempio, `SUM(m1)`. Ogni volta che un allarme valuta se cambiare stato, CloudWatch tenta di recuperare	SUM(METRICS())/SUM(m1) SUM([m1,m2]) SUM(METRICS("errors"))/SUM(METRICS("requests"))*100	✓

Funzione	Argome	Tipo restituito *	Descrizione	Esempi	Supportato per multi-account?
			un numero maggiore di punti dati rispetto al numero specificato come Periodi di valutazione. Questa funzione agisce in modo diverso quando vengono richiesti dati aggiuntivi.		
TIME_SERIES	S	TS	Restituisce una serie temporale non sparse in cui ogni valore è impostato su un argomento scalare.	TIME_SERIES(MAX(m1)) TIME_SERIES(5*AVG(m1)) TIME_SERIES(10)	✓

*L'utilizzo solo di una funzione che restituisce solo un numero scalare non è valido, così come tutti i risultati finali di espressioni devono essere una singola serie temporale o un array di serie temporali. Al contrario, utilizzare queste funzioni come parte di un'espressione più grande che restituisce una serie temporale.

Utilizzo delle espressioni IF

Utilizzare IF insieme a un operatore di confronto per filtrare i punti dati da una serie temporale o creare una serie temporale mista composta da più serie temporali raccolte.

IF utilizza gli argomenti seguenti:

```
IF(condition, trueValue, falseValue)
```

La condizione valuta FALSE se il valore del punto di dati di condizione è 0 e TRUE se il valore della condizione è qualsiasi altro valore, se tale valore è positivo o negativo. Se la condizione è una serie temporale, viene valutata separatamente per ogni timestamp.

Di seguito sono elencate le sintassi valide. Per ciascuna di queste sintassi, l'output è una singola serie temporale.

- IF(TS ***Comparison Operator*** S, S | TS, S | TS)

  > **Note**
  >
  > Se `TS comparison operator S` è TRUE ma `metric2` non ha un punto dati corrispondente, l'output sarà 0.

- IF(TS, TS, TS)
- IF(TS, S, TS)
- IF(TS, TS, S)
- IF(TS, S, S)
- IF(S, TS, TS)

Le sezioni seguenti forniscono ulteriori dettagli ed esempi per queste sintassi.

IF(TS ***Comparison Operator*** S, scalar2 | metric2, scalar3 | metric3)

Il corrispondente valore della serie temporale di output:

- ha il valore di scalar2 o metric2, se TS ***Comparison Operator*** S è TRUE
- ha il valore di scalar3 o metric3, se TS ***Comparison Operator*** S è FALSE

- ha il valore 0 se l'***operatore di confronto*** TS è TRUE e il punto dati corrispondente in metric2 non esiste.
- ha il valore 0 se l'***operatore di confronto*** TS è FALSE e il punto dati corrispondente in metric3 non esiste.
- è una serie temporale vuota, se il punto di dati corrispondente di non esiste in metric3, o se scalar3/metric3 viene omesso dall'espressione

IF(metric1, metric2, metric3)

Per ogni punto di dati di metric1, il corrispondente valore della serie temporale di output:

- ha il valore di metric2, se il punto di dati corrispondente di metric1 è TRUE.
- ha il valore di metric3, se il punto di dati corrispondente di metric1 è FALSE.
- ha il valore 0, se il punto di dati corrispondente di metric1 è TRUE e il punto di dati corrispondente non esiste in metric2.
- viene eliminato, se il punto dati corrispondente di metric1 è FALSE e il punto dati corrispondente non esiste in metric3
- viene eliminato se il punto dati corrispondente di metric1 è FALSE e metric3 viene omesso dall'espressione.
- viene eliminato, se manca il punto dati corrispondente di metric1.

Nella tabella seguente viene illustrato un esempio per questa sintassi.

Parametro o funzione	Valori
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
(metric3)	[0, 0, 20, -, 20]
IF(metric1, metric2, metric3)	[30, 0, 20, 0, -]

IF(metric1, scalar2, metric3)

Per ogni punto di dati di metric1, il corrispondente valore della serie temporale di output:

- ha il valore di scalar2, se il punto dati corrispondente di metric1 è TRUE.
- ha il valore di metric3, se il punto di dati corrispondente di metric1 è FALSE.
- viene eliminato, se il punto di dati corrispondente di metric1 è FALSE e il punto di dati corrispondente non esiste su metric3 o se metric3 viene omesso dall'espressione.

Parametro o funzione	Valori
(metric1)	[1, 1, 0, 0, -]
scalar2	5
(metric3)	[0, 0, 20, -, 20]
IF(metric1, scalar2, metric3)	[5, 5, 20, -, -]

IF(metric1, metric2, scalar3)

Per ogni punto di dati di metric1, il corrispondente valore della serie temporale di output:

- ha il valore di metric2, se il punto di dati corrispondente di metric1 è TRUE.
- ha il valore di scalar3, se il punto di dati corrispondente di metric1 è FALSE.
- ha il valore 0, se il punto di dati corrispondente di metric1 è TRUE e il punto di dati corrispondente non esiste in metric2.
- viene eliminato se il punto dati corrispondente in metric1 non esiste.

Parametro o funzione	Valori
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
scalar3	5
IF(metric1, metric2, scalar3)	[30, 0, 5, 5, -]

IF(scalar1, metric2, metric3)

Il corrispondente valore della serie temporale di output:

- ha il valore di metric2, se scalar1 è TRUE.
- ha il valore di metric3, se scalar1 è FALSE.
- è una serie temporale vuota, se metric3 viene omesso dall'espressione.

Esempi di casi d'uso per le espressioni IF

Gli esempi seguenti illustrano i possibili usi della funzione IF .

- Per visualizzare solo i valori inferiori di un parametro:

  IF(metric1<400, metric1)
- Per modificare ogni punto di dati in un parametro in uno dei due valori, per mostrare valori massimi e minimi relativi del parametro originale:

  IF(metric1<400, 10, 2)
- Per visualizzare un 1 per ogni timestamp in cui la latenza supera la soglia e visualizzare uno 0 per tutti gli altri punti di dati:

  IF(latency>threshold, 1, 0)

Usa la matematica metrica con l'operazione API GetMetricData

Puoi utilizzare `GetMetricData` per eseguire calcoli utilizzando espressioni matematiche, nonché per recuperare batch di dati di parametri di grandi dimensioni in una chiamata API. Per ulteriori informazioni, consulta. GetMetricData

Rilevamento di anomalie sulla matematica del parametro

Il rilevamento di anomalie sulla matematica del parametro è una funzione che è possibile utilizzare per creare allarmi di rilevamento anomalie su singole metriche e output di espressioni matematiche del parametro. È possibile utilizzare queste espressioni per creare grafici che visualizzano le bande di rilevamento delle anomalie. La funzione supporta funzioni aritmetiche di base, operatori logici e confronto e la maggior parte delle altre funzioni.

Il rilevamento di anomalie sulla matematica del parametro non supporta le seguenti funzioni:

- Espressioni che contengono più di unaANOMALY_DETECTION_BANDnella stessa riga.
- Espressioni che contengono più di 10 metriche o espressioni matematiche.
- Espressioni che contengono ilMETRICHEespressione.
- Espressioni che contengono ilRICERCAfunzione.
- Espressioni che utilizzano la funzione DP_PERF_INSIGHTS.
- Espressioni che utilizzano metriche con periodi diversi.
- Rilevatori di anomalie matematiche metriche che utilizzano metriche ad alta risoluzione come input.

Per ulteriori informazioni su questa funzionalità, consulta Using CloudWatch anomaly detection nella Amazon CloudWatch User Guide.

Utilizzo delle espressioni di ricerca nei grafici

Le espressioni di ricerca sono un tipo di espressione matematica che puoi aggiungere ai grafici. CloudWatch Le espressioni di ricerca consentono di aggiungere rapidamente più parametri correlati a un grafico. Inoltre, consentono di creare grafici dinamici che aggiungono automaticamente i parametri adeguati per la visualizzazione, anche se tali parametri non esistevano al momento della creazione del grafico.

Ad esempio, puoi creare un'espressione di ricerca che mostra il parametro `AWS/EC2` `CPUUtilization` per tutte le istanze della regione. Se successivamente avvii una nuova istanza, la funzione `CPUUtilization` della nuova istanza viene aggiunta automaticamente al grafico.

Quando utilizzi un'espressione di ricerca in un grafico, la ricerca trova l'espressione di ricerca nei nomi parametro, nello spazio dei nomi, nei nomi e nei valori delle dimensioni. Puoi utilizzare operatori booleani per ricerche più complesse e potenti. Un'espressione di ricerca può trovare solo i parametri che hanno riportato dati nelle ultime due settimane.

Non puoi creare un allarme in base all'espressione SEARCH. Questo perché le espressioni di ricerca restituiscono più serie temporali e un allarme basato su un'espressione matematica può osservare solo una serie temporale.

Se utilizzi un account di monitoraggio in modalità osservabile su CloudWatch più account, le espressioni di ricerca possono trovare le metriche negli account di origine collegati a quell'account di monitoraggio.

Argomenti

CloudWatch sintassi delle espressioni di ricerca

Un'espressione di ricerca valida ha il seguente formato.

```
SEARCH(' {Namespace, DimensionName1, DimensionName2, ...} SearchTerm', 'Statistic')
```

Ad esempio:

```
SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')
```

- La prima parte della query dopo la parola `SEARCH`, fra parentesi graffe, è lo schema di parametri in cui eseguire la ricerca. Lo schema di parametri include un spazio dei nomi parametro e uno o più nomi delle dimensioni. L'inclusione di uno schema di parametri in una query di ricerca è opzionale. Se specificato, lo schema di parametri deve includere uno spazio dei nomi e può opzionalmente includere uno o più nomi delle dimensioni validi per tale spazio dei nomi.

  Non è necessario utilizzare virgolette all'interno dello schema di parametri a meno che lo spazio dei nomi o il nome delle dimensioni non includa spezi o caratteri non alfanumerici. In questo caso, dovrai inserire fra virgolette doppie il nome che contiene tali caratteri.
- Anche la funzione `SearchTerm` è facoltativa, ma una ricerca valida deve includere lo schema di parametri, la funzione `SearchTerm` o entrambi. La funzione `SearchTerm` in genere include uno o più ID account, nomi parametro o valori di dimensioni. La funzione `SearchTerm` può includere più termini da cercare, sia con corrispondenze parziali che con corrispondenze esatte. Può anche includere operatori booleani.

  L'utilizzo di un ID account in a `SearchTerm` funziona solo negli account configurati come account di monitoraggio per l'osservabilità CloudWatch tra account. La sintassi per un ID account in `SearchTerm` è `:aws.AccountId = "`*`444455556666`*`"`. Puoi anche usare `'LOCAL'` per specificare l'account di monitoraggio stesso: `:aws.AccountId = 'LOCAL'`

  Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

La funzione `SearchTerm` può includere uno o più identificatori, ad esempio `MetricName=` come in questo esempio, ma l'utilizzo di tali identificatori non è obbligatorio.

Lo schema dei parametri e la funzione `SearchTerm` devono essere racchiusi insieme fra virgolette singole.

- `Statistic`È il nome di qualsiasi statistica valida CloudWatch . Deve essere racchiusa tra virgolette singole. Per ulteriori informazioni, consulta Statistiche.

Nell'esempio precedente, era stata effettuata una ricerca nello spazio dei nomi `AWS/EC2` dei parametri con `InstanceId` come nome di dimensione. Restituisce tutti i parametri `CPUUtilization` individuati e il grafico mostra la statistica `Average`.

Un'espressione di ricerca può trovare solo i parametri che hanno riportato dati nelle ultime due settimane.

Limiti dell'espressione di ricerca

Le dimensioni massime della query dell'espressione di ricerca è di 1024 caratteri. In un grafico puoi avere fino a 100 espressioni di ricerca. Un grafico può possibile visualizzare fino a 500 serie temporali.

CloudWatch espressioni di ricerca: tokenizzazione

Quando si specifica a`SearchTerm`, la funzione di ricerca cerca i token, che sono sottostringhe generate CloudWatch automaticamente da nomi di metriche completi, nomi di dimensione, valori di dimensione e namespace. CloudWatch genera token distinti dalla maiuscola camel-case nella stringa originale. I caratteri numerici fungono anche da inizio dei nuovi token, mentre i caratteri non alfanumerici fungono da delimitatori, creando token prima e dopo i caratteri non alfanumerici.

Una stringa continua dello stesso tipo del carattere di delimitazione del token crea un token.

Tutti i token generati sono in lettere minuscole. La tabella seguente mostra alcuni esempi di token generati.

Stringa originale	Token generato
CustomCount1	`customcount1` , `custom`, `count`, `1`

Stringa originale	Token generato
SDBFailure	`sdbfailure` , `sdb`, `failure`
Project2-trial333	`project2trial333` , `project`, `2`, `trial`, `333`

CloudWatch espressioni di ricerca: corrispondenze parziali

Quando si specifica a`SearchTerm`, anche il termine di ricerca viene tokenizzato. CloudWatchtrova le metriche in base alle corrispondenze parziali, ovvero le corrispondenze di un singolo token generato dal termine di ricerca a un singolo token generato da un nome di metrica, uno spazio dei nomi, un nome di dimensione o un valore di dimensione.

Le ricerche di corrispondenze parziali di un singolo token prevedono una distinzione tra lettere maiuscole e minuscole. Ad esempio, l'utilizzo di uno dei seguenti termini di ricerca può restituire il parametro `CustomCount1`:

- `count`
- `Count`
- `COUNT`

Tuttavia, utilizzando `couNT` come termine di ricerca non sarà possibile trovare `CustomCount1` perché l'uso di lettere maiuscole e minuscole nel termine di ricerca `couNT` è tokenizzato in `cou` e in `NT`.

Le ricerche, inoltre, possono restituire token compositi, ovvero più token che appaiono consecutivamente nel nome originale. Per individuare un token composito, la ricerca prevede una distinzione tra lettere maiuscole e minuscole. Ad esempio, se il termine originale è `CustomCount1`, le ricerche `CustomCount` o `Count1` andranno a buon fine, mentre quelle di `customcount` o `count1` non produrranno risultati.

CloudWatch espressioni di ricerca: corrispondenze esatte

Puoi definire una ricerca per trovare solo le corrispondenze esatte del termine di ricerca racchiudendo fra virgolette doppie la parte del termine di ricerca che richiede una corrispondenza esatta. Le virgolette doppie vengono racchiuse nelle virgolette singole utilizzate per l'intero termine di ricerca. Ad esempio, **`SEARCH(' {MyNamespace}, "CustomCount1" ', 'Maximum')`** trova la stringa

esatta `CustomCount1` se esiste come nome parametro, nome o valore di dimensione nello spazio dei nomi `MyNamespace`. Tuttavia, le ricerche di **`SEARCH(' {MyNamespace}, "customcount1" ', 'Maximum')`** o **`SEARCH(' {MyNamespace}, "Custom" ', 'Maximum')`** non troveranno questa stringa.

Puoi combinare termini di corrispondenza parziale e di corrispondenza esatta in una sola espressione di ricerca. Ad esempio, **`SEARCH(' {AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')`** restituisce il parametro Elastic Load Balancing denominato `ConsumedLCUs` così come tutti i parametri o le dimensioni Elastic Load Balancer che includono il token `flow`.

L'utilizzo della corrispondenza esatta rappresenta un buon metodo per individuare nomi con caratteri speciali, come caratteri non alfanumerici o spazi, come nell'esempio seguente.

```
SEARCH(' {"My Namespace", "Dimension@Name"}, "Custom:Name[Special_Characters" ',
 'Maximum')
```

CloudWatch espressioni di ricerca: esclusione di uno schema metrico

Tutti gli esempi mostrati finora includono uno schema di parametri, racchiuso fra parentesi graffe. Anche le ricerche che escludono uno schema di parametri sono valide.

Ad esempio, **`SEARCH(' "CPUUtilization" ', 'Average')`** restituisce tutti i nomi parametro, i nomi e i valori delle dimensioni e gli spazi dei nomi che rappresentano una corrispondenza esatta della stringa `CPUUtilization`. Nei namespace delle AWS metriche, questo può includere metriche di diversi servizi tra cui Amazon EC2, Amazon ECS e altri. SageMaker

Per restringere questa ricerca a un solo AWS servizio, la best practice consiste nello specificare lo spazio dei nomi e tutte le dimensioni necessarie nello schema delle metriche, come nell'esempio seguente. Benché questa soluzione restringa la ricerca allo spazio dei nomi `AWS/EC2`, restituirà comunque i risultati di altri parametri se hai definito `CPUUtilization` come un valore di dimensione per tali parametri.

```
SEARCH(' {AWS/EC2, InstanceType} "CPUUtilization" ', 'Average')
```

In alternativa, puoi aggiungere lo spazio dei nomi in `SearchTerm`, come nell'esempio seguente. Tuttavia, in questo esempio, la ricerca restituirebbe come corrispondenza qualsiasi stringa `AWS/EC2`, anche se si fosse trattato di un valore o di un nome di dimensione personalizzato.

```
SEARCH(' "AWS/EC2" MetricName="CPUUtilization" ', 'Average')
```

CloudWatch espressioni di ricerca: specificazione dei nomi delle proprietà nella ricerca

La seguente ricerca di corrispondenze esatte di `"CustomCount1"` restituisce tutti i parametri che hanno stesso nome.

```
SEARCH(' "CustomCount1" ', 'Maximum')
```

Tuttavia, restituisce anche parametri con i nomi e valori di dimensioni o spazi dei nomi di `CustomCount1`. Per strutturare ulteriormente la ricerca, puoi specificare il nome della proprietà del tipo dell'oggetto da trovare nelle ricerche. L'esempio seguente ricerca tutti gli spazi dei nomi e restituisce i parametri denominati `CustomCount1`.

```
SEARCH(' MetricName="CustomCount1" ', 'Maximum')
```

Puoi anche utilizzare gli spazi dei nomi e le coppie nome/valore delle dimensioni come nomi delle proprietà, come negli esempi seguenti. Il primo di questi esempi mostra che come sia possibile utilizzare i nomi delle proprietà con ricerche di corrispondenze parziali.

```
SEARCH(' InstanceType=micro ', 'Average')
```

```
SEARCH(' InstanceType="t2.micro" Namespace="AWS/EC2" ', 'Average')
```

CloudWatch espressioni di ricerca: caratteri non alfanumerici

I caratteri non alfanumerici fungono da delimitatori e contrassegnano il punto in cui i nomi parametro, le dimensioni, gli spazi dei nomi e i termini di ricerca devono essere separati in token. Quando termini sono tokenizzati, i caratteri non alfanumerici vengono esclusi e non vengono visualizzati nei token. Ad esempio, `Network-Errors_2` genera i token `network`, `errors` e `2`.

Il termine di ricerca può includere caratteri non alfanumerici. Se questi caratteri appaiono nel termine di ricerca, possono specificare token compositi in una corrispondenza parziale. Ad esempio, tutte le ricerche seguenti individuerebbero i parametri denominati `Network-Errors-2` o `NetworkErrors2`.

```
network/errors
network+errors
network-errors
Network_Errors
```

Quando effettui la ricerca di valore esatto, gli eventuali caratteri non alfanumerici utilizzate nella ricerca esatta devono essere i caratteri corretti che appaiono nella stringa da ricercare. Ad esempio, per trovare `Network-Errors-2`, la ricerca `"Network-Errors-2"` andrà a buon fine, a differenza di quella di `"Network_Errors_2"`.

Quando esegui una ricerca di corrispondenze esatte, i seguenti caratteri devono essere preceduti da barra rovesciata.

```
" \ ( )
```

Ad esempio, per trovare il nome parametro `Europe\France Traffic(Network)` con corrispondenza esatta, utilizza il termine di ricerca **`"Europe\\France Traffic\(Network\)"`**

CloudWatch espressioni di ricerca: operatori booleani

La ricerca supporta l'utilizzo degli operatori booleani `AND`, `OR` e `NOT` all'interno del `SearchTerm`. Gli operatori booleani sono racchiusi fra le stesse virgolette singole utilizzate per racchiudere l'intero termine di ricerca. Gli operatori booleani prevedono una distinzione tra lettere maiuscole e minuscole. Pertanto, `and`, `or` e `not` non sono operatori booleani validi.

Puoi utilizzare `AND` esplicitamente nella ricerca, ad esempio **`SEARCH('{AWS/EC2,InstanceId} network AND packets', 'Average')`**. Il mancato utilizzo di operatori booleani fra i termini di ricerca ne implica la relativa ricerca come se si trattasse di un operatore `AND`. Pertanto, **`SEARCH(' {AWS/EC2,InstanceId} network packets ', 'Average')`** restituirà gli stessi risultati di ricerca.

Utilizza `NOT` per escludere sottoinsiemi di dati dai risultati. Ad esempio, **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT i-1234567890123456 ', 'Average')`** restituirà `CPUUtilization` per tutte le istanze, tranne per quella `i-1234567890123456`. Puoi anche utilizzare una clausola `NOT` come unico termine di ricerca. Ad esempio, **`SEARCH( 'NOT Namespace=AWS ', 'Maximum')`** restituisce tutti i parametri personalizzati (parametri con spazi dei nomi che non includono `AWS`).

Puoi utilizzare più frasi `NOT` in una query. Ad esempio, **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT "ProjectA" NOT "ProjectB" ', 'Average')`** restituirà `CPUUtilization` per tutte le istanze nella regione, tranne per quelle con valori di dimensioni pari a `ProjectA` o `ProjectB`.

Puoi combinare operatori booleani per ricerche più potenti e dettagliate, come negli esempi seguenti. Utilizza le parentesi per raggruppare gli operatori.

I prossimi due esempi restituiscono tutti i nomi parametro contenenti `ReadOps` dagli spazi dei nomi EC2 ed EBS.

```
SEARCH(' (EC2 OR EBS) AND MetricName=ReadOps ', 'Maximum')
```

```
SEARCH(' (EC2 OR EBS) MetricName=ReadOps ', 'Maximum')
```

L'esempio seguente restringe la precedente ricerca solo ai risultati che includono `ProjectA`, che potrebbe essere il valore di una dimensione.

```
SEARCH(' (EC2 OR EBS) AND ReadOps AND ProjectA ', 'Maximum')
```

L'esempio seguente utilizza il raggruppamento nidificato. Restituisce i parametri Lambda per `Errors` da tutte le funzioni e `Invocations` di funzioni con nomi che includono le stringhe `ProjectA` o `ProjectB`.

```
SEARCH(' {AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

CloudWatch espressioni di ricerca: utilizzo di espressioni matematiche

Puoi utilizzare un'espressione di ricerca all'interno di una matematica in un grafico.

Ad esempio, **`SUM(SEARCH(' {AWS/Lambda, FunctionName} MetricName="Errors" ', 'Sum'))`** restituisce la somma del parametro `Errors` di tutte le funzioni Lambda.

Utilizzando righe separate per l'espressione di ricerca e quella matematica potrebbe produrre risultati più utili. Ad esempio, supponi di utilizzare le seguenti due espressioni in un grafico. Nella prima riga sono visualizzare righe `Errors` separate per tutte le funzioni Lambda. L'ID di questa espressione è `e1`. La seconda riga aggiunge un'altra riga in cui è mostrata la somma degli errori di tutte le funzioni.

```
SEARCH(' {AWS/Lambda, FunctionName}, MetricName="Errors" ', 'Sum')
SUM(e1)
```

CloudWatch esempi di espressioni di ricerca

Gli esempi seguenti mostrano vari utilizzi e sintassi delle espressione di ricerca. Iniziamo con la ricerca di `CPUUtilization` tra tutte le istanze nella regione e quindi esaminiamo le variazioni.

In questo esempio viene visualizzata una riga per ogni istanza nella regione, mostrando il parametro `CPUUtilization` dello spazio dei nomi `AWS/EC2`.

```
SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" ', 'Average')
```

Modificando `InstanceId` in `InstanceType`, il grafico mostrerà una riga per ogni tipo di istanza usato nella regione. I dati provenienti da tutte le istanze di ciascun tipo sono aggregati in una riga per quel tipo di istanza.

```
SEARCH(' {AWS/EC2,InstanceType} MetricName="CPUUtilization" ', 'Average')
```

L'esempio seguente aggrega il `CPUUtilization` per tipo di istanza e visualizza una riga per ciascun tipo di istanza che include la stringa `micro`.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType=micro MetricName="CPUUtilization" ',
 'Average')
```

Questo esempio limita l'esempio precedente, modificando `InstanceType` in una ricerca esatte delle istanze t2.micro.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType="t2.micro" MetricName="CPUUtilization" ',
 'Average')
```

La seguente ricerca rimuove la parte `{metric schema}` della query. Pertanto, nel grafico viene visualizzato il parametro `CPUUtilization` di tutti gli spazi dei nomi. Ciò può restituire parecchi risultati perché il grafico include più righe per la `CPUUtilization` metrica di ciascun AWS servizio, aggregate in base a dimensioni diverse.

```
SEARCH('MetricName="CPUUtilization" ', 'Average')
```

Per limitare tali risultati, puoi specificare due determinati spazi dei nomi parametro.

```
SEARCH('MetricName="CPUUtilization" AND ("AWS/ECS" OR "AWS/ES") ', 'Average')
```

L'esempio precedente è l'unico modo per eseguire una ricerca di più spazi dei nomi specifici con una query di ricerca, poiché è possibile specificare un solo schema di parametri in ciascuna query. Tuttavia, per ottimizzare la strutturazione, potresti utilizzare due query nel grafico, come mostrato

nell'esempio seguente. In questo esempio, inoltre, la struttura risulta più solida grazie alla specifica di una dimensione da utilizzare per aggregare i dati Amazon ECS.

```
SEARCH('{AWS/ECS ClusterName}, MetricName="CPUUtilization" ', 'Average')
SEARCH(' {AWS/EBS} MetricName="CPUUtilization" ', 'Average')
```

L'esempio seguente restituisce il parametro Elastic Load Balancing denominato `ConsumedLCUs` così come tutti i parametri o le dimensioni Elastic Load Balancer che includono il token `flow`.

```
SEARCH('{AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')
```

L'esempio seguente utilizza il raggruppamento nidificato. Restituisce i parametri Lambda per `Errors` da tutte le funzioni e `Invocations` di funzioni con nomi che includono le stringhe `ProjectA` o `ProjectB`.

```
SEARCH('{AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

L'esempio seguente visualizza tutti i parametri personalizzati, esclusi quelli generati dai servizi AWS .

```
SEARCH('NOT Namespace=AWS ', 'Average')
```

L'esempio seguente mostra i parametri con nomi parametro, spazio dei nomi, nomi e valori delle dimensioni contenenti la stringa `Errors` come parte del nome.

```
SEARCH('Errors', 'Average')
```

L'esempio seguente restringe tale ricerca alle corrispondenze esatte. Ad esempio, questa ricerca trova il nome parametro `Errors`, ma non i parametri `ConnectionErrors` o `errors`.

```
SEARCH(' "Errors" ', 'Average')
```

L'esempio seguente mostra come specificare nomi contenenti spazi o caratteri speciali nella parte dello schema di parametri del termine di ricerca.

```
SEARCH('{"Custom-Namespace", "Dimension Name With Spaces"}, ErrorCount ', 'Maximum')
```

CloudWatch esempi di espressioni di ricerca osservabili su più account

CloudWatch esempi di osservabilità tra account

Se hai effettuato l'accesso a un account configurato come account di monitoraggio nell'osservabilità CloudWatch tra account, puoi utilizzare la funzione RICERCA per restituire le metriche dagli account di origine specificati. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

L'esempio seguente recupera tutte le metriche Lambda dall'account con l'ID account 111122223333.

```
SEARCH(' AWS/Lambda :aws.AccountId = "111122223333" ', 'Average')
```

L'esempio seguente recupera tutte le metriche `AWS/EC2` da due account: 111122223333 e 777788889999.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR "777788889999") ', 'Average')
```

L'esempio seguente recupera tutti i parametri `AWS/EC2` dall'account di origine 111122223333 e dall'account di monitoraggio stesso.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR 'LOCAL') ', 'Average')
```

L'esempio seguente recupera la `SUM` del parametro `MetaDataToken` dall'account `444455556666` con la dimensione `InstanceId`.

```
SEARCH('{AWS/EC2,InstanceId} :aws.AccountId=444455556666 MetricName=\"MetadataNoToken
\"','Sum')
```

Crea un CloudWatch grafico con un'espressione di ricerca

Sulla CloudWatch console, puoi accedere alle funzionalità di ricerca quando aggiungi un grafico a una dashboard o utilizzando la vista Metriche.

Non puoi creare un allarme in base a un'espressione SEARCH. Questo perché le espressioni di ricerca restituiscono più serie temporali e un allarme basato su un'espressione matematica può osservare solo una serie temporale.

Per aggiungere un grafico con un'espressione di ricerca a un pannello di controllo esistente

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel pannello di navigazione, seleziona Dashboards (Pannelli di controllo), quindi seleziona un pannello di controllo.
3. Seleziona Add widget (Aggiungi widget).
4. Seleziona Line (Linea) o Stacked area (Area in pila), quindi seleziona Configure (Configura).
5. Nella scheda Graphed metrics (Parametri del grafico), scegli Add a math expression (Aggiungi un'espressione matematica).
6. In Details (Dettagli), immetti l'espressione di ricerca desiderata. Ad esempio, **`SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')`**
7. (Opzionale) Per aggiungere al grafico un'altra espressione di ricerca o un'espressione matematica, scegli Add a math expression (Aggiungi una espressione matematica)
8. (Facoltativo) Dopo aver aggiunto un'espressione di ricerca, è possibile specificare che un'etichetta dinamica venga visualizzata sulla legenda del grafico legenda per ogni parametro. Le etichette dinamiche visualizzano una statistica sul parametro e vengono aggiornate automaticamente quando il pannello di controllo o il grafico viene aggiornato. Per aggiungere un'etichetta dinamica, scegli Graphed metrics (Parametri definiti), quindi Dynamic labels (Etichette dinamiche).

   Per impostazione predefinita, i valori dinamici aggiunti all'etichetta vengono visualizzati all'inizio dell'etichetta. È quindi possibile fare clic sul valore Label (Etichetta) per il parametro per modificare l'etichetta. Per ulteriori informazioni, consulta Utilizzo di etichette dinamiche.
9. (Opzionale) Per aggiungere un singolo parametro grafico, scegli la scheda All metrics (Tutti i parametri) e sposti su quello desiderato.
10. (Opzionale) Per modificare l'intervallo temporale mostrato sul grafico, scegli custom (personalizzato) nella parte superiore del grafico o uno dei periodi temporali a sinistra di custom (personalizzato).
11. (Facoltativo) Le annotazioni orizzontali consentono agli utenti del pannello di controllo di vedere rapidamente quando un parametro è aumentato fino a un determinato livello o se il parametro si trova all'interno di un intervallo predefinito. Per aggiungere un'annotazione orizzontale, scegli Graph options (Opzioni del grafico), Add horizontal annotation (Aggiungi annotazione orizzontale):

    a. In Label (Etichetta), immetti un'etichetta per l'annotazione.
    b. In Value (Valore), immetti un valore del parametro in cui è visualizzata l'annotazione orizzontale.

c. In Fill (Riempi), specifica se utilizzare un'ombreggiatura di riempimento con questa annotazione. Ad esempio, seleziona `Above` o `Below` per l'area corrispondente da riempire. Se specifichi `Between`, viene visualizzato un altro campo `Value` e viene riempita l'area del grafico tra i due valori.

d. Se il grafico include più parametri, in Axis (Asse), specificare se i numeri di `Value` fanno riferimento al parametro associato all'asse y sinistro o all'asse y destro.

Puoi modificare il colore di riempimento di un'annotazione selezionando il quadrato colori nella colonna sinistra dell'annotazione.

Ripeti queste fasi per aggiungere più annotazioni orizzontali allo stesso grafico.

Per nascondere un'annotazione, deseleziona la casella di controllo nella colonna sinistra per tale annotazione.

Per cancellare un'annotazione, seleziona x nella colonna Actions (Operazioni).

12. (Facoltativo) Le annotazioni verticali consentono di contrassegnare milestone in un grafico, ad esempio eventi operativi o l'inizio e la fine di una distribuzione. Per aggiungere un'annotazione verticale, scegli Graph options (Opzioni del grafico), quindi Add vertical annotation (Aggiungi annotazione verticale):

a. In Label (Etichetta), immetti un'etichetta per l'annotazione. Per mostrare solo la data e l'orario dell'annotazione, lasciare il campo Label (Etichetta) vuoto.

b. In Date (Data), specificare la data e l'ora in cui è visualizzata l'annotazione verticale.

c. In Fill (Riempi), specificare se utilizzare un'ombreggiatura di riempimento prima o dopo un'annotazione verticale o tra due annotazioni verticali. Ad esempio, seleziona `Before` o `After` per l'area corrispondente da riempire. Se specifichi `Between`, viene visualizzato un altro campo `Date` e viene riempita l'area del grafico tra i due valori.

Ripeti queste fasi per aggiungere più annotazioni verticali allo stesso grafico.

Per nascondere un'annotazione, deseleziona la casella di controllo nella colonna sinistra per tale annotazione.

Per cancellare un'annotazione, seleziona x nella colonna Actions (Operazioni).

13. Seleziona Crea widget.

14. Seleziona Salva pannello di controllo.

Per utilizzare la vista Metrics (Parametri) per la rappresentazione grafica dei parametri cercati

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nel campo di ricerca, immetti il token da cercare: ad esempio, **`cpuutilization t2.small`**.

   Verranno visualizzati i risultati che corrispondono alla ricerca.
4. Per rappresentare graficamente tutti i parametri che corrispondono alla ricerca, scegli Graph search (Ricerca del grafico).

   oppure

   Per perfezionare la ricerca, scegli uno degli spazi dei nomi visualizzati nei risultati di ricerca.
5. Se hai selezionato uno spazio dei nomi per limitare i risultati, puoi eseguire le operazioni descritte di seguito.

   a. Per rappresentare graficamente uno o più parametri, seleziona la casella di controllo accanto a ciascun parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.

   b. Per delimitare la ricerca, passa il mouse su un nome parametro e scegli Add to search (Aggiungi alla ricerca) o Search for this only (Cerca solo questo).

   c. Per visualizzare la guida di un parametro, seleziona il nome parametro, quindi seleziona What is this? (Che cos'è questo?).

   I parametri selezionati vengono visualizzati nel grafico.
6. (Opzionale) Seleziona uno dei pulsanti nella barra di ricerca per modificare tale parte del termine di ricerca.
7. (Opzionale) Per aggiungere il grafico a un pannello di controllo, seleziona Actions (Operazioni), Add to dashboard (Aggiungi a pannello di controllo).

Ottenere le statistiche di un parametro

CloudWatch definizioni statistiche

Le statistiche sono aggregazioni di dati di parametri durante determinati periodi di tempo. Quando si tracciano grafici o si recuperano le statistiche per un parametro, è necessario specificare il valore Periodo (Periodo di tempo), ad esempio cinque minuti, da utilizzare per calcolare ogni valore statistico. Ad esempio, se il valore Period (Periodo di tempo) è 5 minuti, il valore Sum (Somma) è la somma di tutti i valori campione raccolti durante il periodo di cinque minuti, mentre il valore Minimum (Minimo) è il valore più basso raccolto durante il periodo di cinque minuti.

CloudWatch supporta le seguenti statistiche per le metriche.

- SampleCountè il numero di punti dati durante il periodo.
- Sum (Somma) è la somma dei valori di tutti i punti dati raccolti durante il periodo.
- Average (Media) è il valore di `Sum/SampleCount` durante il periodo specificato.
- Minimum (Minimo) è valore più basso osservato durante il periodo specificato.
- Maximum (Massimo) è il valore più alto osservato durante il periodo specificato.
- Percentile (p) (Percentile) indica lo stato relativo di un valore in un set di dati. Ad esempio, p95 è il 95° percentile e vuol dire che il 95% dei dati entro il periodo è inferiore a questo valore e il 5% dei dati è superiore a questo valore. I percentili aiutano a comprendere meglio la distribuzione dei dati del parametro.
- Trimmed mean (TM) (Media troncata) è la media di tutti i valori che si trovano tra due limiti specificati. I valori al di fuori dei limiti vengono ignorati quando viene calcolata la media. I limiti vengono definiti come uno o due numeri compresi tra 0 e 100, fino a 10 cifre decimali. I numeri possono essere valori assoluti o percentuali. Ad esempio, tm90 calcola la media dopo aver rimosso il 10% dei punti dati con i valori più alti. TM(2%:98%) calcola la media dopo aver rimosso i punti dati più bassi del 2% e i punti dati più alti del 2%. TM(150:1000) calcola la media dopo aver rimosso tutti i punti dati inferiori o uguali a 150 o superiori a 1000.
- Interquartile mean (IQM) (Media interquartile) è la media troncata dell'intervallo interquartile, o il 50% medio dei valori. Equivale a TM(25%:75%).
- Winsorized mean (WM) (Media winsorizzata) è simile alla media troncata. Tuttavia, con la media winsorizzata, i valori che si trovano al di fuori dell'edge non vengono ignorati, ma sono considerati uguali al valore al margine del limite appropriato. Dopo questa normalizzazione, viene calcolata la media. I limiti vengono definiti come uno o due numeri compresi tra 0 e 100, fino a 10 cifre

decimali. Ad esempio, wm98 calcola la media trattando il 2% dei valori più alti in modo che sia uguale al valore al 98° percentile. WM(10%:90%) calcola la media trattando il 10% più alto dei punti dati come valore del limite del 90% e trattando il 10% più basso dei punti dati come valore del limite del 10%.

- Percentile rank (PR) (Rango percentile) è la percentuale di valori che soddisfano una soglia fissa. Ad esempio, PR(:300) restituisce la percentuale di punti dati che hanno un valore pari o inferiore a 300. PR(100:2000) restituisce la percentuale di punti dati che hanno un valore compreso tra 100 e 2000.

  Il rango percentile è esclusivo nel limite inferiore e inclusivo nel limite superiore.
- Trimmed count (TC) (Conteggio troncato) è il numero di punti dati nell'intervallo scelto per una statistica media troncata. Ad esempio, tc90 restituisce il numero di punti dati che non includono punti dati che rientrano nel 10% più alto dei valori. TC(0.005:0.030) restituisce il numero di punti dati con valori compresi tra 0,005 (escluso) e 0,030 (incluso).
- Trimmed sum (TC) (Somma troncata) è la somma dei valori dei punti dati in un intervallo scelto per una statistica media troncata. È equivalente a (Media troncata) * (Conteggio troncato). Ad esempio, ts90 restituisce la somma dei punti dati che non includono punti dati che rientrano nel 10% più alto dei valori. TS(80%:) restituisce la somma dei valori dei punti dati, esclusi i punti dati con valori nell'80% più basso dell'intervallo di valori.

> **Note**
>
> Per Trimmed Mean (Media troncata), Trimmed Count (Conteggio troncato), Trimmed Sum (Somma troncata) e Winsorized Mean (Media winsorizzata), se si definiscono due limiti come valori fissi anziché percentuali, il calcolo include valori uguali al limite superiore, ma non i valori uguali al limite inferiore.

Sintassi

Per Trimmed Mean (Media troncata), Trimmed Count (Conteggio troncato), Trimmed Sum (Somma troncata) e Winsorized Mean (Media winsorizzata), si applicano le seguenti regole di sintassi:

- L'utilizzo di parentesi con uno o due numeri con segni di percentuale definisce i limiti da utilizzare come valori nel set di dati compresi tra i due percentili specificati. Ad esempio, TM(10%:90%) utilizza solo i valori compresi tra il decimo e il 90° percentile. TM(:95%) utilizza i valori dall'estremità più bassa del set di dati fino al 95° percentile, ignorando il 5% dei punti dati con i valori più alti.

- L'utilizzo di parentesi senza uno o due numeri con segni di percentuale definisce i limiti da utilizzare come valori nel set di dati compresi tra i valori espliciti specificati. Ad esempio, TC(80:500) utilizza solo i valori compresi tra 80 (escluso) e 500 (incluso). TC(:0.5) utilizza solo i valori che sono uguali o inferiori a 0,5.
- L'utilizzo di un numero senza parentesi consente di calcolare le percentuali, ignorando i punti dati superiori al percentile specificato. Ad esempio, tm99 calcola la media ignorando l'1% dei punti dati con il valore più alto. È uguale a TM(:99%).
- Trimmed Mean (Media troncata), Trimmed Count (Conteggio troncato), Trimmed Sum (Somma troncata) e Winsorized Mean (Media winsorizzata) possono essere abbreviati utilizzando lettere maiuscole quando si specifica un intervallo, ad esempio TM(5%:95%), TM(100:200) o TM(:95%). Possono essere abbreviati solo utilizzando lettere minuscole quando si specifica un solo numero, ad esempio tm99.

Casi d'uso della statistica

- Trimmed mean (Media troncata) è più utile per i parametri con una dimensione di esempio di grandi dimensioni, ad esempio la latenza della pagina Web. Ad esempio, tm99 ignora i valori anomali estremi che potrebbero derivare da problemi di rete o errori umani, per dare un numero più accurato per la latenza media delle richieste tipiche. Allo stesso modo, TM(10%:) ignora il 10% più basso dei valori di latenza, come quelli risultanti dai riscontri nella cache. E TM (10%:99%) esclude entrambi questi tipi di valori anomali. Ti consigliamo di utilizzare la media tagliata per il monitoraggio della latenza.
- È una buona idea tenere d'occhio il conteggio troncato ogni volta che si utilizza la media troncata, per assicurarti che il numero di valori utilizzati nei calcoli della media troncata sia sufficiente per essere statisticamente significativo.
- Il rango percentile consente di inserire valori in “contenitori” di intervalli, ed è possibile utilizzarlo per creare manualmente un istogramma. Per fare ciò, suddividi i tuoi valori in vari contenitori, come PR(:1), PR(1:5), PR(5:10) e PR(10:). Metti ciascun contenitore in una visualizzazione come grafici a barre così da avere un istogramma.

  Il rango percentile è esclusivo nel limite inferiore e inclusivo nel limite superiore.

Percentili rispetto alla media troncata

Un percentile come p99 e una media troncata come tm99 misurano valori simili, ma non identici. Sia p99 che tm99 ignorano l'1% dei punti dati con i valori più alti, che sono considerati valori anomali.

Dopo di che, p99 è valore massimo del restante 99%, mentre tm99 è la media del restante 99%. Se stai guardando la latenza delle richieste Web ,p99 indica la peggiore esperienza cliente, ignorando i valori anomali, mentre tm99 indica l'esperienza cliente media, ignorando i valori anomali.

La media troncata è una buona statistica di latenza da guardare se stai cercando di ottimizzare la tua esperienza cliente.

Requisiti per l'uso di percentili, media troncata e altre statistiche

CloudWatch necessita di punti dati grezzi per calcolare le seguenti statistiche:

- Percentili
- Media troncata
- Media Interquartile
- Media winsorizzata
- Somma troncata
- Conteggio troncato
- Rango percentile

Se pubblichi dati per una statistica personalizzata utilizzando un set di statistiche invece di dati non elaborati, puoi recuperare questi tipi di statistiche dei percentili per questi dati solo se risulta vera una delle seguenti condizioni:

- Il SampleCount valore del set di statistiche è 1 e Min, Max e Sum sono tutti uguali.
- Min e Max sono uguali e Sum è uguale a Min moltiplicato per. SampleCount

I seguenti AWS servizi includono metriche che supportano questi tipi di statistiche.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Sistema di bilanciamento del carico elastico
- Kinesis
- Amazon RDS

Inoltre questi tipi di statistiche non sono disponibili per i parametri quando uno qualsiasi dei valori dei parametri è un numero negativo.

Gli esempi seguenti mostrano come ottenere statistiche per le CloudWatch metriche delle tue risorse, come le istanze EC2.

Esempi

- Ottenimento di statistiche per una risorsa specifica
- Aggregazione di statistiche tra risorse
- Aggregazione di statistiche per gruppo Auto Scaling
- Aggregazione di statistiche per Amazon Machine Image (AMI)

Ottenimento di statistiche per una risorsa specifica

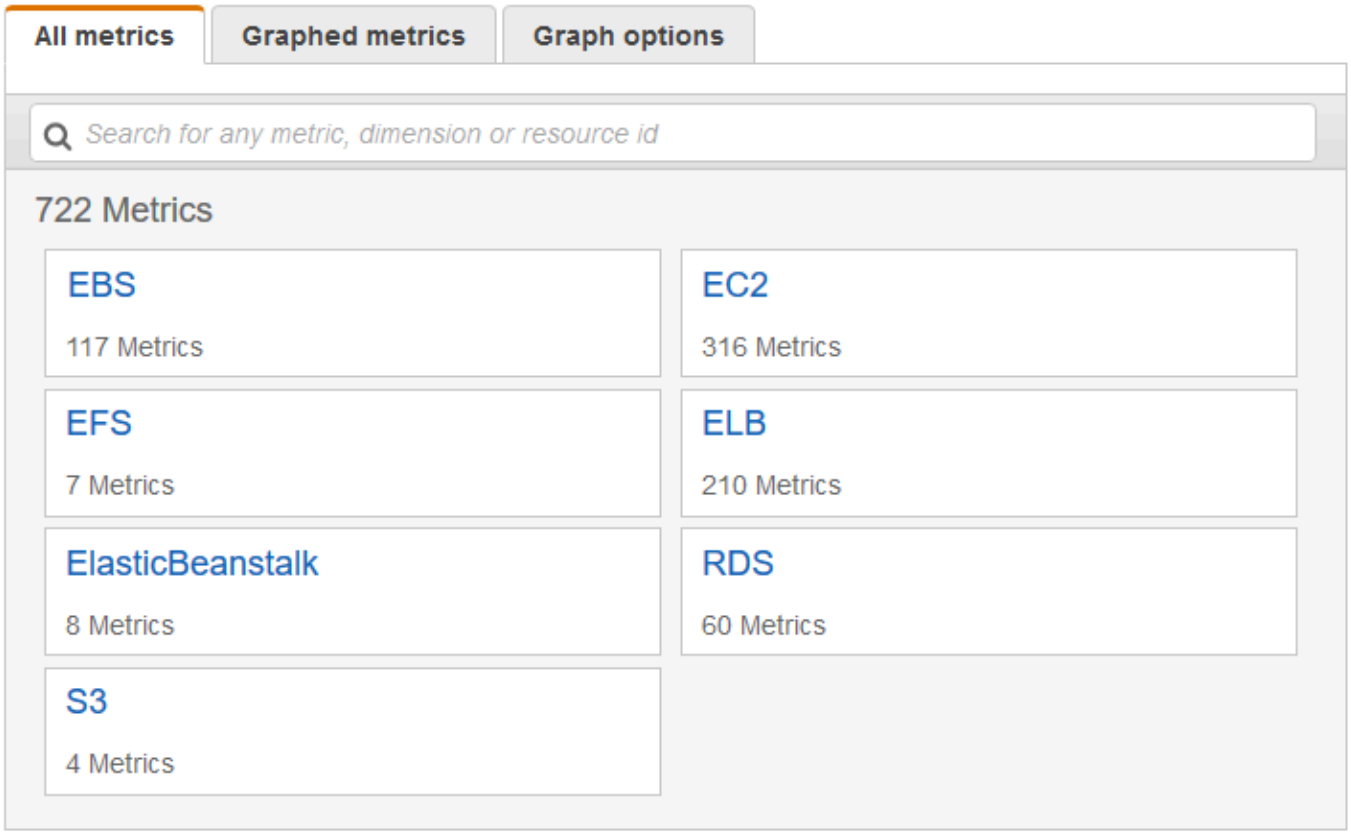
L'esempio seguente illustra come determinare l'utilizzo della CPU massimo di un'istanza EC2 specifica.

Requisiti

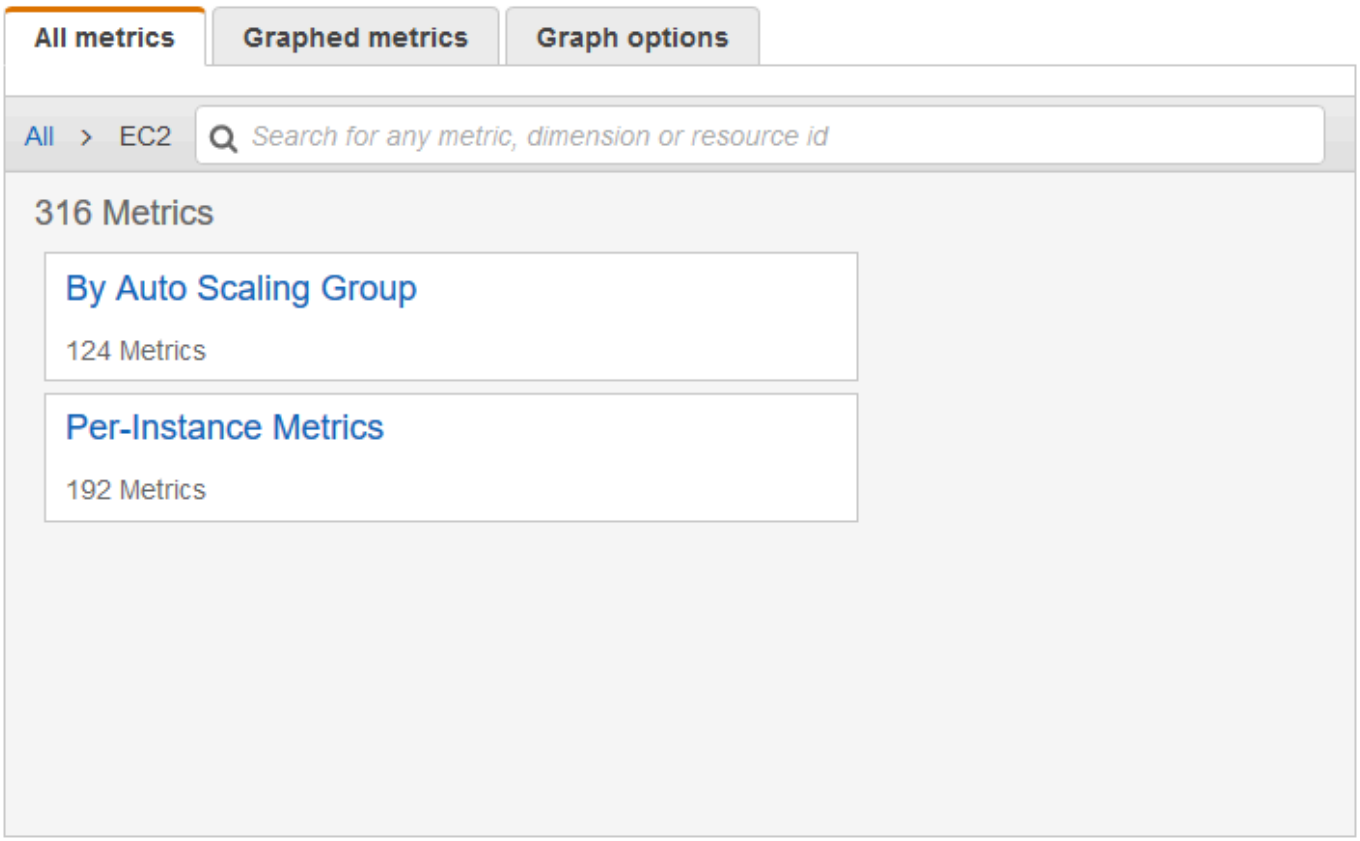
- Devi disporre dell'ID dell'istanza. Puoi ottenere l'ID dell'istanza tramite la console Amazon EC2 o tramite il comando describe-instances.
- Per impostazione predefinita, il monitoraggio base è abilitato, ma puoi tuttavia abilitare il monitoraggio dettagliato. Per ulteriori informazioni, consulta Abilitazione o disabilitazione del monitoraggio dettagliato delle istanze nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Visualizzazione dell'utilizzo medio della CPU di un'istanza specifica tramite la console

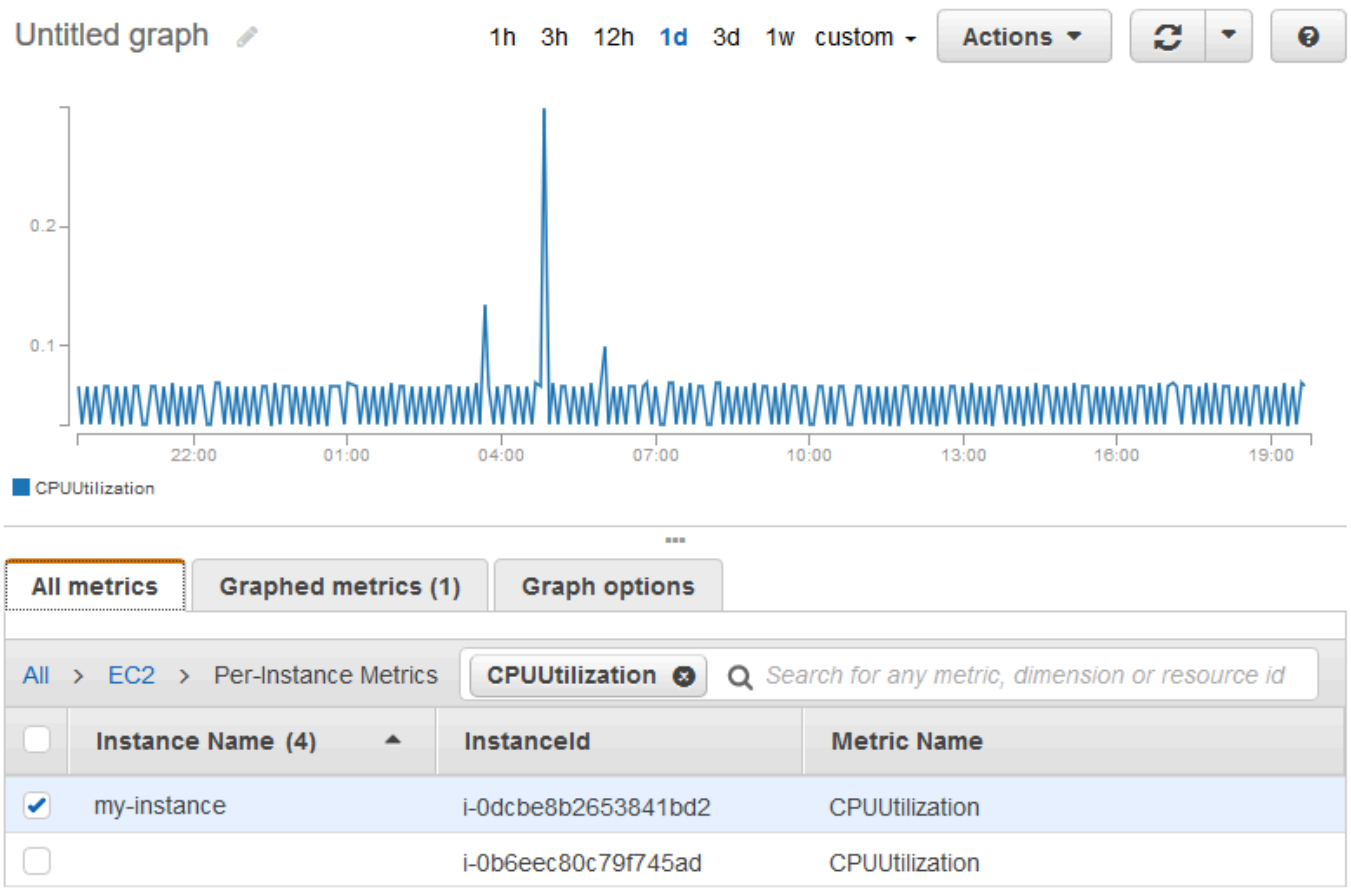
1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Seleziona lo spazio dei nomi parametro EC2.



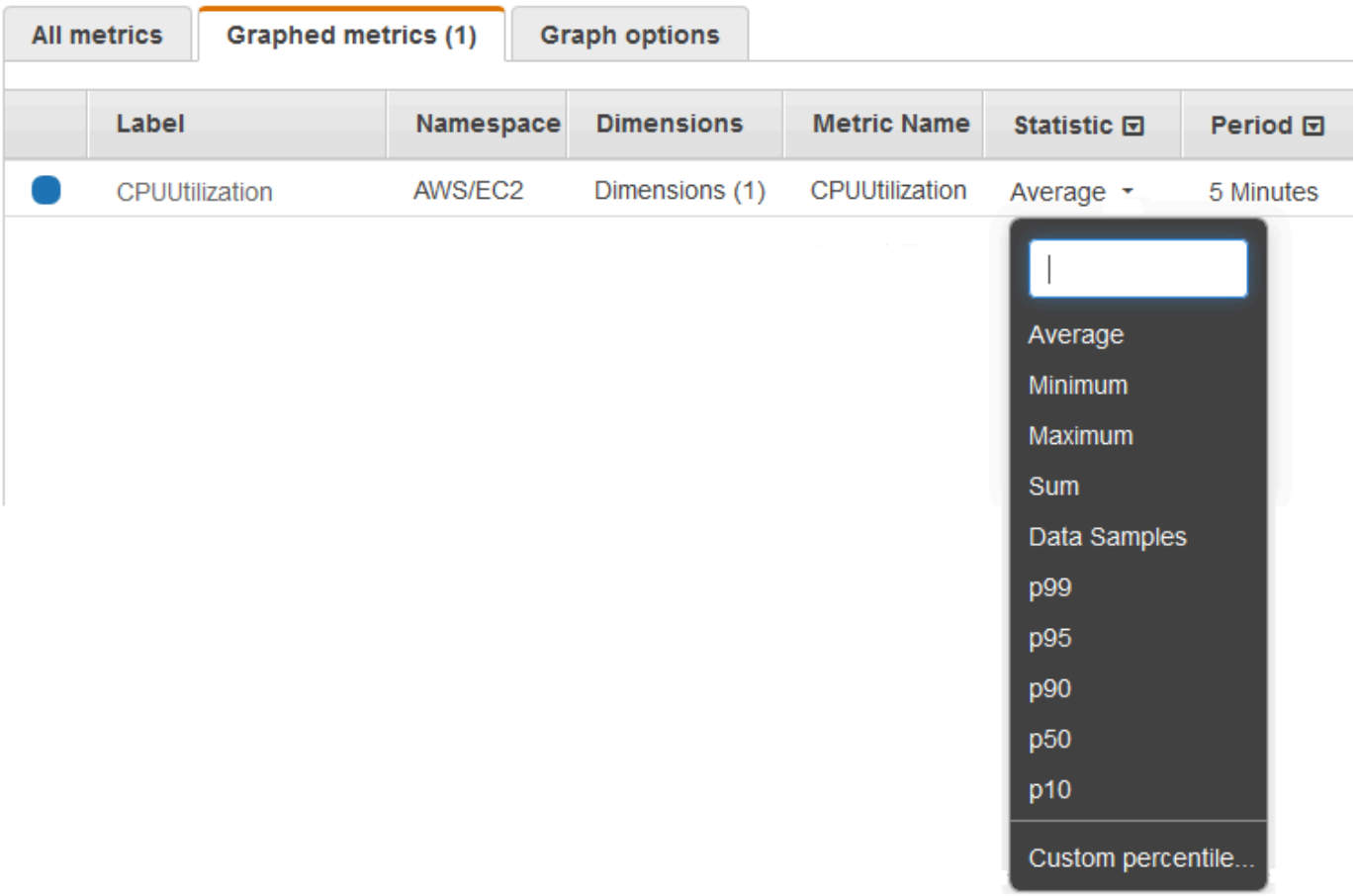
4. Seleziona la dimensione Per-Instance Metrics (Parametri per istanza).



5. Nel campo di ricerca digitare **`CPUUtilization`** e premere Invio. Seleziona la riga dell'istanza specifica, che visualizza un grafico del parametro `CPUUtilization` dell'istanza. Per modificare il nome del grafico, seleziona l'icona a forma di matita. Per modificare l'intervallo di tempo, seleziona uno dei valori predefiniti o scegli custom (personalizzato).



6. Per modificare la statistica, seleziona la scheda Graphed metrics (Parametri nel grafico). Scegli l'intestazione di colonna o un valore singolo, quindi seleziona una delle statistiche o dei percentili predefiniti oppure specifica un percentile personalizzato (ad esempio, **`p99.999`**).



7. Per modificare il periodo, seleziona la scheda Graphed metrics (Parametri nel grafico). Scegli l'intestazione di colonna o un valore singolo, quindi scegli un valore diverso.

Per ottenere l'utilizzo della CPU per istanza EC2, utilizza AWS CLI

Usa il get-metric-statisticscomando come segue per ottenere la `CPUUtilization` metrica per l'istanza specificata.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=InstanceId,Value=i-1234567890abcdef0 --statistics Maximum \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00 --period 360
```

Le statistiche restituite sono valori di 6 minuti per l'intervallo di tempo di 24 ore richiesto. Ogni valore rappresenta la percentuale massima di utilizzo della CPU dell'istanza specificata per un determinato periodo di tempo di 6 minuti. I punti dati non vengono restituiti in ordine cronologico. Di seguito è riportato l'inizio dell'output di esempio (l'output completo include i punti dati per ogni 6 minuti del periodo di 24 ore).

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Aggregazione di statistiche tra risorse

È possibile aggregare le metriche relative AWS alle risorse su più risorse. I parametri sono completamente separati tra le regioni, ma puoi utilizzare la matematica dei parametri per aggregare parametri simili tra più regioni. Per ulteriori informazioni, consulta Utilizzare la matematica dei parametri.

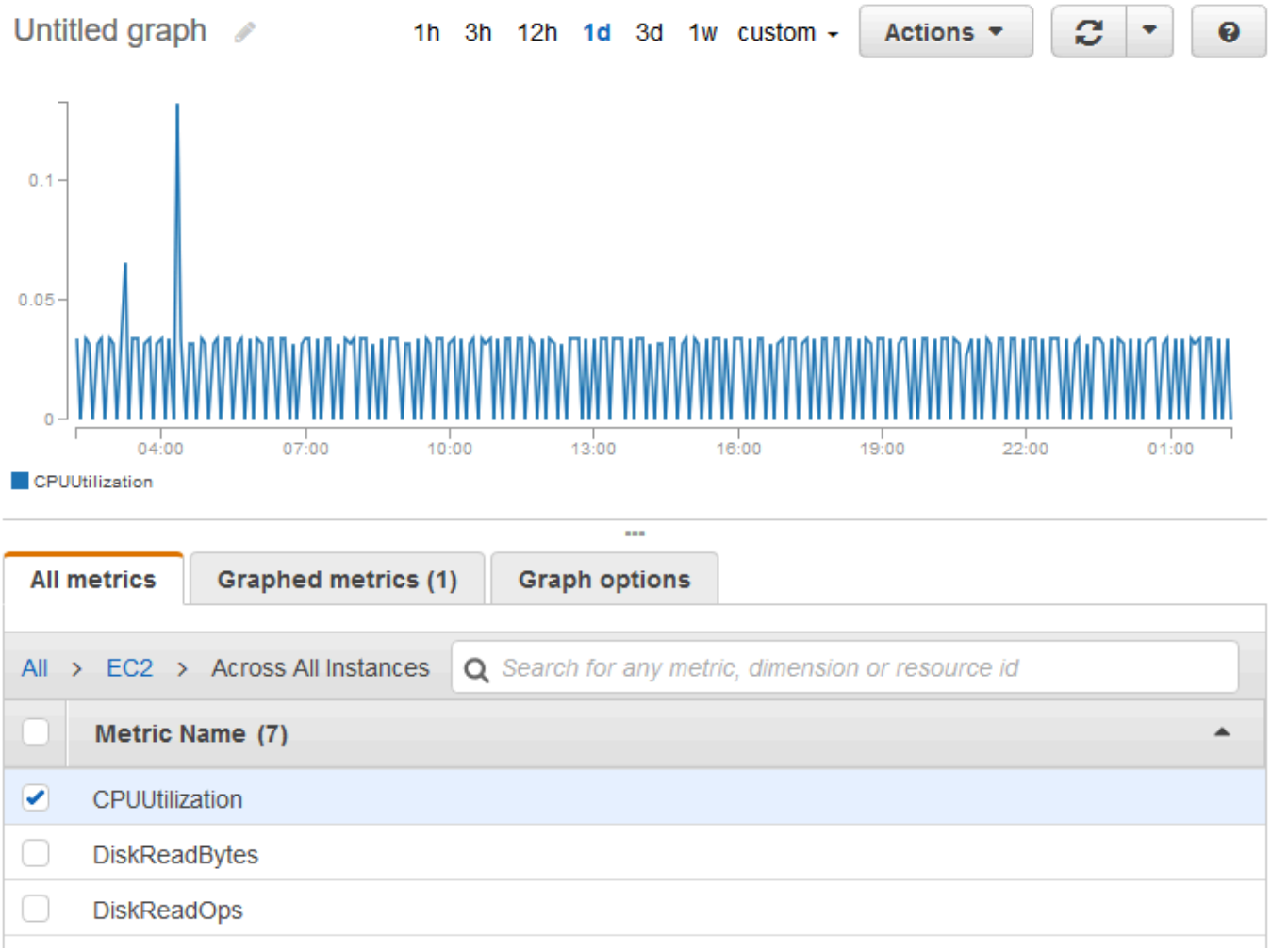
Ad esempio, puoi aggregare le statistiche per le istanze EC2 aventi il monitoraggio dettagliato abilitato. Le istanze che utilizzano il monitoraggio base non sono incluse. Pertanto, è necessario abilitare il monitoraggio dettagliato (a un costo aggiuntivo), che fornisce dati in periodi di 1 minuto. Per ulteriori informazioni, consulta Abilitazione o disabilitazione del monitoraggio dettagliato delle istanze nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Questo esempio illustra come ottenere l'utilizzo medio della CPU delle istanze EC2. Poiché non è specificata alcuna dimensione, CloudWatch restituisce le statistiche per tutte le dimensioni nel namespace. `AWS/EC2` Per ottenere le statistiche di altri parametri, consulta AWS servizi che pubblicano CloudWatch metriche.

> **Important**
>
> Questa tecnica per recuperare tutte le dimensioni in un AWS namespace non funziona per gli spazi dei nomi personalizzati in cui pubblichi. CloudWatch Con gli spazi dei nomi personalizzati, è necessario specificare il set completo delle dimensioni associate a un determinato punto dati per recuperare le statistiche comprendenti il punto dati.

Visualizzazione dell'utilizzo medio della CPU delle istanze EC2

1. Apri CloudWatch la console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Seleziona lo spazio dei nomi EC2, quindi seleziona Across All Instances (Per tutte le istanze).
4. Seleziona la riga contenente `CPUUtilization`, che visualizza un grafico del parametro per tutte le istanze EC2. Per modificare il nome del grafico, seleziona l'icona a forma di matita. Per modificare l'intervallo di tempo, seleziona uno dei valori predefiniti o scegli custom (personalizzato).



5. Per modificare la statistica, seleziona la scheda Graphed metrics (Parametri nel grafico). Scegli l'intestazione di colonna o un valore singolo, quindi seleziona una delle statistiche o dei percentili predefiniti oppure specifica un percentile personalizzato (ad esempio, **`p95.45`**).
6. Per modificare il periodo, seleziona la scheda Graphed metrics (Parametri nel grafico). Seleziona l'intestazione di colonna o un singolo valore, quindi scegli un valore diverso.

Per ottenere un utilizzo medio della CPU su tutte le istanze EC2, utilizza AWS CLI

Utilizza il comando get-metric-statistics come segue:

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00 --period 3600
```

Di seguito è riportato un output di esempio:

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2016-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```


Aggregazione di statistiche per gruppo Auto Scaling

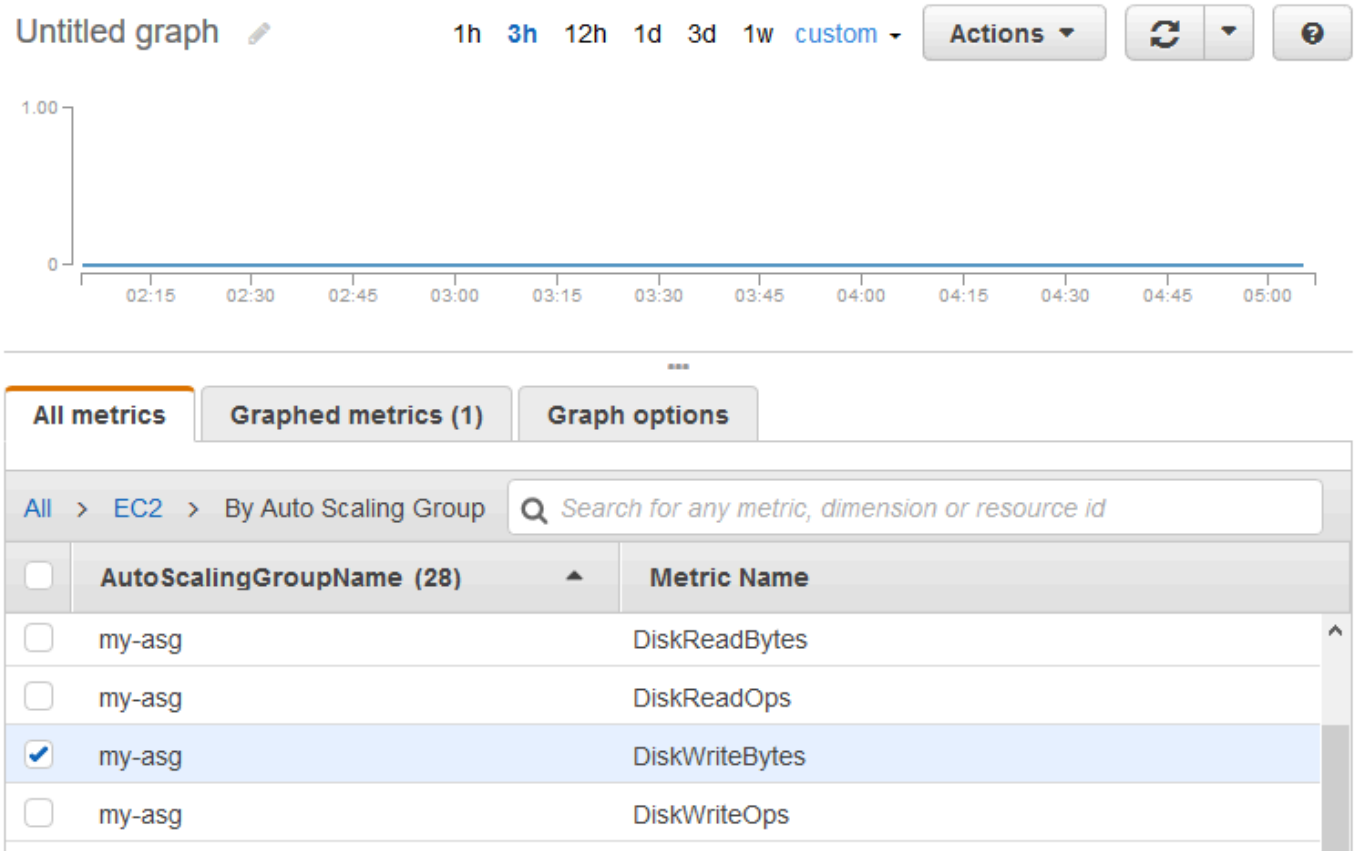
Puoi aggregare le statistiche per le istanze EC2 in un gruppo di Auto Scaling. Le metriche sono completamente separate tra le regioni, ma puoi utilizzare la matematica CloudWatch metrica per aggregare e trasformare le metriche di più regioni. Inoltre puoi utilizzare il pannello di controllo tra account per eseguire calcoli matematici sui parametri di account diversi.

Questo esempio illustra come ottenere il numero totale di byte scritti su disco per un gruppo Auto Scaling. Il totale viene calcolato per periodi di 1 minuto per un intervallo di 24 ore all'interno di tutte le istanze EC2 nel gruppo Auto Scaling specificato.

DiskWriteBytes Per visualizzare le istanze in un gruppo Auto Scaling utilizzando la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.

3. Seleziona lo spazio dei nomi EC2, quindi seleziona By Auto Scaling Group (Per gruppo Auto Scaling).
4. Seleziona la riga per la DiskWriteBytesmetrica e il gruppo Auto Scaling specifico, che visualizza un grafico per la metrica per le istanze nel gruppo Auto Scaling. Per modificare il nome del grafico, seleziona l'icona a forma di matita. Per modificare l'intervallo di tempo, seleziona uno dei valori predefiniti o scegli custom (personalizzato).



5. Per modificare la statistica, seleziona la scheda Graphed metrics (Parametri nel grafico). Scegli l'intestazione di colonna o un valore singolo, quindi seleziona una delle statistiche o dei percentili predefiniti oppure specifica un percentile personalizzato (ad esempio, **`p95.45`**).
6. Per modificare il periodo, seleziona la scheda Graphed metrics (Parametri nel grafico). Seleziona l'intestazione di colonna o un singolo valore, quindi scegli un valore diverso.

DiskWriteBytes Per ottenere le istanze in un gruppo Auto Scaling utilizzando il AWS CLI

Utilizza il comando get-metric-statistics come riportato di seguito.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
--dimensions Name=AutoScalingGroupName,Value=my-asg --statistics "Sum" "SampleCount" \
```

```
--start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00 --period 360
```

Di seguito è riportato un output di esempio.

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2016-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2016-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

Aggregazione di statistiche per Amazon Machine Image (AMI)

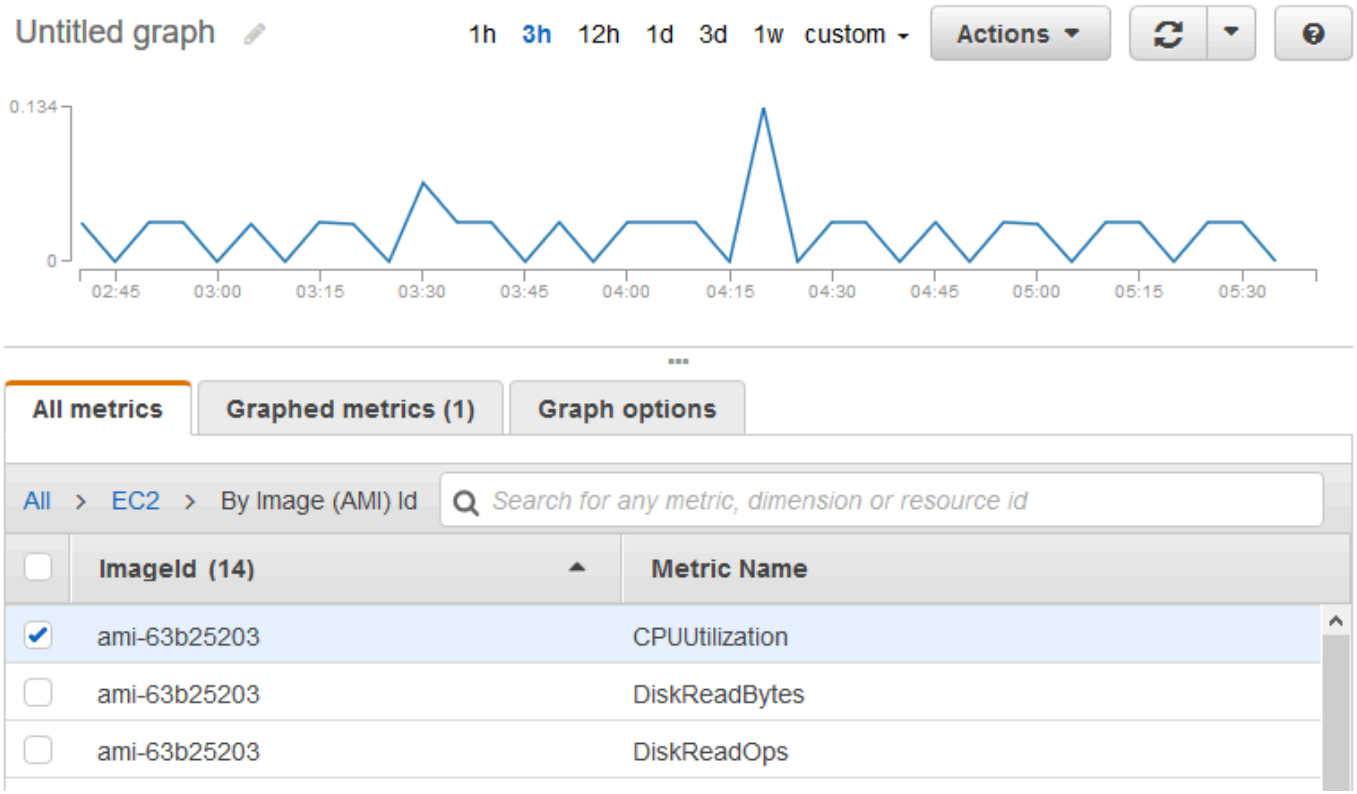
Puoi aggregare le statistiche per le istanze EC2 aventi il monitoraggio dettagliato abilitato. Le istanze che utilizzano il monitoraggio base non sono incluse. Per ulteriori informazioni, consulta Abilitazione o disabilitazione del monitoraggio dettagliato delle istanze nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Questo esempio illustra come determinare l'utilizzo medio della CPU per tutte le istanze che utilizzano l'AMI specificata. La media supera intervalli di tempo di 60 secondi per un periodo di un giorno.

Per visualizzare l'utilizzo medio della CPU per AMI tramite la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Seleziona lo spazio dei nomi EC2, quindi seleziona By Image (AMI) Id (Per ID immagine (AMI)).
4. Seleziona la riga del parametro `CPUUtilization` e l'AMI specifica, che visualizza un grafico del parametro per l'AMI specificata. Per modificare il nome del grafico, seleziona l'icona a forma

di matita. Per modificare l'intervallo di tempo, seleziona uno dei valori predefiniti o scegli custom (personalizzato).



5. Per modificare la statistica, seleziona la scheda Graphed metrics (Parametri nel grafico). Scegli l'intestazione di colonna o un valore singolo, quindi seleziona una delle statistiche o dei percentili predefiniti oppure specifica un percentile personalizzato (ad esempio, **`p95.45`**).
6. Per modificare il periodo, seleziona la scheda Graphed metrics (Parametri nel grafico). Seleziona l'intestazione di colonna o un singolo valore, quindi scegli un valore diverso.

Per ottenere l'utilizzo medio della CPU da parte dell'AMI utilizzando il AWS CLI

Utilizza il comando get-metric-statistics come riportato di seguito.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=ImageId,Value=ami-3c47a355 --statistics Average \
--start-time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00 --period 3600
```

L'operazione restituisce statistiche che sono valori di un'ora per l'intervallo di un giorno. Ogni valore rappresenta una percentuale di utilizzo medio della CPU per le istanze EC2 che eseguono l'AMI specificata. Di seguito è riportato un output di esempio.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Pubblicare i parametri personalizzati di

Puoi pubblicare le tue metriche CloudWatch utilizzando AWS CLI o un'API. Puoi visualizzare grafici statistici delle metriche pubblicate con. AWS Management Console

CloudWatch archivia i dati relativi a una metrica come una serie di punti dati. Ogni punto dati viene associato a un timestamp. Puoi inoltre pubblicare un set aggregato di punti di dati denominato set di statistiche.

Argomenti

Parametri ad alta risoluzione

Ogni parametro appartiene a una delle seguenti categorie:

- Risoluzione standard, con dati aventi una granularità di un minuto
- Alta risoluzione, con dati aventi una granularità di un secondo

Per impostazione predefinita, le metriche prodotte dai AWS servizi hanno una risoluzione standard. Quando pubblichi un parametro personalizzato, puoi definirlo sia come risoluzione standard che come alta risoluzione. Quando pubblichi una metrica ad alta risoluzione, la CloudWatch archivia con una risoluzione di 1 secondo e puoi leggerla e recuperarla con un periodo di 1 secondo, 5 secondi, 10 secondi, 30 secondi o qualsiasi multiplo di 60 secondi.

I parametri ad alta risoluzione ti offrono un'analisi più immediata sull'attività inferiore al minuto dell'applicazione. Tieni presente che ogni chiamata `PutMetricData` per un parametro personalizzato viene addebitata, quindi frequenti chiamate a `PutMetricData` su un parametro ad alta risoluzione potrebbero portare a costi più elevati. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Se imposti un allarme su un parametro ad alta risoluzione, puoi specificare un allarme ad alta risoluzione con un periodo di 10 secondi o 30 secondi, oppure puoi impostare un allarme regolare con un periodo di qualsiasi multiplo di più di 60 secondi. Viene addebitato un costo maggiore per gli allarmi ad alta risoluzione con un periodo di 10 o 30 secondi.

Utilizzo delle dimensioni

Nei parametri personalizzati, il parametro `--dimensions` è comune. Una dimensione chiarisce ulteriormente le caratteristiche del parametro e i dati archiviati. Puoi avere un massimo di 30 dimensioni assegnate a un parametro, ognuna delle quali è definita da una coppia formata da nome e valore.

Il modo in cui si specifica una dimensione è diverso quando utilizzi comandi differenti. Con put-metric-data, specifichi ogni dimensione come *`MyName`*= *`MyValue`*e con get-metric-statisticso put-metric-alarmusi il formato `Name`= *`MyName`*, `Value`= *`MyValue`*. Ad esempio, il seguente comando consente di pubblicare un parametro `Buffers` con due dimensioni denominate `InstanceId` e `InstanceType`.

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace MyNameSpace --unit
 Bytes --value 231434333 --dimensions InstanceId=1-23456789,InstanceType=m1.small
```

Questo comando recupera le statistiche per quello stesso parametro. Separa con virgole le parti di Nome e Valore di una singola dimensione, ma utilizza uno spazio tra una dimensione e quella successiva se disponi di più dimensioni.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace --
dimensions Name=InstanceId,Value=1-23456789 Name=InstanceType,Value=m1.small --start-
time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average --period
 60
```

Se una singola metrica include più dimensioni, è necessario specificare un valore per ogni dimensione definita quando si utilizza get-metric-statistics. Ad esempio, la metrica di Amazon S3 `BucketSizeBytes` include le dimensioni `BucketName` e`StorageType`, pertanto, è necessario specificare entrambe le dimensioni con. get-metric-statistics

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --start-time
 2017-01-23T14:23:00Z --end-time 2017-01-26T19:30:00Z --period 3600 --namespace
 AWS/S3 --statistics Maximum --dimensions Name=BucketName,Value=MyBucketName
 Name=StorageType,Value=StandardStorage --output table
```

Per visualizzare le dimensioni definite per un parametro, è disponibile il comando list-metrics.

Pubblicazione di singoli punti dati

Per pubblicare un singolo punto dati per una metrica nuova o esistente, usa il put-metric-datacomando con un valore e un timestamp. Ad esempio, ciascuna delle seguenti operazioni pubblica un punto dati.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 2 --timestamp 2016-10-20T12:00:00.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 4 --timestamp 2016-10-20T12:00:01.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 5 --timestamp 2016-10-20T12:00:02.000Z
```

Se chiami questo comando con un nuovo nome di metrica, CloudWatch crea una metrica per te. Altrimenti, CloudWatch associa i tuoi dati alla metrica esistente che hai specificato.

> **Note**
>
> Quando crei una metrica, possono essere necessari fino a 2 minuti prima di poter recuperare le statistiche per la nuova metrica utilizzando il comando. get-metric-statistics Tuttavia, possono essere necessari fino a 15 minuti prima che il nuovo parametro venga visualizzato nell'elenco di quelli recuperati tramite il comando list-metrics.

Sebbene sia possibile pubblicare punti dati con timestamp granulari fino a un millesimo di secondo, CloudWatch aggrega i dati con una granularità minima di 1 secondo. CloudWatch registra la media (somma di tutti gli elementi divisa per il numero di elementi) dei valori ricevuti per ogni periodo, nonché il numero di campioni, il valore massimo e il valore minimo per lo stesso periodo di tempo. Ad esempio, il parametro `PageViewCount` degli esempi precedenti contiene tre punti di dati con timestamp distanti di pochi secondi. Se il periodo è impostato su 1 minuto, CloudWatch aggrega i tre punti dati perché hanno tutti un timestamp entro un periodo di 1 minuto.

Puoi utilizzare il comando get-metric-statistics per recuperare le statistiche in base ai punti dati pubblicati.

```
aws cloudwatch get-metric-statistics --namespace MyService --metric-name PageViewCount
 \
--statistics "Sum" "Maximum" "Minimum" "Average" "SampleCount" \
--start-time 2016-10-20T12:00:00.000Z --end-time 2016-10-20T12:05:00.000Z --period 60
```

Di seguito è riportato un output di esempio.

```
{
    "Datapoints": [
        {
            "SampleCount": 3.0,
            "Timestamp": "2016-10-20T12:00:00Z",
            "Average": 3.6666666666666665,
            "Maximum": 5.0,
            "Minimum": 2.0,
            "Sum": 11.0,
            "Unit": "None"
        }
    ],
    "Label": "PageViewCount"
}
```

Pubblicazione di set di statistiche

Puoi aggregare i tuoi dati prima di pubblicarli su. CloudWatch Quando sono presenti più punti dati al minuto, l'aggregazione dei dati riduce al minimo il numero di chiamate a put-metric-data. Ad esempio, invece di chiamare più volte put-metric-data per tre punti dati a distanza di 3 secondi l'uno dall'altro, è possibile aggregare i dati in un set di statistiche da pubblicare con un'unica chiamata, tramite il parametro `--statistic-values`.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService
 --statistic-values Sum=11,Minimum=2,Maximum=5,SampleCount=3 --
timestamp 2016-10-14T12:00:00.000Z
```

CloudWatch necessita di punti dati grezzi per calcolare i percentili. Se pubblichi dati utilizzando un set di statistiche, invece, non potrai recuperarne le relative statistiche dei percentili, a meno che non si verifichi una delle seguenti condizioni:

- Il `SampleCount` del set di statistiche è 1
- I valori `Minimum` e `Maximum` del set di statistiche sono uguali

Pubblicazione del valore zero

Quando i tuoi dati sono più sporadici e sono presenti periodi senza dati associati, puoi scegliere di pubblicare il valore zero (`0`) per tale periodo oppure nessun valore. Se utilizzi chiamate periodiche a `PutMetricData` per monitorare lo stato delle applicazioni, potresti voler pubblicare zero invece di nessun valore. Ad esempio, puoi impostare un CloudWatch allarme per avvisarti se l'applicazione non riesce a pubblicare le metriche ogni cinque minuti. Desideri che tale applicazione pubblichi valori zero per i periodi senza dati associati.

Puoi inoltre pubblicare valori zero se intendi monitorare il numero totale di punti di dati o se desideri che le statistiche di tipo minima e media includano i punti di dati con il valore 0.

Interrompi i parametri di pubblicazione

Per interrompere la pubblicazione di metriche personalizzate su CloudWatch, modifica il codice dell'applicazione o del servizio in modo che smetta di utilizzarle. PutMetricData CloudWatch non estrae metriche dalle applicazioni, riceve solo ciò che gli viene inviato, quindi per interrompere la pubblicazione delle metriche è necessario interromperle alla fonte.

Utilizzo degli CloudWatch allarmi Amazon

Puoi creare allarmi metrici e compositi in Amazon. CloudWatch

- Un allarme metrico controlla una singola CloudWatch metrica o il risultato di un'espressione matematica basata su metriche. CloudWatch L'allarme esegue una o più operazioni basate sul valore del parametro o espressione relativa a una soglia su un certo numero di periodi. L'azione può essere l'invio di una notifica a un argomento di Amazon SNS, l'esecuzione di un'azione Amazon EC2 o un'azione Amazon EC2 Auto Scaling o la creazione di un incidente o in Systems Manager. OpsItem
- Un allarme composito include un'espressione di regola che tiene conto degli stati di avviso di altri avvisi creati. L'allarme composito entra in stato ALARM solo se tutte le condizioni della regola sono soddisfatte. Gli allarmi specificati nell'espressione di regola di un allarme composito possono includere allarmi di parametri e altri allarmi compositi.

  L'utilizzo di allarmi compositi consente di ridurre il rumore dell'allarme. È possibile creare più allarmi di parametri e anche creare un allarme composito e impostare gli avvisi solo per l'allarme composito. Ad esempio, un composito potrebbe entrare in stato ALARM solo quando tutti gli allarmi dei parametri sottostanti sono in stato ALARM.

  Gli allarmi compositi possono inviare notifiche Amazon SNS quando cambiano stato e possono creare Systems OpsItems Manager o incidenti quando entrano nello stato ALARM, ma non possono eseguire azioni EC2 o azioni di Auto Scaling.

> **Note**
>
> Puoi creare tutti gli allarmi che desideri nel tuo account. AWS

Puoi aggiungere allarmi alle dashboard, in modo da monitorare e ricevere avvisi sulle tue AWS risorse e applicazioni in più regioni. Dopo avere aggiunto un allarme a un pannello di controllo, l'allarme diventa grigio quando è nello stato `INSUFFICIENT_DATA` e rosso quando è nello stato `ALARM`. L'allarme non ha alcun colore quando è nello stato `OK`.

Puoi anche aggiungere ai preferiti gli allarmi visitati di recente dall'opzione Preferiti e recenti nel pannello di navigazione della console. CloudWatch L'opzione Favorites and recents (Preferiti e recenti) contiene colonne per gli allarmi contrassegnati come preferiti e quelli consultati di recente.

Un allarme richiama le operazioni solo quando lo stato dell'allarme cambia. L'eccezione è per gli allarmi con operazioni Auto Scaling. Per operazioni Auto Scaling, l'allarme continua a richiamare l'operazione una volta per ogni minuto durante il quale rimane nel nuovo stato.

Un allarme può osservare una metrica nello stesso account. Se hai abilitato la funzionalità tra account nella tua CloudWatch console, puoi anche creare allarmi che controllano le metriche di altri account. AWS La creazione di allarmi compositi tra più account non è supportata. È supportata la creazione di allarmi tra più account che utilizzano espressioni matematiche, ad eccezione del fatto che le funzioni `ANOMALY_DETECTION_BAND`, `INSIGHT_RULE` e `SERVICE_QUOTA` non sono supportate per gli allarmi tra più account.

> **Note**
>
> CloudWatch non verifica o convalida le azioni specificate, né rileva errori di Amazon EC2 Auto Scaling o Amazon SNS derivanti dal tentativo di richiamare azioni inesistenti. Assicurati che le tue operazioni relative agli allarmi esistano.

Stati degli allarmi di parametri

Un allarme di parametri può trovarsi nei possibili stati elencati di seguito:

- `OK` - Il parametro o espressione rientra nella soglia definita.
- `ALARM` - Il parametro o espressione non rientra nella soglia definita.
- `INSUFFICIENT_DATA` - L'allarme è stato appena attivato, il parametro non è disponibile o la quantità di dati non è sufficiente affinché il parametro determini lo stato dell'allarme.

Valutazione di un allarme

Quando crei un allarme, specifichi tre impostazioni per consentire di valutare quando CloudWatch modificare lo stato dell'allarme:

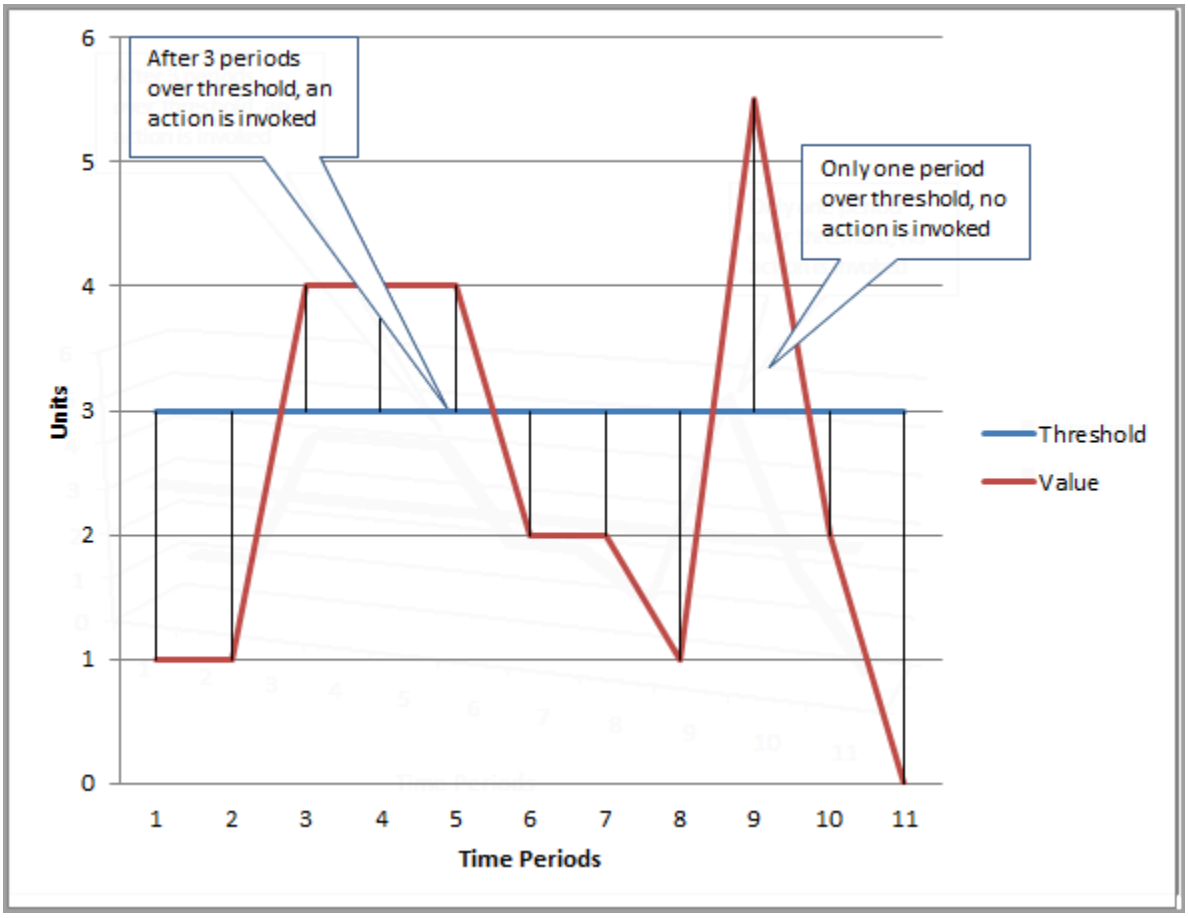
- Periodo è l'intervallo di tempo su cui valutare il parametro o l'espressione e creare ogni singolo punto di dati per un allarme. Viene espresso in secondi.
- Evaluation Periods (Periodi di valutazione) è il numero di periodi più recenti, o punti di dati, per valutare quando stabilire lo stato di allarme.

- Datapoints to Alarm (Punti di dati all'allarme) è il numero punti di dati all'interno dei periodi di valutazione che devono essere violati per fare in modo che l'allarme sia nello stato `ALARM`. I punti dati oggetto della violazione non devono essere consecutivi, ma devono solo essere tutti all'interno dell'ultimo numero di punti dati pari all'Evaluation Period (Periodo di valutazione).

Per un periodo di almeno un minuto, un allarme viene valutato ogni minuto e la valutazione si basa sulla finestra temporale definita da Periodo e Periodi di valutazione. Ad esempio, se Periodo è di 5 minuti (300 secondi) e Periodi di valutazione è 1, alla fine del minuto 5 l'allarme viene valutato in base ai dati compresi tra 1 e 5 minuti. Quindi, alla fine del minuto 6, l'allarme viene valutato in base ai dati dal secondo al sesto minuto.

Se il periodo di allarme è di 10 secondi o 30 secondi, l'allarme viene valutato ogni 10 secondi.

Nella figura seguente, la soglia di allarme per un allarme dei parametri è impostata su tre unità. Sia l'Evaluation Period (Periodo di valutazione) che Datapoints to Alarm (Punti di dati all'allarme) sono 3. Quando tutti i punti dati esistenti nei tre periodi consecutivi più recenti sono sopra la soglia, l'allarme passa allo stato `ALARM`. Nella figura, questo accade dal terzo al quinto periodo di tempo. Al periodo sei, il valore scende sotto la soglia, perciò uno dei periodi valutati non effettua una violazione e lo stato dell'allarme cambia in `OK`. Durante il nono periodo di tempo, la soglia viene nuovamente superata, ma per un solo periodo. Di conseguenza, lo stato dell'allarme rimane `OK`.



Quando si configura Evaluation Periods (Periodi di valutazione) e Datapoints to Alarm (Punti dati all'allarme) come valori diversi, si imposta un allarme "M su N". Datapoints to Alarm (Punti di dati all'allarme) è ("M") e Evaluation Periods (Periodi di valutazione) è ("N"). L'intervallo di valutazione è il numero di punti dati moltiplicato per il periodo. Ad esempio, se configuri 4 punti dati su 5 con un periodo di 1 minuto, l'intervallo di valutazione è di 5 minuti. Se configuri 3 punti dati su 3 con un periodo di 10 minuti, l'intervallo di valutazione è di 30 minuti.

> **Note**
>
> Se i punti dati mancano subito dopo la creazione di un allarme e la metrica veniva riportata CloudWatch prima della creazione dell'allarme, CloudWatch recupera i punti dati più recenti precedenti alla creazione dell'allarme durante la valutazione dell'allarme.

Operazioni per gli allarmi

È possibile specificare le operazioni intraprese da un allarme quando cambia stato tra gli stati OK, ALARM e INSUFFICIENT_DATA.

È possibile impostare la maggior parte delle operazioni per la transizione in ciascuno dei tre stati. Ad eccezione delle operazioni di dimensionamento automatico, le operazioni si verificano solo nelle transizioni di stato e non vengono più eseguite se la condizione persiste per ore o giorni. È possibile sfruttare il fatto che sono consentite più operazioni per un allarme per inviare un'e-mail quando viene superata una soglia e poi un'altra quando la condizione di violazione termina. Ciò consente di verificare che le operazioni di dimensionamento o ripristino vengano attivate quando previsto e funzionino come desiderato.

Le seguenti sono supportate come operazioni di allarme.

- Inviare una notifica a uno o più abbonati tramite un argomento Amazon Simple Notification Service. Gli abbonati possono essere sia applicazioni che persone. Per informazioni complete su Amazon SNS, consulta Che cos'è Amazon SNS?.
- Richiamare una funzione Lambda. Questo è il modo più semplice per automatizzare le azioni personalizzate relative alle dello stato degli allarmi.
- Gli allarmi basati su parametri EC2 possono eseguire operazioni EC2, ad esempio l'arresto, l'interruzione, il riavvio o il ripristino di un'istanza EC2. Per ulteriori informazioni, consulta Creazione di allarmi per arrestare, terminare, riavviare o recuperare un'istanza EC2.
- Gli allarmi possono anche eseguire operazioni per dimensionare un gruppo Auto Scaling. Per ulteriori informazioni, consulta Dimensionamento per fasi e policy di dimensionamento semplice per Amazon EC2 Auto Scaling.
- Gli allarmi possono essere creati OpsItems in Systems Manager Ops Center o creare incidenti in AWS Systems Manager Incident Manager. Queste operazioni vengono eseguite solo quando l'allarme entra in stato ALARM. Per ulteriori informazioni, vedere Configurazione per la creazione CloudWatch a OpsItems partire da allarmi e Creazione di incidenti.

Operazioni allarme Lambda

CloudWatch alarms garantisce un'invocazione asincrona della funzione Lambda per un determinato cambio di stato, tranne nei seguenti casi:

- Quando la funzione non esiste.

- When non CloudWatch è autorizzato a richiamare la funzione Lambda.

Se non CloudWatch riesce a raggiungere il servizio Lambda o il messaggio viene rifiutato per un altro motivo, CloudWatch riprova finché la chiamata non ha esito positivo. Lambda mette in coda il messaggio e gestisce i tentativi di esecuzione. Per ulteriori informazioni su questo modello di esecuzione, incluse informazioni su come Lambda gestisce gli errori, consulta Asynchronous invocation nella Developer Guide. AWS Lambda

È possibile richiamare una funzione Lambda nello stesso account o in AWS altri account.

Quando specifichi un allarme per richiamare una funzione Lambda come operazione di allarme, puoi scegliere di specificare il nome della funzione, l'alias della funzione o una versione specifica di una funzione.

Quando si specifica una funzione Lambda come azione di allarme, è necessario creare una politica delle risorse per la funzione per consentire al responsabile del CloudWatch servizio di richiamare la funzione.

Un modo per farlo è utilizzare AWS CLI, come nell'esempio seguente:

```
aws lambda add-permission \
--function-name my-function-name \
--statement-id AlarmAction \
--action 'lambda:InvokeFunction' \
--principal lambda.alarms.cloudwatch.amazonaws.com \
--source-account 111122223333 \
--source-arn arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name
```

In alternativa, puoi creare una policy simile a uno degli esempi riportati di seguito e assegnarla alla funzione.

L'esempio seguente specifica l'account in cui si trova l'allarme, in modo che solo gli allarmi in quell'account (111122223333) possano richiamare la funzione.

```
{
    "Version": "2012-10-17",
    "Id": "default",
    "Statement": [{
        "Sid": "AlarmAction",
        "Effect": "Allow",
        "Principal": {
```

```
            "Service": "lambda.alarms.cloudwatch.amazonaws.com"
        },
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "111122223333"
            }
        }
    }]
}
```

L'esempio seguente ha un ambito più limitato e consente solo all'allarme specificato nell'account specificato di richiamare la funzione.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "AlarmAction",
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.alarms.cloudwatch.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "111122223333",
          "AWS:SourceArn": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
      }
    }]
}
```

Non è consigliabile creare una policy che non specifichi un account di origine, poiché tali policy sono vulnerabili a problemi di "confused deputy".

Oggetto evento inviato da CloudWatch Lambda

Quando configuri una funzione Lambda come azione di allarme, CloudWatch fornisce un payload JSON alla funzione Lambda quando richiama la funzione. Questo payload JSON funge da oggetto

evento per la funzione. Puoi estrarre dati da questo oggetto JSON e utilizzarli nella tua funzione. Di seguito è riportato un esempio di oggetto evento da un allarme del parametro.


```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:444455556666:alarm:lambda-demo-metric-
alarm',
  'accountId': '444455556666',
  'time': '2023-08-04T12:36:15.490+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'lambda-demo-metric-alarm',
    'state': {
      'value': 'ALARM',
      'reason': 'test',
      'timestamp': '2023-08-04T12:36:15.490+0000'
    },
    'previousState': {
      'value': 'INSUFFICIENT_DATA',
      'reason': 'Insufficient Data: 5 datapoints were unknown.',
      'reasonData':
 '{"version":"1.0","queryDate":"2023-08-04T12:31:29.591+0000","statistic":"Average","period":60
[],"threshold":5.0,"evaluatedDatapoints":[{"timestamp":"2023-08-04T12:30:00.000+0000"},
{"timestamp":"2023-08-04T12:29:00.000+0000"},
{"timestamp":"2023-08-04T12:28:00.000+0000"},
{"timestamp":"2023-08-04T12:27:00.000+0000"},
{"timestamp":"2023-08-04T12:26:00.000+0000"}]}',
      'timestamp': '2023-08-04T12:31:29.595+0000'
    },
    'configuration': {
      'description': 'Metric Alarm to test Lambda actions',
      'metrics': [
        {
          'id': '1234e046-06f0-a3da-9534-EXAMPLEe4c',
          'metricStat': {
            'metric': {
              'namespace': 'AWS/Logs',
              'name': 'CallCount',
              'dimensions': {
                'InstanceId': 'i-12345678'
              }
            },
            'period': 60,
```

```
            'stat': 'Average',
            'unit': 'Percent'
          },
          'returnData': True
        }
      ]
    }
  }
}
```

Di seguito è riportato un esempio di oggetto evento da un allarme composito.

```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:111122223333:alarm:SuppressionDemo.Main',
  'accountId': '111122223333',
  'time': '2023-08-04T12:56:46.138+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'CompositeDemo.Main',
    'state': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:56:46.138+0000'
    },
    'previousState': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:54:46.138+0000',
      'actionsSuppressedBy': 'WaitPeriod',
      'actionsSuppressedReason': 'Actions suppressed by WaitPeriod'
    },
    'configuration': {
```

```
      'alarmRule': 'ALARM(CompositeDemo.FirstChild) OR
 ALARM(CompositeDemo.SecondChild)',
      'actionsSuppressor': 'CompositeDemo.ActionsSuppressor',
      'actionsSuppressorWaitPeriod': 120,
      'actionsSuppressorExtensionPeriod': 180
    }
  }
}
```

Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch

A volte, non tutti i dati previsti per una metrica vengono segnalati. CloudWatch Questo può accadere, ad esempio, quando una connessione viene persa, un server è inattivo oppure quando un parametro comunica dati solo a intermittenza per progetto.

CloudWatch consente di specificare come trattare i punti dati mancanti durante la valutazione di un allarme. Questo può aiutare a configurare l'allarme in modo che entri nello stato `ALARM` solo quando richiesto per il tipo di dati monitorati. È possibile evitare falsi positivi quando i dati mancanti non indicano un problema.

Analogamente al modo in cui ogni allarme si trova sempre in uno dei tre stati, ogni punto dati specifico a cui viene segnalato CloudWatch rientra in una delle tre categorie seguenti:

- Non superato (entro la soglia)
- Superato (fuori dalla soglia)
- Mancante

Per ogni allarme, puoi specificare di CloudWatch trattare i punti dati mancanti in uno dei seguenti modi:

- `notBreaching` - I punti dati mancanti vengono trattati come se fossero “corretti” e all'interno della soglia
- `breaching` - I punti dati mancanti vengono trattati come se fossero “errati” e violassero la soglia
- `ignore` - Lo stato dell'allarme attuale viene mantenuto
- `missing` - Se mancano tutti i punti di dati nell'intervallo di valutazione degli allarmi, l'allarme passa a INSUFFICIENT_DATA.

La scelta migliore dipende dal tipo di metrica e dallo scopo dell'allarme. Ad esempio, se state creando un allarme di rollback delle applicazioni utilizzando una metrica che riporta continuamente i dati, potreste considerare i punti dati mancanti come violazioni, perché ciò potrebbe indicare che qualcosa non va. Tuttavia, per un parametro che genera punti dati solo quando si verifica un errore, ad esempio `ThrottledRequests` in Amazon DynamoDB, i dati mancanti potrebbero venire trattati come `notBreaching`. Il comportamento predefinito è `missing`.

> **Important**
>
> Gli allarmi configurati sui parametri di Amazon EC2 possono entrare temporaneamente nello stato INSUFFICIENT_DATA se mancano dei punti dati dei parametri. Ciò è raro, ma può verificarsi quando il reporting dei parametri viene interrotto, anche quando l'istanza Amazon EC2 è integra. Per gli allarmi sui parametri di Amazon EC2 configurati per eseguire azioni di arresto, terminazione, riavvio o ripristino, ti consigliamo di configurare tali allarmi in modo che trattino i dati mancanti `missing` come e che questi allarmi si attivino solo quando si trova nello stato ALARM.

La scelta dell'opzione migliore per il tuo allarme previene modifiche della condizione dell'allarme non necessarie e fuorvianti e indica anche in maniera più accurata la verifica del sistema.

> **Important**
>
> Gli allarmi che valutano i parametri nello`AWS/DynamoDB`spazio dei nomi ignorano sempre i dati mancanti anche se si sceglie un'opzione diversa per il modo in cui l'allarme dovrebbe trattare i dati mancanti. Quando un`AWS/DynamoDB`parametro ha dati mancanti, gli allarmi che valutano quel parametro rimangono nello stato attuale.

Come viene valutato lo stato dell'allarme quando mancano i dati

Ogni volta che un allarme valuta se cambiare stato, CloudWatch tenta di recuperare un numero di punti dati superiore al numero specificato come Periodi di valutazione. L'esatto numero di punti di dati che tenta di recuperare dipende dalla durata del periodo di allarme e se si basa su un parametro con risoluzione standard o ad alta risoluzione. L'intervallo di tempo dei punti dati che tenta di recuperare è l'intervallo di valutazione.

Una volta CloudWatch recuperati questi punti dati, accade quanto segue:

- Se non mancano punti dati nell'intervallo di valutazione, CloudWatch valuta l'allarme in base ai punti dati più recenti raccolti. Il numero di punti dati valutato è uguale agli Evaluation Periods (Periodi di valutazione) per l'allarme. I punti dati aggiuntivi provenienti da un punto più indietro nel tempo nell'intervallo di valutazione non sono necessari e vengono ignorati.
- Se mancano alcuni punti dati nell'intervallo di valutazione, ma il numero totale di punti dati esistenti che sono stati recuperati con successo dall'intervallo di valutazione è uguale o superiore ai Periodi di valutazione dell'allarme, CloudWatch valuta lo stato dell'allarme in base ai punti dati reali più recenti che sono stati recuperati con successo, inclusi i punti dati aggiuntivi necessari da più lontano nell'intervallo di valutazione. In questo caso, il valore impostato per la modalità di gestione dei dati mancanti non è necessario e verrà ignorato.
- Se mancano alcuni punti dati nell'intervallo di valutazione e il numero di punti dati effettivi recuperati è inferiore al numero di periodi di valutazione dell'avviso, CloudWatch inserisce i punti dati mancanti con il risultato specificato per il trattamento dei dati mancanti, quindi valuta l'allarme. Tuttavia, tutti i punti dati reali nell'intervallo di valutazione sono inclusi nella valutazione. CloudWatch utilizza i punti dati mancanti solo il minor numero di volte possibile.

> **Note**
>
> Un caso particolare di questo comportamento è che gli CloudWatch allarmi potrebbero rivalutare ripetutamente l'ultimo set di punti dati per un periodo di tempo dopo che la metrica ha smesso di scorrere. Questa rivalutazione può comportare la modifica dello stato dell'allarme e una nuova esecuzione delle operazioni, se lo stato fosse stato modificato immediatamente prima dell'arresto del flusso del parametro. Per mitigare questo comportamento, utilizzare periodi più brevi.

Le seguenti tabelle illustrano esempi del comportamento di valutazione dell'allarme. Nella prima tabella, Datapoints to Alarm e Evaluation Periods sono entrambi 3. CloudWatch recupera i 5 punti dati più recenti durante la valutazione dell'allarme, nel caso in cui manchino alcuni dei 3 punti dati più recenti. 5 è l'intervallo di valutazione dell'allarme.

Nella colonna 1 vengono visualizzati i 5 punti dati più recenti, poiché l'intervallo di valutazione è 5. Questi punti dati vengono visualizzati con il punto di dati più recente a destra. 0 è un punto di dati che non supera la soglia, X è un punto di dati che viola la soglia e - è un punto di dati mancante.

Nella colonna 2 sono mostrati quanti dei 3 punti di dati necessari risultano mancanti. Anche se vengono valutati gli ultimi 5 punti di dati, ne sono necessari solo 3 (l'impostazione di Evaluation

Periods (Periodi di valutazione)) per valutare lo stato dell'allarme. Il numero di punti di dati nella colonna 2 rappresenta il numero di dati che devono essere "riempiti", utilizzando l'impostazione relativa al trattamento dei dati mancanti.

Nelle colonne 3-6, le intestazioni di colonna sono i valori possibili per come trattare i dati mancanti. Le righe di queste colonne mostrano lo stato di allarme impostato per ciascuno di questi modi possibili per trattare i dati mancanti.

Punti di dati	N di punti di dati che devono essere riempiti	MANCANTE	IGNORA	VIOLAZIONE	NON VIOLAZIONE
0 - X - X	0	OK	OK	OK	OK
- - - - 0	2	OK	OK	OK	OK
- - - - -	3	INSUFFICIENT_DATA	Mantieni lo stato attuale	ALARM	OK
0 X X - X	0	ALARM	ALARM	ALARM	ALARM
- - X - -	2	ALARM	Mantieni lo stato attuale	ALARM	OK

Nella seconda riga della tabella precedente, l'allarme rimane OK anche se i dati mancanti vengono trattati come violazione, perché il singolo punto dati esistente non sta effettuando una violazione e questo aspetto viene valutato insieme a due punti dati mancanti trattati come violazione. La prossima volta in cui questo allarme viene valutato, se i dati sono ancora mancanti si visualizzerà ALARM e il punto dati di non-violazione non rientrerà più nell'intervallo di valutazione.

La terza riga, in cui mancano tutti e cinque i punti di dati più recenti, illustra come le varie impostazioni per il trattamento dei dati mancanti influiscano sullo stato dell'allarme. Se i punti di dati mancanti sono considerati una violazione, l'allarme entra in stato ALARM, mentre se non sono considerati una violazione, l'allarme entra in stato OK. Se i punti di dati mancanti vengono ignorati, l'allarme mantiene lo stato corrente che aveva prima dei punti di dati mancanti. E se i punti di dati

mancanti sono solo considerati mancanti, allora l'allarme non ha abbastanza dati reali recenti per effettuare una valutazione ed entra nello stato INSUFFICIENT_DATA.

Nella quarta riga, l'allarme entra nello stato ALARM in tutti i casi perché i tre punti di dati più recenti costituiscono una violazione, e gli Evaluation Periods (Periodi di valutazione) e i Datapoints to Alarm (Punti di dati all'allarme) sono entrambi impostati su 3. In questo caso, il punto di dati mancante viene ignorato e l'impostazione della modalità di valutazione dei dati mancanti non è necessaria, poiché sono disponibili 3 punti di dati reali da valutare.

La riga 5 rappresenta un caso speciale di valutazione dell'allarme chiamato stato di allarme prematuro. Per ulteriori informazioni, consulta la pagina Evitare transizioni premature allo stato di allarme.

Nella tabella seguente, il Period (Periodo) è di nuovo impostato su 5 minuti e Datapoints to Alarm (Punti dati all'allarme) è solo 2 mentre i Evaluation Periods (Periodi di valutazione) è 3. Questo è un 2 su 3, allarme M di N.

L'intervallo di valutazione è 5. Questo è il numero massimo di punti dati recenti che vengono recuperati e che è possibile utilizzare nel caso in cui alcuni punti dati risultino mancanti.

Punti di dati	Numero di punti di dati mancanti	MANCANTE	IGNORA	VIOLAZIONE	NON VIOLAZIONE
0 - X - X	0	ALARM	ALARM	ALARM	ALARM
0 0 X 0 X	0	ALARM	ALARM	ALARM	ALARM
0 - X - -	1	OK	OK	ALARM	OK
- - - - 0	2	OK	OK	ALARM	OK
- - - - X	2	ALARM	Mantieni lo stato attuale	ALARM	OK

Nelle righe 1 e 2, l'allarme passa sempre allo stato ALARM perché 2 dei 3 punti di dati più recenti stanno costituendo una violazione. Nella riga 2, i due punti di dati più vecchi nell'intervallo di valutazione non sono necessari perché non manca nessuno dei tre punti di dati più recenti, quindi questi due punti di dati meno recenti vengono ignorati.

Nelle righe 3 e 4, l'allarme passa allo stato ALARM solo se i dati mancanti vengono trattati come violazione, nel qual caso i due punti di dati mancanti più recenti vengono entrambi trattati come violazione. Nella riga 4, questi due punti di dati mancanti trattati come una violazione forniscono i due punti di dati oggetto violazione necessari per attivare lo stato ALARM.

La riga 5 rappresenta un caso speciale di valutazione dell'allarme chiamato stato di allarme prematuro. Per ulteriori informazioni, consulta la sezione seguente.

Evitare transizioni premature allo stato di allarme

CloudWatch la valutazione degli allarmi include una logica per cercare di evitare falsi allarmi, in cui l'allarme passa prematuramente allo stato ALARM quando i dati sono intermittenti. L'esempio mostrato nella riga 5 delle tabelle della sezione precedente illustra questa logica. In queste righe e negli esempi seguenti, gli Evaluation Periods (Periodi di valutazione) sono 3 e l'intervallo di valutazione è di 5 punti di dati. Datapoints to Alarm (Punti di dati all'allarme) è 3, ad eccezione dell'esempio M di N, dove Datapoints to alarm (Punti di dati all'allarme) è 2.

Supponiamo che i dati più recenti di un allarme siano - - - - X, con quattro punti di dati mancanti e quindi un punto di dati oggetto di violazione come punto di dati più recente. Poiché il punto di dati successivo potrebbe non costituire una violazione, l'allarme non entra immediatamente in stato ALARM quando i dati sono - - - - X o - - - X - e Datapoints to Alarm (Punti di dati all'allarme) è 3. In questo modo, i falsi positivi vengono evitati quando il punto di dati successivo non costituisce una violazione e fa sì che i dati siano - - - X 0 o - - X - 0.

Tuttavia, se gli ultimi punti di dati sono - - X - -, l'allarme entra in stato ALARM anche se i punti di dati mancanti vengono trattati come mancanti. Questo perché gli allarmi sono progettati per entrare sempre nello stato ALARM quando il punto di dati oggetto di violazione meno recente disponibile durante il numero di Periodi di valutazione di punti di dati è vecchio almeno quanto il valore di Punti di dati all'allarme e tutti gli altri punti di dati più recenti costituiscono una violazione o sono mancanti. In questo caso, l'allarme entra in stato ALARM anche se il numero totale di punti di dati disponibili è inferiore a M (Datapoints to Alarm (Punti di dati all'allarme)).

Questa logica di allarme si applica anche a M allarmi su N. Se il punto di dati oggetto di violazione meno recente durante l'intervallo di valutazione è vecchio almeno quanto il valore di Datapoints to Alarm (Punti di dati all'allarme) e tutti i punti di dati più recenti costituiscono una violazione o sono mancanti, l'allarme entra in stato ALARM indipendentemente dal valore di M (Datapoints to Alarm (Punti di dati all'allarme)).

Allarmi ad alta risoluzione

Se imposti un allarme su un parametro ad alta risoluzione, puoi specificare un allarme ad alta risoluzione con un periodo di 10 secondi o 30 secondi, oppure puoi impostare un allarme regolare con un periodo di qualsiasi multiplo di più di 60 secondi. Per gli allarmi ad alta risoluzione il costo è più elevato. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Pubblicare i parametri personalizzati di .

Allarmi basati su espressioni matematiche

È possibile impostare un allarme in base al risultato di un'espressione matematica basata su una o più metriche. CloudWatch Un'espressione matematica utilizzata per un allarme può includere fino a 10 parametri. Ogni parametro deve utilizzare lo stesso periodo.

Per un allarme basato su un'espressione matematica, puoi specificare come vuoi CloudWatch trattare i punti dati mancanti. In questo caso, il punto dati viene considerato mancante se l'espressione matematica non restituisce un valore per quel punto dati.

Allarmi basati su espressioni matematiche non possono eseguire operazioni Amazon EC2.

Per ulteriori informazioni sulle espressioni matematiche e la sintassi dei parametri, consulta Utilizzare la matematica dei parametri.

CloudWatch Allarmi basati su percentili ed esempi di dati limitati

Quando si imposta un percentile come la statistica per un allarme, puoi specificare come gestire i dati che non sono sufficienti per una buona valutazione statistica. Puoi scegliere di impostare l'allarme in modo che valuti in ogni caso le statistiche e, possibilmente, modifichi lo stato dell'allarme. In alternativa, puoi impostare l'allarme in modo che ignori il parametro quando le dimensioni dell'esempio sono ridotte e in modo che attenda per valutarli finché non sono presenti abbastanza dati per essere significativi a livello statistico.

Per percentili tra 0,5 (incluso) e 1,00 (escluso), questa impostazione viene utilizzata quando sono presenti meno punti di dati di 10/(1-percentile) durante il periodo di valutazione. Ad esempio, questa impostazione potrebbe essere utilizzata se fossero presenti meno di 1.000 esempi per un allarme su un percentile di p99. Per percentili tra 0 e 0,5 (escluso), l'impostazione viene utilizzata quando sono presenti meno punti di dati di 10/percentile.

Caratteristiche comuni degli allarmi CloudWatch

Le seguenti funzionalità si applicano a tutti gli CloudWatch allarmi:

- Non è previsto alcun limite per il numero di allarmi che puoi creare. Per creare o aggiornare un avviso, si utilizza la CloudWatch console, l'azione PutMetricAlarmAPI o il put-metric-alarmcomando contenuto in. AWS CLI
- I nomi degli allarmi devono contenere solo caratteri UTF-8 e non possono contenere caratteri di controllo ASCII
- Puoi elencare alcuni o tutti gli allarmi attualmente configurati ed elencare tutti gli allarmi in uno stato particolare utilizzando la CloudWatch console, l'azione DescribeAlarmsAPI o il comando describe-alarms in. AWS CLI
- Puoi disabilitare e abilitare gli allarmi utilizzando le azioni DisableAlarmActionse EnableAlarmActionsAPI o i comandi and in. disable-alarm-actionsenable-alarm-actions AWS CLI
- È possibile testare un allarme impostandolo su qualsiasi stato utilizzando l'azione SetAlarmStateAPI o il set-alarm-statecomando in. AWS CLI Questa modifica temporanea dello stato permane solamente finché non viene effettuato un successivo confronto tra allarmi.
- È possibile creare un allarme per una metrica personalizzata prima di creare quella metrica personalizzata. Affinché l'allarme sia valido, è necessario includere tutte le dimensioni per il parametro personalizzato in aggiunta allo spazio dei nomi parametro e al nome parametro nella definizione dell'allarme. A tale scopo, è possibile utilizzare l'azione PutMetricAlarmAPI o il put-metric-alarmcomando contenuto in AWS CLI.
- È possibile visualizzare la cronologia di un allarme utilizzando la CloudWatch console, l'azione DescribeAlarmHistoryAPI o il describe-alarm-historycomando in AWS CLI. CloudWatch conserva la cronologia degli allarmi per 30 giorni. Ogni transizione di stato viene contrassegnata con un timestamp univoco. In rari casi, la cronologia potrebbe mostrare più di una notifica per una modifica di stato. Il timestamp consente di confermare le modifiche di stato univoche.
- Puoi aggiungere avvisi ai preferiti dall'opzione Preferiti e recenti nel pannello di navigazione della CloudWatch console passando il mouse sulla sveglia che desideri aggiungere ai preferiti e scegliendo il simbolo a forma di stella accanto ad essa.
- Il numero di periodi di valutazione per un allarme moltiplicato per la durata di ciascun periodo di valutazione non può superare un giorno.

> **Note**
>
> Alcune AWS risorse non inviano dati metrici a in determinate condizioni. CloudWatch Ad esempio, Amazon EBS potrebbe non inviare i dati dei parametri per un volume disponibile non collegato a un'istanza Amazon EC2, poiché non vi è alcuna attività dei parametri da monitorare per tale volume. Se disponi di un set di allarmi per tale parametro, il relativo stato potrebbe cambiare in `INSUFFICIENT_DATA`. Questo potrebbe indicare che la risorsa non è attiva e non necessariamente significare la presenza di un problema. È possibile specificare il modo in cui ogni allarme tratta i dati mancanti. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .

Le migliori pratiche e consigli sugli allarmi per i servizi AWS

CloudWatch fornisce consigli sugli allarmi dei out-of-the box. Si tratta di CloudWatch allarmi che consigliamo di creare per le metriche pubblicate da altri AWS servizi. Queste raccomandazioni possono aiutarti a identificare i parametri per i quali attivare gli allarmi, in linea con le best practice per il monitoraggio. Le raccomandazioni suggeriscono anche le soglie di allarme da impostare. Seguire questi consigli può aiutarvi a non perdere un importante monitoraggio della vostra AWS infrastruttura.

Per trovare i consigli sugli allarmi, utilizza la sezione delle metriche della CloudWatch console e seleziona l'interruttore del filtro dei consigli sugli allarmi. Se accedi agli avvisi consigliati nella console e poi crei un allarme consigliato, CloudWatch puoi precompilare alcune delle impostazioni degli allarmi. Per alcuni allarmi raccomandati, anche il valore della soglia di allarme è precompilato. Puoi anche utilizzare la console per scaricare infrastructure-as-code le definizioni degli allarmi consigliati e quindi utilizzare questo codice per creare l'allarme in AWS CloudFormation AWS CLI, the o Terraform.

È inoltre possibile visualizzare l'elenco degli allarmi consigliati in Allarmi raccomandati.

Ti vengono addebitati i costi per gli allarmi che crei, alla stessa velocità di qualsiasi altro allarme creato in. CloudWatch L'utilizzo delle raccomandazioni non comporta costi aggiuntivi. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

Ricerca e creazione di allarmi raccomandati

Segui questi passaggi per trovare le metriche per cui CloudWatch ti consigliamo di impostare gli allarmi e, facoltativamente, per creare uno di questi allarmi. La prima procedura spiega come trovare i parametri per i quali gli allarmi sono raccomandati e come creare uno di questi allarmi.

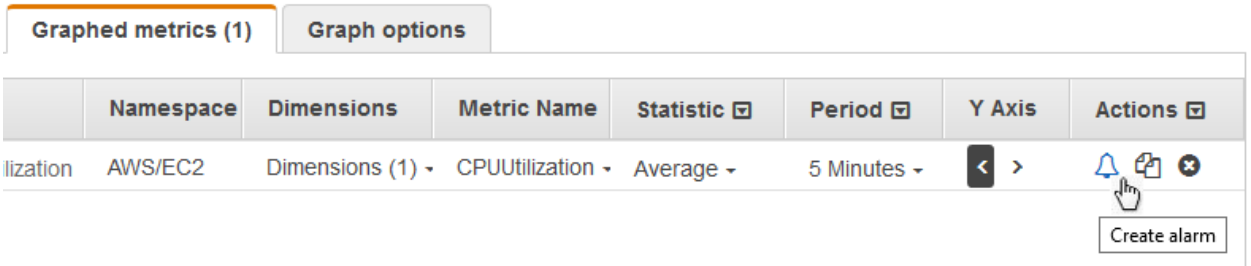
Puoi anche scaricare in blocco le definizioni degli infrastructure-as-code allarmi per tutti gli allarmi consigliati in un AWS namespace, ad esempio o. `AWS/Lambda` `AWS/S3` Le relative istruzioni vengono fornite più avanti in questo argomento.

Individuazione dei parametri con gli allarmi raccomandati e creazione di un unico allarme raccomandato

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nel pannello di navigazione, seleziona Metrics (Parametri), All metrics (Tutti i parametri).
3. Sopra la tabella Parametri, scegli Raccomandazioni di allarme.

   L'elenco degli spazi dei nomi dei parametri viene filtrato in modo da includere soltanto i parametri che dispongono di raccomandazioni di allarme e che i servizi dell'account stanno pubblicando.
4. Scegli lo spazio dei nomi per un servizio.

   L'elenco dei parametri in questo spazio dei nomi viene filtrato in modo da includere soltanto quelli che dispongono di raccomandazioni di allarme.
5. Per visualizzare lo scopo dell'allarme e la soglia raccomandata per un parametro, scegli Visualizza dettagli.
6. Per creare un allarme per uno dei parametri, procedi in uno dei seguenti modi:
   - Per utilizzare la console per creare un allarme, procedi come segue:
     a. Seleziona la casella di controllo relativa al parametro e scegli la scheda Parametri definiti.
     b. Seleziona l'icona dell'allarme.



     Viene visualizzata la procedura guidata per la creazione dell'allarme, con il nome del parametro, la statistica e il periodo compilati in base alla raccomandazione di allarme. Se la raccomandazione include un valore di soglia specifico, anche quel valore è precompilato.
     c. Seleziona Successivo.

d. In Notifica, seleziona un argomento SNS per segnalare quando l'allarme passa allo stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

   Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

   Per fare in modo che l'allarme non invii notifiche, scegli Remove (Rimuovi).

e. Per fare in modo che l'allarme esegua operazioni Auto Scaling o EC2, scegli il pulsante appropriato e scegli lo stato di allarme e l'operazione da eseguire.

f. Al termine, scegli Apply (Applica).

g. Inserisci un nome e una descrizione per l'allarme. Il nome deve contenere solo caratteri ASCII. Quindi scegli Successivo.

h. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

- Per scaricare una definizione di infrastructure-as-code allarme da utilizzare in Terraform o in Terraform, scegli Scarica codice di allarme e seleziona il formato che desideri. AWS CloudFormation AWS CLI Il codice scaricato avrà le impostazioni raccomandate per il nome del parametro, la statistica e la soglia.

Per scaricare le definizioni infrastructure-as-code degli allarmi per tutti gli allarmi consigliati per un servizio AWS

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), All metrics (Tutti i parametri).
3. Sopra la tabella Parametri, scegli Raccomandazioni di allarme.

   L'elenco degli spazi dei nomi dei parametri viene filtrato in modo da includere soltanto i parametri che dispongono di raccomandazioni di allarme e che i servizi dell'account stanno pubblicando.
4. Scegli lo spazio dei nomi per un servizio.

   L'elenco dei parametri in questo spazio dei nomi viene filtrato in modo da includere soltanto quelli che dispongono di raccomandazioni di allarme.
5. Il codice di allarme Scarica mostra quanti allarmi sono consigliati per le metriche in questo namespace. Per scaricare le definizioni infrastructure-as-code degli allarmi per tutti gli allarmi consigliati, scegli Scarica codice di allarme, quindi scegli il formato di codice che desideri.

Allarmi raccomandati

Le sezioni seguenti elencano i parametri per i quali suggeriamo di attivare gli allarmi basati sulle best practice. Per ogni parametro, vengono visualizzati anche le dimensioni, lo scopo dell'allarme, la soglia consigliata, la giustificazione della soglia, la durata del periodo e il numero di punti dati.

Alcuni parametri potrebbero apparire due volte nell'elenco. Ciò accade quando allarmi diversi sono raccomandati per combinazioni differenti di dimensioni di tale parametro.

Il valore Punti dati su cui attivare allarmi rappresenta il numero di punti dati che devono essere violati per mettere l'allarme nello stato ALARM. Il valore Periodi di valutazione rappresenta il numero di periodi che vengono presi in considerazione quando viene valutato l'allarme. Se questi due numeri coincidono, l'allarme passa allo stato ALARM solo quando tale numero di periodi consecutivi presenta valori che superano la soglia. Se Punti dati su cui attivare allarmi è inferiore a Periodi di valutazione, l'allarme è del tipo "M su N" e passa allo stato ALARM se almeno un numero di punti dati pari a Punti dati su cui attivare allarmi viene violato all'interno di qualsiasi set di punti dati pari a Periodi di valutazione. Per ulteriori informazioni, consulta Valutazione di un allarme.

Argomenti

- Lambda Insights
- Amazon VPC (AWS/NATGateway)
- AWS Link privato () AWS/PrivateLinkEndpoints
- AWS Link privato (AWS/PrivateLinkServices)
- Amazon RDS
- Amazon Route 53 Public Data Plane
- Amazon S3
- S3ObjectLambda
- Amazon SNS
- Amazon SQS
- AWS VPN

Amazon API Gateway

4XXError

Dimensioni:ApiName, Stage

Descrizione dell'allarme: questo allarme rileva un tasso elevato di errori lato client. Ciò può indicare un problema nei parametri di autorizzazione o di richiesta del client. Potrebbe anche significare che una risorsa è stata rimossa o che un client ne sta richiedendo una che non esiste. Valuta la possibilità di abilitare CloudWatch i registri e verificare la presenza di eventuali errori che potrebbero causare gli errori 4XX. Inoltre, valuta la possibilità di abilitare CloudWatch metriche dettagliate per visualizzare questa metrica per risorsa e metodo e restringere la fonte degli errori. Gli errori potrebbero essere causati anche dal superamento del limite di limitazione della larghezza di banda della rete impostato. Se le risposte e i log riportano percentuali elevate e impreviste di 429 errori, segui questa guida per risolvere il problema.

Scopo: questo allarme può rilevare tassi elevati di errori lato client per le richieste di Gateway API.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: la soglia raccomandata rileva quando più del 5% del totale delle richieste presenta errori di tipo 4XX. Tuttavia, puoi regolare la soglia in base al traffico relativo alle richieste e ai tassi di errore accettabili. Inoltre, è possibile analizzare i dati storici per determinare

il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. È necessario attivare gli allarmi per gli errori 4XX che si verificano frequentemente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

5XXError

Dimensioni:ApiName, Stage

Descrizione dell'allarme: questo allarme aiuta a rilevare un tasso elevato di errori lato client. Ciò può indicare un errore nel back-end dell'API, nella rete o nell'integrazione tra il gateway API e l'API di back-end. Questa documentazione può aiutarti a risolvere la causa degli errori 5xx.

Scopo: questo allarme può rilevare tassi elevati di errori lato server per le richieste di Gateway API.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: la soglia raccomandata rileva quando più del 5% del totale delle richieste presenta errori di tipo 5XX. Tuttavia, è possibile regolare la soglia in base al traffico delle richieste e ai tassi di errore accettabili. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. È necessario attivare gli allarmi per gli errori 5XX che si verificano frequentemente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: GREATER_THAN_THRESHOLD

Conteggio

Dimensioni:ApiName, Stage

Descrizione dell'allarme: questo allarme aiuta a rilevare un basso volume di traffico per la fase della REST API. Questo potrebbe segnalare un problema con l'applicazione che chiama l'API, ad esempio a causa dell'utilizzo di endpoint errati. Potrebbe anche essere un indicatore di un problema relativo alla configurazione o alle autorizzazioni dell'API che la rendono irraggiungibile per i client.

Scopo: questo allarme può rilevare un volume di traffico inaspettatamente basso per la fase di REST API. Consigliamo di creare questo allarme se l'API invia un numero prevedibile e regolare di richieste in condizioni normali. Se hai abilitato le CloudWatch metriche dettagliate e puoi prevedere il normale volume di traffico per metodo e risorsa, ti consigliamo di creare allarmi alternativi per avere un monitoraggio più preciso delle cadute di volume di traffico per ogni risorsa e metodo. Questo allarme non è raccomandato per le API che non prevedono un traffico costante e regolare.

Statistica: SampleCount

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base all'analisi dei dati storici per determinare il numero di richieste di base previsto per l'API. L'impostazione della soglia su un valore molto alto potrebbe rendere l'allarme troppo sensibile nei periodi di traffico normale e di traffico basso previsto. Viceversa, l'impostazione su un valore molto basso potrebbe far sì che l'allarme non rilevi piccoli cali anomali del volume di traffico.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: LESS_THAN_THRESHOLD

Conteggio

Dimensioni: faseApiName, risorsa, metodo

Descrizione dell'allarme: questo allarme aiuta a rilevare un basso volume di traffico per la risorsa e il metodo nella fase di REST API. Ciò può indicare un problema con l'applicazione

che chiama l'API, ad esempio a causa dell'utilizzo di endpoint errati. Potrebbe anche essere un indicatore di un problema relativo alla configurazione o alle autorizzazioni dell'API che la rendono irraggiungibile per i client.

Scopo: questo allarme può rilevare un volume di traffico inaspettatamente basso per la risorsa e il metodo nella fase di REST API. Consigliamo di creare questo allarme se l'API invia un numero prevedibile e regolare di richieste in condizioni normali. Questo allarme non è raccomandato per le API che non prevedono un traffico costante e regolare.

Statistica: SampleCount

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base all'analisi dei dati storici per determinare il numero di richieste di base previsto per l'API. L'impostazione della soglia su un valore molto alto potrebbe rendere l'allarme troppo sensibile nei periodi di traffico normale e di traffico basso previsto. Viceversa, l'impostazione su un valore molto basso potrebbe far sì che l'allarme non rilevi piccoli cali anomali del volume di traffico.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: LESS_THAN_THRESHOLD

Conteggio

Dimensioni:ApiId, Stage

Descrizione dell'allarme: questo allarme aiuta a rilevare un basso volume di traffico per la fase dell'API HTTP. Ciò può indicare un problema con l'applicazione che chiama l'API, ad esempio a causa dell'utilizzo di endpoint errati. Potrebbe anche essere un indicatore di un problema relativo alla configurazione o alle autorizzazioni dell'API che la rendono irraggiungibile per i client.

Scopo: questo allarme può rilevare un volume di traffico inaspettatamente basso per la fase dell'API HTTP. Consigliamo di creare questo allarme se l'API invia un numero prevedibile e regolare di richieste in condizioni normali. Se hai abilitato le CloudWatch metriche dettagliate e puoi prevedere il normale volume di traffico per percorso, ti consigliamo di creare allarmi alternativi a questo per avere un monitoraggio più preciso delle cadute di volume di traffico per

ogni percorso. Questo allarme non è raccomandato per le API che non prevedono un traffico costante e regolare.

Statistica: SampleCount

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta il valore della soglia in base all'analisi dei dati storici per determinare il numero di richieste di base previsto per l'API. L'impostazione della soglia su un valore molto alto potrebbe rendere l'allarme troppo sensibile nei periodi di traffico normale e di traffico basso previsto. Viceversa, l'impostazione su un valore molto basso potrebbe far sì che l'allarme non rilevi piccoli cali anomali del volume di traffico.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: LESS_THAN_THRESHOLD

Conteggio

Dimensioni: faseApiId, risorsa, metodo

Descrizione dell'allarme: questo allarme aiuta a rilevare un basso volume di traffico per la route dell'API HTTP nella fase. Ciò può indicare un problema con l'applicazione che chiama l'API, ad esempio a causa dell'utilizzo di endpoint errati. Potrebbe anche indicare un problema relativo alla configurazione o alle autorizzazioni dell'API che la rendono irraggiungibile per i client.

Scopo: questo allarme può rilevare un volume di traffico inaspettatamente basso per la route dell'API HTTP nella fase. Consigliamo di creare questo allarme se l'API invia un numero prevedibile e regolare di richieste in condizioni normali. Questo allarme non è raccomandato per le API che non prevedono un traffico costante e regolare.

Statistica: SampleCount

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta il valore della soglia in base all'analisi dei dati storici per determinare il numero di richieste di base previsto per l'API. L'impostazione della soglia su un

valore molto alto potrebbe rendere l'allarme troppo sensibile nei periodi di traffico normale e di traffico basso previsto. Viceversa, l'impostazione su un valore molto basso potrebbe far sì che l'allarme non rilevi piccoli cali anomali del volume di traffico.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: LESS_THAN_THRESHOLD

IntegrationLatency

Dimensioni:ApiId, Stage

Descrizione dell'allarme: questo allarme aiuta a rilevare un'elevata latenza di integrazione per le richieste API in una fase. Puoi stabilire un collegamento tra il valore del parametro `IntegrationLatency` e il parametro della latenza corrispondente per il back-end, ad esempio il parametro `Duration` per le integrazioni Lambda. Questo ti aiuta a determinare se il back-end dell'API impiega più tempo a elaborare le richieste dei client a causa di problemi di prestazioni o se sussiste qualche altro sovraccarico dovuto all'inizializzazione o all'avvio a freddo. Inoltre, valuta la possibilità di abilitare CloudWatch i log per la tua API e di controllare i log per eventuali errori che potrebbero causare problemi di elevata latenza. Inoltre, valuta la possibilità di abilitare CloudWatch metriche dettagliate per avere una visione di questa metrica per percorso, per aiutarti a restringere l'origine della latenza di integrazione.

Scopo: questo allarme può rilevare quando le richieste di Gateway API in una fase hanno una latenza di integrazione elevata. Consigliamo questo allarme per le WebSocket API e lo consideriamo facoltativo per le API HTTP, in quanto offrono già raccomandazioni di allarme separate per la metrica di latenza. Se hai abilitato CloudWatch metriche dettagliate e hai requisiti prestazionali di latenza di integrazione diversi per percorso, ti consigliamo di creare allarmi alternativi per avere un monitoraggio più dettagliato della latenza di integrazione per ciascuna route.

Statistica: p90

Soglia raccomandata: 2.000,0

Giustificazione della soglia: il valore di soglia raccomandato non funziona per tutti i carichi di lavoro dell'API. Tuttavia, puoi utilizzarlo come punto di partenza per la soglia. Dopodiché, è

possibile scegliere valori di soglia diversi in base al carico di lavoro e ai requisiti di latenza, prestazioni e SLA accettabili per l'API. Se è accettabile che l'API abbia una latenza più elevata in generale, imposta un valore di soglia più alto per rendere l'allarme meno sensibile. Tuttavia, se è necessario che l'API fornisca risposte quasi in tempo reale, imposta un valore di soglia inferiore. Inoltre, è possibile analizzare i dati storici per determinare la latenza di base prevista per il carico di lavoro dell'applicazione e ottimizzare il valore della soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

IntegrationLatency

Dimensioni:, tappa, percorso ApiId

Descrizione dell'allarme: questo allarme aiuta a rilevare se c'è un'elevata latenza di integrazione per le richieste WebSocket API per un percorso in una fase. Puoi stabilire un collegamento tra il valore del parametro `IntegrationLatency` e il parametro della latenza corrispondente per il back-end, ad esempio il parametro `Duration` per le integrazioni Lambda. Questo ti aiuta a determinare se il back-end dell'API impiega più tempo a elaborare le richieste dei client a causa di problemi di prestazioni o se sussiste qualche altro sovraccarico dovuto all'inizializzazione o all'avvio a freddo. Inoltre, valuta la possibilità di abilitare CloudWatch i log per la tua API e di controllare i log per eventuali errori che potrebbero causare problemi di alta latenza.

Scopo: questo allarme può rilevare quando le richieste di Gateway API per una route in una fase hanno una latenza di integrazione elevata.

Statistica: p90

Soglia raccomandata: 2.000,0

Giustificazione della soglia: il valore di soglia raccomandato non funziona per tutti i carichi di lavoro dell'API. Tuttavia, puoi utilizzarlo come punto di partenza per la soglia. Dopodiché, è possibile scegliere valori di soglia diversi in base al carico di lavoro e ai requisiti di latenza, prestazioni e SLA accettabili per l'API. Se è accettabile che l'API abbia una latenza più elevata in generale, puoi impostare un valore di soglia più alto per rendere l'allarme meno sensibile.

Tuttavia, se è necessario che l'API fornisca risposte quasi in tempo reale, imposta un valore di soglia inferiore. Inoltre, è possibile analizzare i dati storici per determinare la latenza di base prevista per il carico di lavoro dell'applicazione e ottimizzare il valore della soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latenza

Dimensioni:, palco ApiName

Descrizione dell'allarme: questo allarme rileva un'elevata latenza in una fase. Trova il valore del parametro `IntegrationLatency` per verificare la latenza del back-end dell'API. Se i due parametri sono per lo più allineati, il back-end dell'API è l'origine della latenza più elevata e dovrebbe essere esaminato per individuare eventuali problemi. Prendi in considerazione anche l'attivazione CloudWatch dei log e il controllo degli errori che potrebbero causare l'elevata latenza. Inoltre, valuta la possibilità di abilitare CloudWatch metriche dettagliate per visualizzare questa metrica per risorsa e metodo e restringere la fonte della latenza. Se applicabile, consulta le guide alla risoluzione dei problemi con Lambda o alla risoluzione dei problemi per gli endpoint API ottimizzati per l'edge.

Scopo: questo allarme può rilevare quando le richieste di Gateway API in una fase hanno una latenza elevata. Se hai abilitato CloudWatch metriche dettagliate e hai requisiti di prestazioni di latenza diversi per ogni metodo e risorsa, ti consigliamo di creare allarmi alternativi per avere un monitoraggio più dettagliato della latenza per ogni risorsa e metodo.

Statistica: p90

Soglia raccomandata: 2.500,0

Giustificazione della soglia: il valore di soglia raccomandato non funziona per tutti i carichi di lavoro dell'API. Tuttavia, puoi utilizzarlo come punto di partenza per la soglia. Dopodiché, è possibile scegliere valori di soglia diversi in base al carico di lavoro e ai requisiti di latenza, prestazioni e SLA accettabili per l'API. Se è accettabile che l'API abbia una latenza più elevata in generale, puoi impostare un valore di soglia più alto per rendere l'allarme meno sensibile. Tuttavia, se è necessario che l'API fornisca risposte quasi in tempo reale, imposta un valore di

soglia inferiore. Inoltre, è possibile analizzare i dati storici per determinare la latenza di base prevista per il carico di lavoro dell'applicazione e ottimizzare il valore della soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latenza

Dimensioni: fase, risorsa, metodo ApiName

Descrizione dell'allarme: questo allarme rileva un'elevata latenza per un metodo e una risorsa in una fase. Trova il valore del parametro `IntegrationLatency` per verificare la latenza del back-end dell'API. Se i due parametri sono per lo più allineati, il back-end dell'API è l'origine della latenza più elevata e dovrebbe essere esaminato per individuare eventuali problemi di prestazioni. Prendi in considerazione anche l'attivazione CloudWatch dei log e il controllo di eventuali errori che potrebbero causare l'elevata latenza. Se applicabile, puoi anche fare riferimento alle guide alla risoluzione dei problemi con Lambda o alla risoluzione dei problemi per gli endpoint API ottimizzati per l'edge.

Scopo: questo allarme può rilevare quando le richieste di Gateway API per una risorsa e un metodo in una fase hanno una latenza elevata.

Statistica: p90

Soglia raccomandata: 2.500,0

Giustificazione della soglia: il valore di soglia raccomandato non funziona per tutti i carichi di lavoro dell'API. Tuttavia, puoi utilizzarlo come punto di partenza per la soglia. Dopodiché, è possibile scegliere valori di soglia diversi in base al carico di lavoro e ai requisiti di latenza, prestazioni e SLA accettabili per l'API. Se è accettabile che l'API abbia una latenza più elevata in generale, puoi impostare un valore di soglia più alto per rendere l'allarme meno sensibile. Tuttavia, se è necessario che l'API fornisca risposte quasi in tempo reale, imposta un valore di soglia inferiore. Inoltre, è possibile analizzare i dati storici per determinare la latenza di base prevista per il carico di lavoro dell'applicazione e ottimizzare il valore della soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latenza

Dimensioni:ApiId, Stage

Descrizione dell'allarme: questo allarme rileva un'elevata latenza in una fase. Trova il valore del parametro `IntegrationLatency` per verificare la latenza del back-end dell'API. Se i due parametri sono per lo più allineati, il back-end dell'API è l'origine della latenza più elevata e dovrebbe essere esaminato per individuare eventuali problemi di prestazioni. Valuta anche la possibilità di abilitare CloudWatch i log e di verificare la presenza di eventuali errori che potrebbero causare l'elevata latenza. Inoltre, valuta la possibilità di abilitare CloudWatch metriche dettagliate per visualizzare questa metrica per percorso e restringere la fonte della latenza. Puoi anche fare riferimento alla guida alla risoluzione dei problemi con le integrazioni Lambda, se applicabile.

Scopo: questo allarme può rilevare quando le richieste di Gateway API in una fase hanno una latenza elevata. Se hai abilitato CloudWatch metriche dettagliate e hai requisiti di prestazioni di latenza diversi per percorso, ti consigliamo di creare allarmi alternativi per avere un monitoraggio più dettagliato della latenza per ogni percorso.

Statistica: p90

Soglia raccomandata: 2.500,0

Giustificazione della soglia: il valore di soglia raccomandato non funziona per tutti i carichi di lavoro dell'API. Tuttavia, può essere utilizzato come punto di partenza per la soglia. Dopodiché, è possibile scegliere valori di soglia diversi in base al carico di lavoro e ai requisiti di latenza, prestazioni e SLA accettabili per l'API. Se in generale è accettabile che l'API abbia una latenza più elevata, puoi impostare un valore di soglia più elevato per renderla meno sensibile. Tuttavia, se all'API è richiesto di fornire risposte quasi in tempo reale, imposta un valore di soglia inferiore. Inoltre, è possibile analizzare i dati storici per determinare la latenza di base prevista per il carico di lavoro dell'applicazione e ottimizzare il valore della soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latenza

Dimensioni:, fase, risorsa, metodo ApiId

Descrizione dell'allarme: questo allarme rileva un'elevata latenza per una route in una fase. Trova il valore del parametro `IntegrationLatency` per verificare la latenza del back-end dell'API. Se i due parametri sono per lo più allineati, il back-end dell'API è l'origine della latenza più elevata e dovrebbe essere esaminato per individuare eventuali problemi di prestazioni. Prendi in considerazione anche l'abilitazione CloudWatch dei log e il controllo di eventuali errori che potrebbero causare l'elevata latenza. Puoi anche fare riferimento alla guida alla risoluzione dei problemi con le integrazioni Lambda, se applicabile.

Scopo: questo allarme è impiegato per rilevare quando le richieste di Gateway API per una route in una fase hanno una latenza elevata.

Statistica: p90

Soglia raccomandata: 2.500,0

Giustificazione della soglia: il valore di soglia raccomandato non funziona per tutti i carichi di lavoro dell'API. Tuttavia, può essere utilizzato come punto di partenza per la soglia. Dopodiché, è possibile scegliere valori di soglia diversi in base al carico di lavoro e ai requisiti di latenza, prestazioni e SLA accettabili per l'API. Se è accettabile che l'API abbia una latenza più elevata in generale, puoi impostare un valore di soglia più alto per rendere l'allarme meno sensibile. Tuttavia, se è necessario che l'API fornisca risposte quasi in tempo reale, imposta un valore di soglia inferiore. Inoltre, è possibile analizzare i dati storici per determinare la latenza di base prevista per il carico di lavoro dell'applicazione e ottimizzare il valore della soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

4xx

Dimensioni:ApiId, Stage

Descrizione dell'allarme: questo allarme rileva un tasso elevato di errori lato client. Ciò può indicare un problema nei parametri di autorizzazione o di richiesta del client. Potrebbe anche significare che una route è stata rimossa o che un client sta richiedendo una route che non esiste nell'API. Valuta la possibilità di abilitare CloudWatch i registri e verificare la presenza di eventuali errori che potrebbero causare gli errori 4xx. Inoltre, valuta la possibilità di abilitare CloudWatch metriche dettagliate per visualizzare questa metrica per percorso, per aiutarti a restringere la fonte degli errori. Gli errori potrebbero essere causati anche dal superamento del limite di limitazione della larghezza di banda della rete impostato. Se le risposte e i log riportano percentuali elevate e impreviste di 429 errori, segui questa guida per risolvere il problema.

Scopo: questo allarme può rilevare tassi elevati di errori lato client per le richieste di Gateway API.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: la soglia raccomandata rileva quando più del 5% del totale delle richieste presenta errori di tipo 4xx. Tuttavia, puoi regolare la soglia in base al traffico relativo alle richieste e ai tassi di errore accettabili. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. È necessario attivare gli allarmi per gli errori 4XX che si verificano frequentemente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

5xx

Dimensioni:ApiId, Stage

Descrizione dell'allarme: questo allarme aiuta a rilevare un tasso elevato di errori lato client. Ciò può indicare un errore nel back-end dell'API, nella rete o nell'integrazione tra il gateway API e l'API di back-end. Questa documentazione può aiutarti a risolvere la causa degli errori 5xx.

Scopo: questo allarme può rilevare tassi elevati di errori lato server per le richieste di Gateway API.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: la soglia raccomandata rileva quando più del 5% del totale delle richieste presenta errori di tipo 5xx. Tuttavia, puoi regolare la soglia in base al traffico relativo alle richieste e ai tassi di errore accettabili. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. È necessario attivare gli allarmi per gli errori 5xx che si verificano frequentemente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: GREATER_THAN_THRESHOLD

MessageCount

Dimensioni:ApiId, Stage

Descrizione dell'allarme: Questo allarme aiuta a rilevare un basso volume di traffico per la fase WebSocket API. Ciò può indicare un problema quando i client chiamano l'API, ad esempio in caso di utilizzo di endpoint errati o problemi con il back-end che invia messaggi ai client. Potrebbe anche indicare un problema relativo alla configurazione o alle autorizzazioni dell'API che la rendono irraggiungibile per i client.

Intento: questo allarme può rilevare un volume di traffico inaspettatamente basso per la WebSocket fase API. Suggeriamo di creare questo allarme se l'API riceve e invia un numero prevedibile e regolare di messaggi in condizioni normali. Se hai abilitato le CloudWatch metriche dettagliate e puoi prevedere il normale volume di traffico per percorso, è meglio creare allarmi alternativi a questo, in modo da avere un monitoraggio più preciso delle cadute di volume di traffico per ogni percorso. Non suggeriamo questo allarme per le API che non prevedono un traffico regolare e costante.

Statistica: SampleCount

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta il valore della soglia in base all'analisi dei dati storici per determinare il numero di messaggi di base previsto per l'API. L'impostazione della soglia su un valore molto alto potrebbe rendere l'allarme troppo sensibile nei periodi di traffico normale e di traffico basso previsto. Viceversa, l'impostazione su un valore molto basso potrebbe far sì che l'allarme non rilevi piccoli cali anomali del volume di traffico.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: LESS_THAN_THRESHOLD

MessageCount

Dimensioni:ApiId, tappa, percorso

Descrizione dell'allarme: questo allarme aiuta a rilevare un basso volume di traffico per il percorso WebSocket API nella fase. Ciò può indicare un problema relativo ai client che chiamano l'API, ad esempio in caso di utilizzo di endpoint errati o problemi con il back-end che invia messaggi ai client. Potrebbe anche indicare un problema relativo alla configurazione o alle autorizzazioni dell'API che la rendono irraggiungibile per i client.

Intento: questo allarme è in grado di rilevare un volume di traffico inaspettatamente basso per il percorso WebSocket API nella fase. Suggeriamo di creare questo allarme se l'API riceve e invia un numero prevedibile e regolare di messaggi in condizioni normali. Non suggeriamo questo allarme per le API che non prevedono un traffico regolare e costante.

Statistica: SampleCount

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base all'analisi dei dati storici per determinare il numero di messaggi di base previsto per l'API. L'impostazione della soglia su un valore molto alto potrebbe rendere l'allarme troppo sensibile nei periodi di traffico normale e di traffico basso previsto. Viceversa, l'impostazione su un valore molto basso potrebbe far sì che l'allarme non rilevi piccoli cali anomali del volume di traffico.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: LESS_THAN_THRESHOLD

ClientError

Dimensioni:ApiId, Stage

Descrizione dell'allarme: questo allarme rileva un tasso elevato di errori dei client. Ciò può indicare un problema nei parametri di autorizzazione o dei messaggi. Potrebbe anche significare che una route è stata rimossa o che un client sta richiedendo una route che non esiste nell'API. Valuta la possibilità di abilitare CloudWatch i registri e verificare la presenza di eventuali errori che potrebbero causare gli errori 4xx. Inoltre, valuta la possibilità di abilitare CloudWatch metriche dettagliate per visualizzare questa metrica per percorso, per aiutarti a restringere la fonte degli errori. Gli errori potrebbero essere causati anche dal superamento del limite di limitazione della larghezza di banda della rete impostato. Se le risposte e i log riportano percentuali elevate e impreviste di 429 errori, segui questa guida per risolvere il problema.

Intento: questo allarme è in grado di rilevare tassi elevati di errori del client per i messaggi WebSocket API Gateway.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: la soglia raccomandata rileva quando più del 5% del totale delle richieste presenta errori di tipo 4xx. È possibile regolare la soglia in base al traffico delle richieste e ai tassi di errore accettabili. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. È necessario attivare gli allarmi per gli errori 4XX che si verificano frequentemente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

ExecutionError

Dimensioni:ApiId, Palco

Descrizione dell'allarme: questo allarme aiuta a rilevare un tasso elevato di errori di esecuzione. Questo potrebbe derivare da errori 5xx generati dall'integrazione, da problemi di autorizzazione o da altri fattori che ostacolano la corretta invocazione dell'integrazione, come la sua rimozione o la limitazione della larghezza di banda della rete associata. Valuta la possibilità di abilitare CloudWatch i log per la tua API e di controllare i log per il tipo e la causa degli errori. Inoltre, valuta la possibilità di abilitare CloudWatch metriche dettagliate per avere una visione di questa metrica per percorso, per aiutarti a restringere la fonte degli errori. Anche questa documentazione può aiutarti a risolvere la causa degli errori di connessione.

Intento: questo allarme può rilevare alti tassi di errori di esecuzione per i messaggi WebSocket API Gateway.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: la soglia suggerita rileva quando più del 5% del totale delle richieste presenta errori di esecuzione. È possibile regolare la soglia in base al traffico delle richieste e ai tassi di errore accettabili. È possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. È necessario attivare gli allarmi per gli errori di esecuzione che si verificano frequentemente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon EC2 Auto Scaling

GroupInServiceCapacity

Dimensioni: AutoScalingGroupName

Descrizione dell'allarme: questo allarme aiuta a rilevare quando la capacità del gruppo è inferiore alla capacità desiderata richiesta per il carico di lavoro. Per risolvere il problema, controlla le

attività di dimensionamento per individuare eventuali errori di avvio e verifica che la configurazione della capacità desiderata sia corretta.

Scopo: questo allarme può rilevare una bassa disponibilità nel gruppo con dimensionamento automatico a causa di errori di avvio o avvii sospesi.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore della soglia deve essere la capacità minima richiesta per eseguire il carico di lavoro. Nella maggior parte dei casi, puoi impostarlo in modo che corrisponda alla GroupDesiredCapacity metrica.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: LESS_THAN_THRESHOLD

Amazon CloudFront

5 xxErrorRate

Dimensioni:DistributionId, Region=Global

Descrizione dell'allarme: questo allarme monitora la percentuale di 5xx risposte di errore dal server di origine, per aiutarti a rilevare se il CloudFront servizio presenta problemi. Consulta la pagina Troubleshooting error responses from your origin per informazioni su come comprendere i problemi del server. Inoltre, attiva parametri aggiuntivi per ottenere parametri dettagliati sugli errori.

Intento: questo allarme viene utilizzato per rilevare problemi relativi alla gestione delle richieste dal server di origine o problemi di comunicazione tra CloudFront e il server di origine.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dalla tolleranza per le risposte 5xx. È possibile analizzare i dati storici e le tendenze e impostare la soglia di conseguenza. Poiché gli errori 5xx possono essere causati da problemi transitori, consigliamo di impostare la soglia su un valore maggiore di 0 in modo che l'allarme non sia troppo sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

OriginLatency

Dimensioni:DistributionId, Region=Global

Descrizione dell'allarme: l'allarme aiuta a monitorare se il server di origine impiega troppo tempo a rispondere. Se il server impiega troppo tempo a rispondere, potrebbe verificarsi un timeout. Consulta la pagina Find and fix delayed responses from applications on your origin server se riscontri valori di `OriginLatency` costantemente elevati.

Scopo: questo allarme viene utilizzato per rilevare problemi con un server di origine che impiega troppo tempo a rispondere.

Statistica: p90

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: la soglia va impostata su un valore pari a circa l'80% del valore di timeout di risposta del server di origine. Se questo parametro è costantemente vicino al valore di timeout di risposta del server di origine, potresti iniziare a riscontrare errori di tipo 504.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

FunctionValidationErrors

Dimensioni:DistributionId, Region=Global FunctionName

Descrizione dell'allarme: questo allarme consente di monitorare gli errori di convalida delle CloudFront funzioni in modo da poter adottare le misure necessarie per risolverli. Analizza i registri delle CloudWatch funzioni e guarda il codice della funzione per trovare e risolvere la causa principale del problema. Vedi Restrizioni sulle funzioni edge per comprendere gli errori di configurazione comuni delle funzioni. CloudFront

Intento: questo allarme viene utilizzato per rilevare gli errori di convalida delle funzioni. CloudFront

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: un valore maggiore di 0 indica un errore di convalida. Si consiglia di impostare la soglia su 0 perché gli errori di convalida implicano un problema quando CloudFront le funzioni vengono restituite a. CloudFront Ad esempio, CloudFront necessita dell'intestazione HTTP Host per elaborare una richiesta. Non c'è nulla che impedisca a un utente di eliminare l'intestazione Host nel codice delle funzioni. CloudFront Ma quando CloudFront riceve la risposta e manca l'intestazione Host, CloudFront genera un errore di convalida.

Periodo: 60

Punti dati su cui attivare allarmi: 2

Periodi di valutazione: 2

Operatore di confronto: GREATER_THAN_THRESHOLD

FunctionExecutionErrors

Dimensioni:DistributionId,, Region=Global FunctionName

Descrizione dell'allarme: Questo allarme consente di monitorare gli errori di esecuzione CloudFront delle funzioni in modo da poter adottare le misure necessarie per risolverli. Analizza i log delle CloudWatch funzioni e guarda il codice della funzione per trovare e risolvere la causa principale del problema.

Intento: questo allarme viene utilizzato per rilevare gli errori di esecuzione delle funzioni. CloudFront

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: si consiglia di impostare la soglia su 0 perché un errore di esecuzione indica un problema con il codice che si verifica in fase di runtime.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

FunctionThrottles

Dimensioni:DistributionId,, FunctionName Region=Global

Descrizione dell'allarme: Questo allarme ti aiuta a monitorare se la tua CloudFront funzione è limitata. Se la tua funzione è sottoposta a limitazione della larghezza di banda della rete, significa che l'esecuzione sta impiegando troppo tempo. Per evitare limitazioni della larghezza di banda della rete nelle funzioni, valuta la possibilità di ottimizzarne il codice.

Scopo: questo allarme è in grado di rilevare quando la CloudFront funzione è limitata, in modo da consentire all'utente di reagire e risolvere il problema per un'esperienza utente senza intoppi.

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: si consiglia di impostare la soglia su 0, per consentire una risoluzione più rapida delle accelerazioni delle funzioni.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon Cognito

SignUpThrottles

Dimensioni:, UserPool UserPoolClient

Descrizione dell'allarme: questo allarme monitora il numero di richieste sottoposte a limitazione della larghezza di banda della rete. Se gli utenti vengono costantemente sottoposti a limitazione della larghezza di banda della rete, è necessario aumentare il limite richiedendo un aumento della quota di servizio. Consulta la pagina Quotas in Amazon Cognito per informazioni su come richiedere un aumento delle quote. Per intraprendere operazioni in modo proattivo, prendi in considerazione la possibilità di tracciare la quota di utilizzo.

Scopo: questo allarme aiuta a monitorare se si verificano richieste di registrazione sottoposte a limitazione della larghezza di banda della rete. Questo può aiutarti a sapere quando intraprendere operazioni per mitigare qualsiasi deterioramento dell'esperienza di registrazione. Una limitazione prolungata della larghezza di banda della rete per le richieste incide negativamente sull'esperienza di registrazione degli utenti.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: un pool di utenti con provisioning adeguato non dovrebbe subire alcuna limitazione della larghezza di banda della rete su più punti dati. Pertanto, la soglia tipica per un carico di lavoro previsto dovrebbe essere zero. Per un carico di lavoro irregolare con interruzioni frequenti, è possibile analizzare i dati storici per determinare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. Per ridurre al minimo l'impatto sull'applicazione, è necessario ritentare una richiesta sottoposta a limitazione della larghezza di banda della rete.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

SignInThrottles

Dimensioni:UserPool, UserPoolClient

Descrizione dell'allarme: questo allarme monitora il numero di richieste di autenticazione utente sottoposte a limitazione della larghezza di banda della rete. Se gli utenti vengono costantemente sottoposti a limitazione della larghezza di banda della rete, potrebbe essere necessario

aumentare il limite richiedendo un aumento della quota di servizio. Consulta la pagina Quotas in Amazon Cognito per informazioni su come richiedere un aumento delle quote. Per intraprendere operazioni in modo proattivo, prendi in considerazione la possibilità di tracciare la quota di utilizzo.

Scopo: questo allarme aiuta a monitorare se si verificano richieste di accesso sottoposte a limitazione della larghezza di banda della rete. Questo può aiutarti a sapere quando intraprendere operazioni per mitigare qualsiasi deterioramento dell'esperienza di accesso. Una limitazione prolungata della larghezza di banda della rete per le richieste è un'esperienza di autenticazione utente negativa.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: un pool di utenti con provisioning adeguato non dovrebbe subire alcuna limitazione della larghezza di banda della rete su più punti dati. Pertanto, la soglia tipica per un carico di lavoro previsto dovrebbe essere zero. Per un carico di lavoro irregolare con interruzioni frequenti, è possibile analizzare i dati storici per determinare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. Per ridurre al minimo l'impatto sull'applicazione, è necessario ritentare una richiesta sottoposta a limitazione della larghezza di banda della rete.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

TokenRefreshThrottles

Dimensioni:UserPool, UserPoolClient

Descrizione dell'allarme: è possibile impostare il valore di soglia in base al traffico della richiesta e in modo che corrisponda a una limitazione della larghezza di banda della rete accettabile per le richieste di aggiornamento dei token. La limitazione della larghezza di banda della rete viene utilizzata per proteggere il sistema da un numero eccessivo di richieste. Tuttavia, è importante monitorare anche se le risorse di cui si dispone sono insufficienti per il normale traffico. È possibile analizzare i dati storici per individuare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro dell'applicazione e regolare la soglia di allarme in modo che sia superiore al

livello di limitazione accettabile. Le richieste soggette a limitazione della larghezza di banda della rete devono essere ritentate dall'applicazione/dal servizio poiché sono transitorie. Pertanto, un valore di soglia molto basso può rendere l'allarme sensibile.

Scopo: questo allarme aiuta a monitorare se si verificano richieste di aggiornamento dei token sottoposte a limitazione della larghezza di banda della rete. Questo può aiutarti a sapere quando intraprendere operazioni per mitigare eventuali problemi, al fine di garantire un'esperienza utente fluida unita all'integrità e all'affidabilità del tuo sistema di autenticazione. Una limitazione prolungata della larghezza di banda della rete per le richieste è un'esperienza di autenticazione utente negativa.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia può essere impostato/regolato anche in base al traffico della richiesta, nonché a una limitazione della larghezza di banda della rete accettabile per le richieste di aggiornamento dei token. La limitazione della larghezza di banda della rete serve a proteggere il sistema da un numero eccessivo di richieste, tuttavia è importante monitorare anche che le risorse per il traffico normale non siano insufficienti e controllare se è proprio questo a causare l'impatto. È possibile analizzare i dati storici per individuare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro dell'applicazione e regolare la soglia in modo che sia superiore al livello di limitazione accettabile abituale. Le richieste soggette a limitazione della larghezza di banda della rete devono essere ritentate dall'applicazione/dal servizio poiché sono transitorie. Pertanto, un valore di soglia molto basso può rendere l'allarme sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

FederationThrottles

Dimensioni:UserPool, UserPoolClient, IdentityProvider

Descrizione dell'allarme: questo allarme monitora il numero di richieste di federazione delle identità sottoposte a limitazione della larghezza di banda della rete. Se si riscontra una continua limitazione della larghezza di banda della rete, potrebbe essere necessario aumentare il limite

richiedendo un aumento della quota di servizio. Consulta la pagina Quotas in Amazon Cognito per informazioni su come richiedere un aumento delle quote.

Scopo: questo allarme aiuta a monitorare se si verificano richieste di federazione delle identità sottoposte a limitazione della larghezza di banda della rete. Questo può aiutarti a reagire in modo proattivo ai colli di bottiglia in termini di prestazioni o alle configurazioni errate e a garantire un'esperienza di autenticazione fluida per gli utenti. Una limitazione prolungata della larghezza di banda della rete per le richieste è un'esperienza di autenticazione utente negativa.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: è possibile impostare la soglia in base al traffico della richiesta e in modo che corrisponda alla limitazione della larghezza di banda della rete accettabile per le richieste di federazione delle identità. La limitazione della larghezza di banda della rete viene utilizzata per proteggere il sistema da un numero eccessivo di richieste. Tuttavia, è importante monitorare anche se le risorse di cui si dispone sono insufficienti per il normale traffico. È possibile analizzare i dati storici per individuare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro dell'applicazione e regolare il valore della soglia in modo che sia superiore al livello di limitazione accettabile. Le richieste soggette a limitazione della larghezza di banda della rete devono essere ritentate dall'applicazione/dal servizio poiché sono transitorie. Pertanto, un valore di soglia molto basso può rendere l'allarme sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon DynamoDB

AccountProvisionedReadCapacityUtilization

Dimensions: nessuna

Descrizione dell'allarme: questo allarme rileva se la capacità di lettura dell'account sta raggiungendo il limite previsto. In tal caso, è possibile aumentare la quota dell'account per l'utilizzo

della capacità di lettura. È possibile visualizzare le quote correnti per le unità di capacità di lettura e richiedere eventuali aumenti utilizzando Service Quotas.

Scopo: l'allarme può rilevare se l'utilizzo della capacità di lettura dell'account si avvicina all'utilizzo della capacità di lettura assegnata. Se l'utilizzo raggiunge il limite massimo, DynamoDB inizia a limitare la larghezza di banda della rete per le richieste di lettura.

Statistica: Maximum

Soglia raccomandata: 80,0

Giustificazione della soglia: imposta la soglia all'80%, in modo che sia possibile intraprendere operazioni (ad esempio innalzare i limiti dell'account) prima che raggiunga la piena capacità al fine di evitare una limitazione della larghezza di banda della rete.

Periodo: 300

Punti dati su cui attivare allarmi: 2

Periodi di valutazione: 2

Operatore di confronto: GREATER_THAN_THRESHOLD

AccountProvisionedWriteCapacityUtilization

Dimensions: nessuna

Descrizione dell'allarme: questo allarme rileva se la capacità di scrittura dell'account sta raggiungendo il limite previsto. In tal caso, è possibile aumentare la quota dell'account per l'utilizzo della capacità di scrittura. È possibile visualizzare le quote correnti per le unità di capacità di scrittura e richiedere eventuali aumenti utilizzando Service Quotas.

Scopo: questo allarme può rilevare se l'utilizzo della capacità di scrittura dell'account si avvicina all'utilizzo della capacità di scrittura assegnata. Se l'utilizzo raggiunge il limite massimo, DynamoDB inizia a limitare la larghezza di banda della rete per le richieste di scrittura.

Statistica: Maximum

Soglia raccomandata: 80,0

Giustificazione della soglia: imposta la soglia all'80%, in modo che sia possibile intraprendere operazioni (ad esempio innalzare i limiti dell'account) prima che raggiunga la piena capacità al fine di evitare una limitazione della larghezza di banda della rete.

Periodo: 300

Punti dati su cui attivare allarmi: 2

Periodi di valutazione: 2

Operatore di confronto: GREATER_THAN_THRESHOLD

AgeOfOldestUnreplicatedRecord

Dimensioni:TableName, DelegatedOperation

Descrizione dell'allarme: questo allarme rileva il ritardo nella replica in un flusso di dati Kinesis. In condizioni di funzionamento normale, `AgeOfOldestUnreplicatedRecord` dovrebbe essere di pochi millisecondi. Questo numero aumenta in base ai tentativi di replica non riusciti causati da scelte di configurazione controllate dal cliente. Gli esempi di configurazioni controllate dal cliente che portano a tentativi di replica non riusciti sono una capacità del flusso di dati Kinesis con provisioning insufficiente, che comporta una limitazione della larghezza di banda della rete eccessiva, o un aggiornamento manuale delle policy di accesso del flusso di dati Kinesis, che impedisce a DynamoDB di aggiungere dati al flusso dei dati. Per mantenere questo parametro il più basso possibile, potrebbe essere necessario garantire il corretto provisioning della capacità del flusso di dati Kinesis e assicurarsi che le autorizzazioni di DynamoDB siano invariate.

Scopo: questo allarme può monitorare i tentativi di replica non riusciti e il conseguente ritardo nella replica nel flusso di dati Kinesis.

Statistica: Maximum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base al ritardo di replica desiderato misurato in millisecondi. Questo valore dipende dai requisiti del carico di lavoro e dalle prestazioni previste.

Periodo: 300

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: GREATER_THAN_THRESHOLD

FailedToReplicateRecordCount

Dimensioni:TableName, DelegatedOperation

Descrizione dell'allarme: il numero di record che DynamoDB non è riuscito a replicare nel flusso di dati Kinesis. Alcuni elementi di dimensioni superiori a 34 KB potrebbero espandersi per modificare i record di dati superiori al limite di dimensioni di 1 MB degli elementi di Kinesis Data Streams. Questo aumento delle dimensioni si verifica quando gli elementi più grandi di 34 KB includono un numero elevato di valori booleani o vuoti degli attributi. I valori booleani e vuoti degli attributi vengono archiviati come 1 byte in DynamoDB, ma si espandono fino a 5 byte quando vengono serializzati utilizzando JSON standard per la replica di Kinesis Data Streams. DynamoDB non può replicare tali record di modifica nel flusso dei dati Kinesis. DynamoDB ignora questi record di dati di modifica e continua automaticamente a replicare i record successivi.

Scopo: questo allarme può monitorare il numero di record che DynamoDB non è riuscito a replicare nel flusso di dati Kinesis a causa del limite delle dimensioni degli elementi dei flussi di dati Kinesis.

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: imposta la soglia su 0 per rilevare eventuali record che DynamoDB non è riuscito a replicare.

Periodo: 60

Punti dati su cui attivare allarmi: 1

Periodi di valutazione: 1

Operatore di confronto: GREATER_THAN_THRESHOLD

ReadThrottleEvents

Dimensioni: TableName

Descrizione dell'allarme: questo allarme rileva la presenza di un numero elevato di richieste di lettura sottoposte a limitazione della larghezza di banda della rete per la tabella DynamoDB. Per risolvere il problema, consulta la pagina Troubleshooting throttling issues in Amazon DynamoDB.

Scopo: questo allarme può rilevare una limitazione della larghezza di banda della rete prolungata delle richieste di lettura alla tabella DynamoDB. Una limitazione prolungata della larghezza di banda della rete per le richieste di lettura può influire negativamente sulle operazioni di lettura del carico di lavoro e ridurre l'efficienza complessiva del sistema.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base al traffico di lettura previsto per la tabella DynamoDB, tenendo conto di un livello accettabile di limitazione della larghezza di banda della rete. È importante verificare di disporre di un numero sufficiente di risorse e che non si verifichino limitazioni della larghezza di banda della rete costanti. Inoltre, è possibile analizzare i dati storici per individuare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro dell'applicazione e regolare la soglia in modo che sia superiore al livello di limitazione accettabile abituale. Le richieste soggette a limitazione della larghezza di banda della rete devono essere ritentate dall'applicazione/dal servizio poiché sono transitorie. Pertanto, una soglia molto bassa potrebbe rendere l'allarme troppo sensibile, causando transizioni di stato indesiderate.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

ReadThrottleEvents

Dimensioni:TableName, GlobalSecondaryIndexName

Descrizione dell'allarme: questo allarme rileva la presenza di un numero elevato di richieste di lettura sottoposte a limitazione della larghezza di banda della rete per l'indice secondario globale della tabella DynamoDB. Per risolvere il problema, consulta la pagina Troubleshooting throttling issues in Amazon DynamoDB.

Scopo: l'allarme può rilevare una limitazione della larghezza di banda della rete prolungata delle richieste di lettura per l'indice secondario globale della tabella DynamoDB. Una limitazione prolungata della larghezza di banda della rete per le richieste di lettura può influire negativamente sulle operazioni di lettura del carico di lavoro e ridurre l'efficienza complessiva del sistema.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base al traffico di lettura previsto per la tabella DynamoDB, tenendo conto di un livello accettabile di limitazione della larghezza di banda della

rete. È importante verificare di disporre di un numero sufficiente di risorse e che non si verifichino limitazioni della larghezza di banda della rete costanti. Inoltre, è possibile analizzare i dati storici per individuare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro dell'applicazione e regolare la soglia in modo che sia superiore al livello di limitazione accettabile abituale. Le richieste soggette a limitazione della larghezza di banda della rete devono essere ritentate dall'applicazione/dal servizio poiché sono transitorie. Pertanto, una soglia molto bassa potrebbe rendere l'allarme troppo sensibile, causando transizioni di stato indesiderate.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

ReplicationLatency

Dimensioni:TableName, ReceivingRegion

Descrizione dell'allarme: l'allarme rileva se la replica della tabella globale in una regione è in ritardo rispetto alla regione di origine. La latenza può aumentare se una AWS regione si deteriora e in quella regione è presente una tabella di replica. In questo caso, puoi reindirizzare temporaneamente l'attività di lettura e scrittura dell'applicazione verso un'altra regione. AWS Se si utilizza la versione 2017.11.29 (legacy) delle tabelle globali, è necessario verificare che le unità di capacità di scrittura (WCU) siano identiche per ciascuna delle tabelle di replica. Puoi anche assicurarti di seguire le raccomandazioni elencate nella pagina Best practices and requirements for managing capacity.

Scopo: l'allarme può rilevare se la tabella di replica in una regione è in ritardo rispetto alla replica delle modifiche da un'altra regione. Ciò potrebbe far sì che la replica diverga dalle altre repliche. È utile conoscere la latenza di replica di ciascuna AWS regione e avvisare se tale latenza di replica aumenta continuamente. La replica della tabella si applica solo alle tabelle globali.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dal caso d'uso. Solitamente, sono oggetto di indagine le latenze di replica superiori a 3

minuti. Esamina la criticità e i requisiti del ritardo di replica e analizza le tendenze storiche, quindi seleziona la soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

SuccessfulRequestLatency

Dimensioni:, Funzionamento TableName

Descrizione dell'allarme: questo allarme rileva un'elevata latenza per il funzionamento della tabella DynamoDB (indicata dal valore della dimensione `Operation`nell'allarme). Consulta questo documento sulla risoluzione dei problemi per risolvere i problemi di latenza in Amazon DynamoDB.

Scopo: questo allarme può rilevare una latenza elevata delle operazioni della tabella DynamoDB. Una latenza delle operazioni più elevata può influire negativamente sull'efficienza complessiva del sistema.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: DynamoDB fornisce una latenza media di un millisecondo per operazioni singleton come, e così via. GetItem PutItem Tuttavia, puoi impostare la soglia in base alla tolleranza accettabile per la latenza, considerando il tipo di operazione e la tabella coinvolta nel carico di lavoro. È possibile analizzare i dati storici di questo parametro per individuare la latenza abituale per l'operazione sulla tabella e quindi impostare la soglia su un numero che rappresenti il ritardo critico per l'operazione.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_THRESHOLD

SystemErrors

Dimensioni: TableName

Descrizione dell'allarme: questo allarme rileva un numero elevato e prolungato di errori di sistema per le richieste delle tabelle DynamoDB. Se continui a ricevere errori 5xx, apri il Pannello di controllo per lo stato dei servizi AWS per verificare la presenza di problemi operativi con il servizio. Puoi utilizzare questo allarme per ricevere una notifica da DynamoDB in caso di un problema prolungato del servizio interno, aiutandoti a stabilire un collegamento con il problema che l'applicazione client sta riscontrando. Per ulteriori informazioni, consulta la pagina Error handling for DynamoDB.

Scopo: questo allarme può rilevare errori di sistema prolungati nelle richieste della tabella DynamoDB. Gli errori di sistema indicano errori interni del servizio di DynamoDB e contribuiscono a stabilire un collegamento con il problema riscontrato dal client.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base al traffico di lettura previsto, tenendo conto di un livello accettabile di errori di sistema. Inoltre, è possibile analizzare i dati storici per determinare il numero di errori accettabili per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. Gli errori di sistema devono essere ritentati dall'applicazione/dal servizio poiché sono transitori. Pertanto, una soglia molto bassa potrebbe rendere l'allarme troppo sensibile, causando transizioni di stato indesiderate.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

ThrottledPutRecordCount

Dimensioni:TableName, DelegatedOperation

Descrizione dell'allarme: questo allarme rileva i record sottoposti a limitazione della larghezza di banda della rete dal flusso di dati Kinesis durante la replica dell'acquisizione dei dati di modifica

su Kinesis. Questa limitazione della larghezza di banda della rete si verifica a causa della capacità del flusso di dati Kinesis insufficiente. Se si verifica una limitazione eccessiva e regolare, potrebbe essere necessario aumentare il numero di partizioni del flusso Kinesis proporzionalmente alla velocità di scrittura osservata della tabella. Per ulteriori informazioni su come determinare le dimensioni di un flusso di dati Kinesis, consulta Determinazione delle dimensioni iniziali di un flusso di dati Kinesis.

Scopo: questo allarme può monitorare il numero di record sottoposti a limitazione della larghezza di banda della rete dal flusso di dati Kinesis a causa della capacità insufficiente di Kinesis Data Streams.

Statistica: Maximum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: è possibile che si verifichino delle limitazioni della larghezza di banda della rete durante i picchi di utilizzo eccezionali, ma i record limitati devono rimanere il più bassi possibile per evitare una maggiore latenza di replica (DynamoDB riprova a inviare i record limitati al flusso di dati Kinesis). Imposta la soglia su un numero che possa aiutarti a rilevare una limitazione della larghezza di banda della rete regolarmente eccessiva. Inoltre, è possibile analizzare i dati storici di questo parametro per individuare i tassi di limitazione della larghezza di banda della rete accettabili per il carico di lavoro dell'applicazione. Regola la soglia su un valore che l'applicazione può tollerare in base al tuo caso d'uso.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_THRESHOLD

UserErrors

Dimensions: nessuna

Descrizione dell'allarme: questo allarme rileva un numero elevato e prolungato di errori utente per le richieste delle tabelle DynamoDB. È possibile controllare i log delle applicazioni client durante il periodo del problema per vedere perché le richieste non sono valide. Puoi controllare il codice di stato HTTP 400 per vedere il tipo di errore che stai riscontrando e agire di conseguenza. Potrebbe essere necessario correggere la logica dell'applicazione per creare richieste valide.

Scopo: questo allarme può rilevare errori degli utenti prolungati nelle richieste della tabella DynamoDB. Gli errori degli utenti nelle operazioni richieste indicano che il client sta producendo richieste non valide e che l'operazione non va a buon fine.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia su zero per rilevare eventuali errori sul lato client. Oppure impostalo su un valore più alto se vuoi evitare che si attivi l'allarme per un numero molto basso di errori. Decidi in base al tuo caso d'uso e al traffico per le richieste.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_THRESHOLD

WriteThrottleEvents

Dimensioni: TableName

Descrizione dell'allarme: questo allarme rileva la presenza di un numero elevato di richieste di scrittura sottoposte a limitazione della larghezza di banda della rete per la tabella DynamoDB. Per risolvere il problema, consulta la pagina Troubleshooting throttling issues in Amazon DynamoDB.

Scopo: questo allarme può rilevare una limitazione della larghezza di banda della rete prolungata delle richieste di scrittura alla tabella DynamoDB. Una limitazione prolungata della larghezza di banda della rete per le richieste di scrittura può influire negativamente sulle operazioni di scrittura del carico di lavoro e ridurre l'efficienza complessiva del sistema.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base al traffico di scrittura previsto per la tabella DynamoDB, tenendo conto di un livello accettabile di limitazione della larghezza di banda della rete. È importante verificare di disporre di un numero sufficiente di risorse e che non si verifichino limitazioni della larghezza di banda della rete costanti. Inoltre, è possibile analizzare i dati storici per individuare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro

dell'applicazione e regolare la soglia su un valore superiore al livello di limitazione accettabile abituale. Le richieste soggette a limitazione della larghezza di banda della rete devono essere ritentate dall'applicazione/dal servizio poiché sono transitorie. Pertanto, una soglia molto bassa potrebbe rendere l'allarme troppo sensibile, causando transizioni di stato indesiderate.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

WriteThrottleEvents

Dimensioni:TableName, GlobalSecondaryIndexName

Descrizione dell'allarme: questo allarme rileva la presenza di un numero elevato di richieste di scrittura sottoposte a limitazione della larghezza di banda della rete per l'indice secondario globale della tabella DynamoDB. Per risolvere il problema, consulta la pagina Troubleshooting throttling issues in Amazon DynamoDB.

Scopo: questo allarme può rilevare una limitazione della larghezza di banda della rete prolungata delle richieste di scrittura per l'indice secondario globale della tabella DynamoDB. Una limitazione prolungata della larghezza di banda della rete per le richieste di scrittura può influire negativamente sulle operazioni di scrittura del carico di lavoro e ridurre l'efficienza complessiva del sistema.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base al traffico di scrittura previsto per la tabella DynamoDB, tenendo conto di un livello accettabile di limitazione della larghezza di banda della rete. È importante verificare di disporre di un numero sufficiente di risorse e che non si verifichino limitazioni della larghezza di banda della rete costanti. Inoltre, è possibile analizzare i dati storici per individuare la limitazione della larghezza di banda della rete accettabile per il carico di lavoro dell'applicazione e regolare la soglia su un valore superiore al livello di limitazione accettabile abituale. Le richieste soggette a limitazione della larghezza di banda della rete devono essere ritentate dall'applicazione/dal servizio poiché sono transitorie. Pertanto, un valore molto basso potrebbe rendere l'allarme troppo sensibile, causando transizioni di stato indesiderate.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon EBS

VolumeStalledIOCheck

Dimensioni:VolumeId, InstanceId

Descrizione dell'allarme: questo allarme consente di monitorare le prestazioni di I/O dei volumi Amazon EBS. Questo controllo rileva problemi di fondo con l'infrastruttura Amazon EBS, come problemi hardware o software sui sottosistemi di storage alla base dei volumi Amazon EBS, problemi hardware sull'host fisico che influiscono sulla raggiungibilità dei volumi Amazon EBS dall'istanza Amazon EC2, e può rilevare problemi di connettività tra l'istanza e i volumi Amazon EBS. Se lo Stalled IO Check fallisce, puoi AWS attendere la risoluzione del problema oppure puoi intraprendere azioni come sostituire il volume interessato o arrestare e riavviare l'istanza a cui è collegato il volume. Nella maggior parte dei casi, quando questa metrica fallisce, Amazon EBS diagnostica e ripristina automaticamente il volume entro pochi minuti.

Intento: questo allarme è in grado di rilevare lo stato dei volumi Amazon EBS per determinare quando tali volumi sono compromessi e non possono completare le operazioni di I/O.

Statistica: Maximum

Soglia raccomandata: 1,0

Giustificazione della soglia: quando una verifica dello stato non riesce, il valore di questo parametro è 1. La soglia è impostata in modo tale che ogni volta che il controllo dello stato non riesce, l'allarme sia in stato ALARM.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EC2

CPUUtilization

Dimensioni: InstanceId

Descrizione dell'allarme: questo allarme aiuta a monitorare l'utilizzo della CPU di un'istanza EC2. A seconda dell'applicazione, potrebbero essere normali livelli di utilizzo costantemente elevati. Tuttavia, se le prestazioni si deteriorano e l'applicazione non è limitata dall'I/O del disco, dalla memoria o dalle risorse di rete, una CPU al massimo potrebbe indicare un collo di bottiglia in termini di risorse o problemi di prestazioni delle applicazioni. Un utilizzo elevato della CPU potrebbe indicare che è necessario un aggiornamento a un'istanza con un uso più intensivo della CPU. Se è abilitato il monitoraggio dettagliato, è possibile modificare il periodo a 60 secondi anziché 300 secondi. Per ulteriori informazioni, consulta la pagina Enable or turn off detailed monitoring for your instances.

Scopo: questo allarme viene utilizzato per rilevare un elevato utilizzo della CPU.

Statistica: Average

Soglia raccomandata: 80,0

Giustificazione della soglia: in genere, è possibile impostare la soglia per l'utilizzo della CPU al 70-80%. Tuttavia, puoi regolare questo valore in base al livello di prestazioni e alle caratteristiche del carico di lavoro accettabili. Per alcuni sistemi, un utilizzo costantemente elevato della CPU può essere normale e non indicare un problema, mentre per altri può essere motivo di preoccupazione. Analizza i dati storici sull'utilizzo della CPU per stabilire qual è un valore di utilizzo della CPU accettabile per il sistema e imposta la soglia di conseguenza.

Periodo: 300

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: GREATER_THAN_THRESHOLD

StatusCheckFailed

Dimensioni: InstanceId

Descrizione dell'allarme: questo allarme aiuta a monitorare sia i controlli dello stato del sistema che i controlli dello stato dell'istanza. Se uno dei due tipi di controllo dello stato fallisce, questo allarme dovrebbe essere nello stato ALARM.

Scopo: questo allarme viene utilizzato per rilevare i problemi sottostanti delle istanze, inclusi gli errori di controllo dello stato del sistema e gli errori di controllo dello stato delle istanze.

Statistica: Maximum

Soglia raccomandata: 1,0

Giustificazione della soglia: quando una verifica dello stato non riesce, il valore di questo parametro è 1. La soglia è impostata in modo tale che ogni volta che il controllo dello stato non riesce, l'allarme sia in stato ALARM.

Periodo: 300

Punti dati su cui attivare allarmi: 2

Periodi di valutazione: 2

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

StatusCheckFailed_attachedEBS

Dimensioni: InstanceId

Descrizione dell'allarme: questo allarme consente di monitorare se i volumi Amazon EBS collegati a un'istanza sono raggiungibili e in grado di completare le operazioni di I/O. Questo controllo dello stato rileva problemi di fondo con l'infrastruttura di elaborazione o Amazon EBS, come i seguenti:

- Problemi hardware o software sui sottosistemi di storage alla base dei volumi Amazon EBS
- Problemi hardware sull'host fisico che influiscono sulla raggiungibilità dei volumi Amazon EBS
- Problemi di connettività tra l'istanza e i volumi Amazon EBS

Quando il controllo dello stato EBS allegato fallisce, puoi attendere che Amazon risolva il problema oppure puoi intraprendere un'azione come sostituire i volumi interessati o arrestare e riavviare l'istanza.

Intento: questo allarme viene utilizzato per rilevare volumi Amazon EBS non raggiungibili collegati a un'istanza. Questi possono causare guasti nelle operazioni di I/O.

Statistica: Maximum

Soglia raccomandata: 1,0

Giustificazione della soglia: quando una verifica dello stato non riesce, il valore di questo parametro è 1. La soglia è impostata in modo tale che ogni volta che il controllo dello stato non riesce, l'allarme sia in stato ALARM.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon ElastiCache

CPUUtilization

Dimensioni:, CacheClusterId CacheNodeId

Descrizione dell'allarme: questo allarme aiuta a monitorare l'utilizzo della CPU per l'intera ElastiCache istanza, inclusi i processi del motore di database e altri processi in esecuzione sull'istanza. AWS Elasticache supporta due tipi di motore: Memcached e Redis. Se raggiungi un utilizzo della CPU elevato su un nodo Memcached, dovresti prendere in considerazione la possibilità di aumentare il tipo di istanza o aggiungere nuovi nodi di cache. Per Redis, se il carico di lavoro principale riguarda le richieste di lettura, dovresti prendere in considerazione l'aggiunta di altre repliche di lettura al cluster di cache. Se il tuo carico di lavoro principale è costituito da richieste di scrittura, dovresti prendere in considerazione l'aggiunta di altre partizioni per distribuire il carico di lavoro su più nodi primari (se utilizzi la modalità cluster) o l'aumento del tipo di istanza (se esegui Redis in modalità non cluster).

Intento: questo allarme viene utilizzato per rilevare un elevato utilizzo della CPU da parte degli host. ElastiCache È utile avere una visione generale dell'utilizzo della CPU nell'intera istanza, compresi i processi non legati al motore.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia sulla percentuale che riflette un livello di utilizzo della CPU critico per l'applicazione. Per Memcached, il motore può utilizzare fino a num_threads core. Per Redis, il motore è principalmente a thread singolo, ma potrebbe utilizzare core aggiuntivi, se disponibili, per accelerare l'I/O. Nella maggior parte dei casi, è possibile impostare la soglia su circa il 90% della CPU disponibile. Poiché Redis è a thread singolo, il valore di soglia effettivo deve essere calcolato come una frazione della capacità totale del nodo.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

CurrConnections

Dimensioni:, CacheClusterId CacheNodeId

Descrizione dell'allarme: questo allarme rileva un numero elevato di connessioni, che potrebbe indicare problemi di carico o prestazioni elevati. Un aumento costante di `CurrConnections` potrebbe portare all'esaurimento delle 65.000 connessioni disponibili. Potrebbe indicare che le connessioni sono state chiuse erroneamente sul lato dell'applicazione e che sul lato server sono rimaste stabilite. È consigliabile utilizzare il pool di connessioni o i timeout di connessione inattivi per limitare il numero di connessioni effettuate al cluster oppure, per Redis, valutare la possibilità di ottimizzare tcp-keepalive sul cluster per rilevare e terminare potenziali peer morti.

Intento: l'allarme consente di identificare un numero elevato di connessioni che potrebbero influire sulle prestazioni e sulla stabilità del ElastiCache cluster.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dall'intervallo di connessioni accettabile per il cluster. Esamina la capacità e il carico di lavoro previsto del ElastiCache cluster e analizza il numero storico delle connessioni durante l'uso regolare per stabilire una linea di base, quindi seleziona una soglia di conseguenza. Ricorda che ogni nodo può supportare fino a 65.000 connessioni simultanee.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_THRESHOLD

DatabaseMemoryUsagePercentage

Dimensioni: CacheClusterId

Descrizione dell'allarme: questo allarme consente di monitorare l'utilizzo della memoria del cluster. Quando il valore `DatabaseMemoryUsagePercentage` raggiunge il 100%, viene attivata la policy Redis maxmemory e potrebbero verificarsi espulsioni in base alla policy selezionata. Se nessun oggetto nella cache corrisponde alla policy di espulsione, le operazioni di scrittura hanno esito negativo. Alcuni carichi di lavoro prevedono espulsioni o si basano su di esse, ma in caso contrario sarà necessario aumentare la capacità di memoria del cluster. È possibile dimensionare il cluster aggiungendo altri nodi primari oppure aumentarlo utilizzando un tipo di nodo più grande. Per ulteriori informazioni, consulta Scaling ElastiCache for Redis clusters.

Scopo: questo allarme viene utilizzato per rilevare un elevato utilizzo della memoria del cluster in modo da evitare errori durante la scrittura sul cluster. È utile sapere quando sarà necessario aumentare il cluster se per l'applicazione non sono previste espulsioni.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: a seconda dei requisiti di memoria dell'applicazione e della capacità di memoria del ElastiCache cluster, è necessario impostare la soglia sulla percentuale che riflette il livello critico di utilizzo della memoria del cluster. È possibile utilizzare i dati storici sull'utilizzo della memoria come indicazione per la soglia di utilizzo della memoria accettabile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

EngineCPUUtilization

Dimensioni: CacheClusterId

Descrizione dell'allarme: questo allarme aiuta a monitorare l'utilizzo della CPU di un thread del motore Redis all'interno dell' ElastiCache istanza. I motivi più comuni dell'utilizzo della CPU elevato da parte di un motore sono i comandi a esecuzione prolungata con un utilizzo elevato di CPU, un numero elevato di richieste, l'aumento delle nuove richieste di connessione client in un breve periodo di tempo e un numero elevato di espulsioni quando la cache non dispone di memoria sufficiente per contenere nuovi dati. Dovresti prendere in considerazione la possibilità di scalare ElastiCache i cluster Redis aggiungendo più nodi o aumentando il tipo di istanza.

Scopo: questo allarme viene utilizzato per rilevare un elevato utilizzo della CPU da parte del thread del motore Redis. È utile se si desidera monitorare l'utilizzo della CPU del motore di database stesso.

Statistica: Average

Soglia raccomandata: 90,0

Giustificazione della soglia: imposta la soglia su una percentuale che rifletta il livello di utilizzo della CPU del motore critico per l'applicazione. È possibile eseguire il benchmark del cluster utilizzando l'applicazione e il carico di lavoro previsto per stabilire un collegamento tra EngineCPUUtilization e le prestazioni di riferimento, quindi impostare la soglia di conseguenza. Nella maggior parte dei casi, è possibile impostare la soglia su circa il 90% della CPU disponibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

ReplicationLag

Dimensioni: CacheClusterId

Descrizione dell'allarme: questo allarme aiuta a monitorare lo stato della replica del ElastiCache cluster. Un ritardo di replica elevato significa che il nodo primario o la replica non sono in grado di mantenere il ritmo della replica. Se l'attività di scrittura è troppo elevata, valuta la possibilità di dimensionare il cluster aggiungendo altri nodi primari oppure di aumentarlo utilizzando un tipo di nodo più grande. Per i dettagli, consulta Scaling ElastiCache for Redis clusters. Se le repliche di lettura sono sovraccaricate dalla quantità di richieste di lettura, valuta la possibilità di aggiungere altre repliche di lettura.

Scopo: questo allarme viene utilizzato per rilevare un ritardo tra l'aggiornamento dei dati sul nodo primario e la loro sincronizzazione con il nodo di replica. Contribuisce a garantire la coerenza dei dati di un nodo cluster di replica in lettura.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base ai requisiti dell'applicazione e al potenziale impatto del ritardo di replica. È necessario considerare le velocità di scrittura e le condizioni di rete previste dell'applicazione per stabilire un ritardo di replica accettabile.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon EC2 (**`AWS/ElasticGPUs`**)

GPU ConnectivityCheckFailed

Dimensioni:InstanceId, ePUID

Descrizione dell'allarme: questo allarme aiuta a rilevare gli errori di connessione tra l'istanza e l'acceleratore Elastic Graphics. Grafica elastica utilizza la rete di istanze per inviare comandi OpenGL a una scheda grafica collegata in remoto. Inoltre, un desktop che esegue un'applicazione OpenGL con un acceleratore Elastic Graphics è in genere accessibile utilizzando una tecnologia di accesso remoto. È importante distinguere tra un problema di prestazioni relativo al rendering di OpenGL e la tecnologia di accesso remoto del desktop. Per ulteriori informazioni sul problema, consulta la pagina Investigate application performance issues.

Scopo: questo allarme viene utilizzato per rilevare i problemi di connettività dall'istanza all'acceleratore Elastic Graphics.

Statistica: Maximum

Soglia raccomandata: 0,0

Giustificazione della soglia: il valore di soglia 1 indica che la connettività ha riscontrato un errore.

Periodo: 300

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: GREATER_THAN_THRESHOLD

GPU HealthCheckFailed

Dimensioni:InstanceId, ePUID

Descrizione dell'allarme: questo allarme aiuta a sapere quando lo stato dell'acceleratore Elastic Graphics non è integro. Se l'acceleratore non è integro, consulta la procedura di risoluzione dei problemi alla pagina Resolve Unhealthy status issues.

Scopo: questo allarme viene utilizzato per rilevare se l'acceleratore Elastic Graphics non è integro.

Statistica: Maximum

Soglia raccomandata: 0,0

Giustificazione della soglia: il valore della soglia 1 indica che il controllo dello stato non è stato superato.

Periodo: 300

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon ECS

CPUReservation

Dimensioni: ClusterName

Descrizione dell'allarme: questo allarme consente di rilevare un'elevata prenotazione della CPU del cluster ECS. Una prenotazione della CPU elevata potrebbe indicare che il cluster sta esaurendo le CPU registrate per l'attività. Per risolvere il problema, è possibile aggiungere più capacità, dimensionare il cluster o configurare il dimensionamento automatico.

Scopo: l'allarme viene utilizzato per rilevare se le unità di CPU totali riservate dalle attività sul cluster stanno raggiungendo le unità di CPU totali registrate per il cluster. Questo ti aiuta a sapere quando aumentare il cluster. Il raggiungimento del numero totale di unità CPU per il cluster può comportare l'esaurimento della CPU per le attività. Se hai attivato il dimensionamento gestito dai provider di capacità EC2 o hai associato Fargate ai provider di capacità, questo allarme non è consigliato.

Statistica: Average

Soglia raccomandata: 90,0

Giustificazione della soglia: imposta la soglia per la prenotazione della CPU al 90%. In alternativa, è possibile scegliere un valore inferiore in base alle caratteristiche del cluster.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

CPUUtilization

Dimensioni:ClusterName, ServiceName

Descrizione dell'allarme: questo allarme consente di rilevare un elevato utilizzo della CPU del servizio ECS. In assenza di un'implementazione ECS continua, un utilizzo massimo della CPU potrebbe indicare un collo di bottiglia in termini di risorse o problemi di prestazioni delle applicazioni. Per risolvere il problema, è possibile aumentare il limite della CPU.

Scopo: questo allarme viene utilizzato per rilevare un elevato utilizzo della CPU per il servizio ECS. Un utilizzo costantemente elevato della CPU può indicare un collo di bottiglia in termini di risorse o problemi di prestazioni delle applicazioni.

Statistica: Average

Soglia raccomandata: 90,0

Giustificazione della soglia: i parametri di servizio per l'utilizzo della CPU potrebbero superare il 100% di utilizzo. Tuttavia, ti suggeriamo di monitorare il parametro per un elevato utilizzo della CPU per evitare che influisca su altri servizi. Imposta la soglia su circa il 90-95%. Ti suggeriamo di

aggiornare le definizioni delle attività in modo che riflettano l'utilizzo effettivo per evitare problemi futuri con altri servizi.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

MemoryReservation

Dimensioni: ClusterName

Descrizione dell'allarme: questo allarme consente di rilevare un'elevata riserva di memoria del cluster ECS. Una prenotazione di memoria elevata potrebbe indicare un collo di bottiglia in termini di risorse del cluster. Per risolvere il problema, analizza le prestazioni dell'attività di servizio per vedere se l'utilizzo della memoria dell'attività può essere ottimizzato. Inoltre, puoi registrare più memoria o impostare il dimensionamento automatico.

Scopo: l'allarme viene utilizzato per rilevare se le unità di memoria totali riservate dalle attività sul cluster stanno raggiungendo le unità di memoria totali registrate per il cluster. Questo può aiutarti a sapere quando aumentare il cluster. Il raggiungimento delle unità di memoria totali per il cluster può impedire al cluster di avviare nuove attività. Se hai attivato il dimensionamento gestito dai provider di capacità EC2 o hai associato Fargate ai provider di capacità, questo allarme non è consigliato.

Statistica: Average

Soglia raccomandata: 90,0

Giustificazione della soglia: imposta la soglia per la prenotazione della memoria al 90%. È possibile regolare questo valore su un valore inferiore in base alle caratteristiche del cluster.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

HTTPCode_Target_5XX_Count

Dimensioni:ClusterName, ServiceName

Descrizione dell'allarme: questo allarme consente di rilevare un elevato numero di errori lato server per il servizio ECS. Ciò può indicare che sono presenti errori che impediscono al server di evadere le richieste. Per risolvere il problema, controlla i log dell'applicazione.

Scopo: questo allarme viene utilizzato per rilevare un elevato numero di errori lato server per il servizio ECS.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: calcola il valore di circa il 5% del traffico medio e utilizza questo valore come punto di partenza per la soglia. Puoi trovare il traffico medio utilizzando il parametro `RequestCount`. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza. È necessario attivare gli allarmi per gli errori 5XX che si verificano frequentemente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

TargetResponseTime

Dimensioni:ClusterName, ServiceName

Descrizione dell'allarme: questo allarme consente di rilevare un tempo di risposta previsto elevato per le richieste del servizio ECS. Ciò può indicare che sono presenti problemi che impediscono al servizio di evadere le richieste in tempo. Per risolvere il problema, controlla il parametro di utilizzo della CPU per vedere se il servizio sta esaurendo la CPU oppure verifica l'utilizzo della CPU di altri servizi a valle da cui dipende il tuo servizio.

Scopo: questo allarme viene utilizzato per rilevare un tempo di risposta previsto elevato per le richieste del servizio ECS.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dal caso d'uso. Esamina la criticità e i requisiti del tempo di risposta previsto del servizio e analizza il comportamento storico di questo parametro per determinare livelli di soglia ragionevoli.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon ECS con Container Insights

EphemeralStorageUtilized

Dimensioni:ClusterName, ServiceName

Descrizione dell'allarme: questo allarme consente di rilevare un utilizzo elevato dello spazio di archiviazione temporaneo del cluster Fargate. Se è costantemente elevato, puoi controllare l'utilizzo dello spazio di archiviazione temporaneo e aumentarlo.

Intento: questo allarme viene utilizzato per rilevare un utilizzo elevato dello spazio di archiviazione temporaneo per il cluster Fargate. L'uso costantemente elevato dello spazio di archiviazione temporaneo può indicare che il disco è pieno e potrebbe causare errori del container.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia al 90% circa della dimensione dello spazio di archiviazione temporaneo. Puoi modificare questo valore in base all'utilizzo accettabile dello spazio di archiviazione temporaneo del cluster Fargate. Per alcuni sistemi, l'uso costantemente elevato dello spazio di archiviazione temporaneo potrebbe essere normale, mentre per altri potrebbe causare un errore del container.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

RunningTaskCount

Dimensioni:ClusterName, ServiceName

Descrizione dell'allarme: questo allarme consente di rilevare un numero basso di processi in esecuzione del servizio ECS. Un numero di processi in esecuzione troppo basso può indicare che l'applicazione non è in grado di gestire il carico del servizio, il che potrebbe causare problemi di prestazioni. Se non è presente alcun processo in esecuzione, il servizio Amazon ECS potrebbe non essere disponibile o potrebbero esserci problemi di implementazione.

Intento: questo allarme viene utilizzato per rilevare se il numero di processi in esecuzione è troppo basso. Un numero costantemente basso di processi in esecuzione può indicare problemi di implementazione o di prestazioni del servizio ECS.

Statistica: Average

Soglia raccomandata: 0,0

Giustificazione della soglia: puoi modificare la soglia in base al numero minimo di processi in esecuzione del servizio ECS. Se il numero di processi in esecuzione è 0, il servizio Amazon ECS non sarà disponibile.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: LESS_THAN_OR_EQUAL_TO_THRESHOLD

instance_filesystem_utilization

Dimensioni:InstanceId, ContainerInstanceId, ClusterName

Descrizione dell'allarme: questo allarme consente di rilevare un utilizzo elevato del file system del cluster ECS. Se l'utilizzo del file system è costantemente elevato, controlla l'utilizzo del disco.

Intento: questo allarme rileva un elevato utilizzo del file system per il cluster Amazon ECS. Un utilizzo costantemente elevato del file system può indicare un collo di bottiglia delle risorse o problemi di prestazioni delle applicazioni e impedire l'esecuzione di nuove attività.

Statistica: Average

Soglia raccomandata: 90,0

Giustificazione della soglia: puoi impostare la soglia per l'utilizzo del file system al 90-95% circa. Puoi modificare questo valore in base al livello di capacità accettabile del file system del cluster Amazon ECS. Per alcuni sistemi, un utilizzo costantemente elevato del file system potrebbe essere normale e non indicare alcun problema, mentre per altri potrebbe essere motivo di preoccupazione e causare problemi di prestazioni e impedire l'esecuzione di nuove attività.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon EFS

PercentIOLimit

Dimensioni: FileSystemId

Descrizione dell'allarme: questo allarme aiuta a garantire che il carico di lavoro rimanga entro il limite di I/O disponibile per il file system. Se il parametro raggiunge costantemente il limite di I/O, valuta la possibilità di spostare l'applicazione su un file system che utilizzi le prestazioni di I/O massime come modalità. Per la risoluzione dei problemi, controlla i client connessi al file system e le applicazioni dei client che limitano la larghezza di banda della rete del file system.

Scopo: questo allarme viene utilizzato per rilevare quanto manca a un file system per raggiungere il limite di I/O della modalità prestazioni per uso generico. Una percentuale di I/O costantemente elevata può indicare che il file system non è in grado di dimensionare in misura sufficiente rispetto alle richieste di I/O e pertanto può rappresentare un collo di bottiglia in termini di risorse per le applicazioni che lo utilizzano.

Statistica: Average

Soglia raccomandata: 100,0

Giustificazione della soglia: quando il file system raggiunge il limite di I/O, può rispondere alle richieste di lettura e scrittura più lentamente. Pertanto, si consiglia di monitorare il parametro per evitare di influire sulle applicazioni che utilizzano il file system. La soglia può essere impostata intorno al 100%. Tuttavia, questo valore può essere regolato su un valore inferiore in base alle caratteristiche del file system.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

BurstCreditBalance

Dimensioni: FileSystemId

Descrizione dell'allarme: questo allarme aiuta a garantire che sia disponibile un saldo di credito di espansione disponibile per l'utilizzo da parte del file system. Quando non è disponibile alcun credito di espansione, l'accesso delle applicazioni al file system sarà limitato a causa della bassa velocità di trasmissione effettiva. Se il parametro scende costantemente a 0, valuta la possibilità di modificare la modalità di velocità di trasmissione effettiva nella modalità di velocità di trasmissione effettiva elastica o assegnata.

Scopo: questo allarme viene utilizzato per rilevare un saldo di credito di espansione basso del file system. Un saldo di credito di espansione costantemente basso può essere un indicatore del rallentamento della velocità di trasmissione effettiva e dell'aumento della latenza di I/O.

Statistica: Average

Soglia raccomandata: 0,0

Giustificazione della soglia: quando il file system esaurisce i crediti burst e anche se la velocità di throughput di base è inferiore, EFS continua a fornire un throughput misurato pari a 1 a tutti i file system. MiBps Tuttavia, si consiglia di monitorare il parametro verificando la presenza di un saldo di credito basso per evitare che il file system costituisca un collo di bottiglia in termini di risorse per le applicazioni. La soglia può essere impostata intorno a 0 byte.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: LESS_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EKS con Container Insights

node_cpu_utilization

Dimensioni: ClusterName

Descrizione dell'allarme: questo allarme aiuta a rilevare un elevato utilizzo della CPU nei nodi worker del cluster EKS. Un utilizzo costantemente elevato potrebbe indicare la necessità di sostituire i nodi worker con istanze con una CPU maggiore o la necessità di eseguire un dimensionamento orizzontale del sistema.

Scopo: questo allarme aiuta a monitorare l'utilizzo della CPU dei nodi worker nel cluster EKS in modo che le prestazioni del sistema non si deteriorino.

Statistica: Maximum

Soglia raccomandata: 80,0

Giustificazione della soglia: si consiglia di impostare la soglia su un valore inferiore o uguale all'80% per avere tempo sufficiente per eseguire il debug del problema prima che il sistema inizi a vederne l'impatto.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

node_filesystem_utilization

Dimensioni: ClusterName

Descrizione dell'allarme: questo allarme aiuta a rilevare un utilizzo elevato del file system nei nodi worker del cluster EKS. Se l'utilizzo è costantemente elevato, potrebbe essere necessario

aggiornare i nodi worker per incrementare le dimensioni del volume del disco oppure potrebbe essere necessario sottoporli a un dimensionamento orizzontale.

Scopo: questo allarme aiuta a monitorare l'utilizzo del file system nei nodi worker del cluster EKS. Se l'utilizzo raggiunge il 100%, può causare errori dell'applicazione, colli di bottiglia nell'I/O del disco, l'espulsione del pod o la completa interruzione della risposta del nodo.

Statistica: Maximum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: in presenza di una capacità di archiviazione prossima alla saturazione (il disco si sta riempiendo), i nodi vengono segnalati come non integri e i pod vengono espulsi dal nodo. I pod su un nodo con un carico elevato sul disco vengono espulsi quando il file system disponibile è inferiore alle soglie di espulsione impostate sul kubelet. Imposta la soglia di allarme in modo da avere abbastanza tempo per reagire prima che il nodo venga espulso dal cluster.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

node_memory_utilization

Dimensioni: ClusterName

Descrizione dell'allarme: questo allarme aiuta a rilevare un elevato utilizzo della memoria nei nodi worker del cluster EKS. Se l'utilizzo è costantemente elevato, potrebbe indicare la necessità di dimensionare il numero di repliche dei pod o ottimizzare l'applicazione.

Scopo: questo allarme aiuta a monitorare l'utilizzo della memoria dei nodi worker nel cluster EKS in modo che le prestazioni del sistema non si deteriorino.

Statistica: Maximum

Soglia raccomandata: 80,0

Giustificazione della soglia: si consiglia di impostare la soglia su un valore inferiore o uguale all'80% per avere tempo sufficiente per eseguire il debug del problema prima che il sistema cominci a vederne l'impatto.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

pod_cpu_utilization_over_pod_limit

Dimensioni:ClusterName, Namespace, Service

Descrizione dell'allarme: questo allarme aiuta a rilevare un elevato utilizzo della CPU nei pod del cluster EKS. Se l'utilizzo è costantemente elevato, potrebbe indicare la necessità di aumentare il limite di CPU per il pod interessato.

Scopo: questo allarme aiuta a monitorare l'utilizzo della CPU dei pod appartenenti a un servizio Kubernetes nel cluster EKS, in modo da poter identificare rapidamente se il pod di un servizio sta utilizzando più CPU del previsto.

Statistica: Maximum

Soglia raccomandata: 80,0

Giustificazione della soglia: si consiglia di impostare la soglia su un valore inferiore o uguale all'80% per avere tempo sufficiente per eseguire il debug del problema prima che il sistema cominci a vederne l'impatto.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

pod_memory_utilization_over_pod_limit

Dimensioni:ClusterName, Namespace, Service

Descrizione dell'allarme: questo allarme aiuta a rilevare un elevato utilizzo della memoria nei pod del cluster EKS. Se l'utilizzo è costantemente elevato, potrebbe indicare la necessità di aumentare il limite di memoria per il pod interessato.

Scopo: questo allarme aiuta a monitorare l'utilizzo della memoria dei pod nel cluster EKS in modo che le prestazioni del sistema non si deteriorino.

Statistica: Maximum

Soglia raccomandata: 80,0

Giustificazione della soglia: si consiglia di impostare la soglia su un valore inferiore o uguale all'80% per avere tempo sufficiente per eseguire il debug del problema prima che il sistema cominci a vederne l'impatto.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon Kinesis Data Streams

GetRecords.IteratorAgeMilliseconds

Dimensioni: StreamName

Descrizione dell'allarme: questo allarme può rilevare se l'età massima dell'iteratore è troppo alta. Per le applicazioni di elaborazione dei dati in tempo reale, configura la conservazione dei dati in base alla tolleranza del ritardo. Di solito si tratta di pochi minuti. Per le applicazioni che elaborano dati storici, utilizza questo parametro per monitorare la velocità di recupero dell'arretrato. Una soluzione rapida per arrestare la perdita di dati consiste nell'aumentare il periodo di conservazione durante la risoluzione del problema. Inoltre, è possibile aumentare il numero di worker che elaborano i record nella propria applicazione per consumatori. Le ragioni più comuni dell'incremento graduale dell'età dell'iteratore includono la mancanza di risorse fisiche adeguate o una logica di elaborazione dei record che non è stata dimensionata per gestire un aumento della velocità di trasmissione effettiva del flusso. Consulta questo collegamento per ulteriori dettagli.

Scopo: questo allarme viene utilizzato per rilevare se i dati del flusso stanno per scadere perché conservati troppo a lungo o perché l'elaborazione dei record è troppo lenta. Contribuisce a evitare la perdita di dati dopo aver raggiunto il 100% del tempo di conservazione del flusso.

Statistica: Maximum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dal periodo di conservazione del flusso e dalla tolleranza del ritardo di elaborazione per i record. Esamina i requisiti e analizza le tendenze storiche, quindi imposta la soglia sul numero di millisecondi che rappresenta un ritardo di elaborazione critico. Se l'età di un iteratore supera il 50% del periodo di conservazione (per impostazione predefinita, 24 ore, configurabile fino a 365 giorni), esiste il rischio di una perdita di dati a causa della scadenza del record. Puoi monitorare il parametro per assicurarti che nessuna delle partizioni si avvicini mai a questo limite.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

GetRecords.Successo

Dimensioni: StreamName

Descrizione dell'allarme: questo parametro aumenta ogni volta che i consumatori leggono correttamente i dati dal flusso. `GetRecords` non restituisce alcun dato quando genera un'eccezione. L'eccezione più comune è `ProvisionedThroughputExceededException`, dovuta al fatto che il tasso di richiesta del flusso è troppo alto o la velocità di trasmissione effettiva disponibile è già utilizzata per il secondo in questione. Riduci la frequenza o le dimensioni delle richieste. Per ulteriori informazioni, consulta la pagina Limits relativa ai flussi nella Guida per gli sviluppatori di Flusso di dati Amazon Kinesis e la pagina Error Retries and Exponential Backoff in AWS.

Scopo: questo allarme è in grado di rilevare se il recupero dei record dal flusso da parte dei consumatori ha esito negativo. Attivando un allarme per questo parametro, puoi rilevare in modo proattivo eventuali problemi relativi all'utilizzo dei dati, ad esempio un aumento dei tassi di errore o un calo dei recuperi riusciti. Ciò consente di intraprendere operazioni tempestive per risolvere potenziali problemi e preservare la fluidità della pipeline di elaborazione dei dati.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: a seconda dell'importanza del recupero dei record dal flusso, imposta la soglia in base alla tolleranza dell'applicazione per i record non riusciti. La soglia deve essere la percentuale corrispondente di operazioni riuscite. È possibile utilizzare i dati GetRecords metrici storici come riferimento per il tasso di errore accettabile. Inoltre, quando si imposta la soglia, è necessario considerare i nuovi tentativi, poiché i record con errori possono essere ritentati. Ciò consente di evitare che i picchi transitori generino avvisi non necessari.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: LESS_THAN_THRESHOLD

PutRecord.Successo

Dimensioni: StreamName

Descrizione dell'allarme: questo allarme rileva quando il numero di operazioni `PutRecord` non riuscite supera la soglia. Esamina i log del produttore di dati per individuare le cause principali degli errori. Il motivo più comune è la velocità di trasmissione effettiva insufficiente assegnata alla partizione che ha causato l'eccezione `ProvisionedThroughputExceededException`. Ciò accade perché il tasso delle richieste per il flusso è troppo alto o la velocità di trasmissione effettiva con cui si tenta di importare nella partizione è troppo alto. Riduci la frequenza o le dimensioni delle richieste. Per ulteriori informazioni, consulta Streams Limits and Error Retries e Exponential Backoff in. AWS

Scopo: questo allarme può rilevare se l'importazione dei record nel flusso non riesce. Ti aiuta a identificare i problemi nella scrittura dei dati nel flusso. Attivando un allarme per questo parametro, puoi rilevare in modo proattivo eventuali problemi dei produttori nella pubblicazione dei dati nel flusso, come un aumento dei tassi di errore o una diminuzione dei record pubblicati con successo. Ciò consente di intraprendere operazioni tempestive per risolvere potenziali problemi e preservare l'affidabilità del processo di importazione dei dati.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: a seconda dell'importanza dell'importazione e dell'elaborazione dei dati per il servizio, imposta la soglia in base alla tolleranza dell'applicazione per i record non

riusciti. La soglia deve essere la percentuale corrispondente di operazioni riuscite. Puoi utilizzare i dati PutRecord metrici storici come riferimento per il tasso di errore accettabile. Inoltre, quando si imposta la soglia, è necessario considerare i nuovi tentativi, poiché i record con errori possono essere ritentati.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: LESS_THAN_THRESHOLD

PutRecords.FailedRecords

Dimensioni: StreamName

Descrizione dell'allarme: questo allarme rileva quando il numero di errori `PutRecords` supera la soglia. Flusso di dati Kinesis tenta di elaborare tutti i record in ogni richiesta `PutRecords`, ma l'errore di un singolo record non interrompe l'elaborazione di quelli successivi. Il motivo principale di questi errori è il superamento della velocità di trasmissione effettiva di un flusso o di una singola partizione. Le cause più comuni sono i picchi di traffico e le latenze di rete che causano l'arrivo dei record al flusso in modo non uniforme. È necessario rilevare i record non elaborati correttamente e ritentarli nella chiamata successiva. Per ulteriori dettagli, fare riferimento a Gestione degli errori durante PutRecords l'utilizzo.

Scopo: questo allarme può rilevare errori costanti quando si utilizza l'operazione in batch per inviare i record al flusso. Attivando un allarme per questo parametro, puoi rilevare in modo proattivo un aumento dei record non riusciti, al fine di intraprendere operazioni tempestive per risolvere i problemi sottostanti e garantire un processo di importazione dei dati fluido e affidabile.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia sul numero di record non riusciti che riflette la tolleranza dell'applicazione per i record non riusciti. È possibile utilizzare i dati storici come indicazione per il valore di errore accettabile. È inoltre necessario considerare i nuovi tentativi quando si imposta la soglia, poiché i record non riusciti possono essere riprovati nelle chiamate successive. PutRecords

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

ReadProvisionedThroughputExceeded

Dimensioni: StreamName

Descrizione dell'allarme: l'allarme tiene traccia del numero di record che determinano una limitazione della larghezza di banda della rete della capacità effettiva di trasmissione in lettura. Se noti che la larghezza di banda della rete viene continuamente limitata, dovresti prendere in considerazione l'aggiunta di altre partizioni al flusso per aumentare la velocità di trasmissione effettiva di lettura assegnata. Se nel flusso è in esecuzione più di un'applicazione consumatore che condivide il limite `GetRecords`, ti consigliamo di registrare le nuove applicazioni consumatori tramite il fan-out avanzato. Se l'aggiunta di altre partizioni non riduce il numero di limitazioni della larghezza di banda della rete, è possibile che una partizione "calda" riceva più letture rispetto alle altre partizioni. Abilita il monitoraggio avanzato, individua la partizione "calda" e suddividila.

Scopo: questo allarme è in grado di rilevare se la larghezza di banda della rete dei consumatori viene limitata quando i consumatori stessi superano la velocità di trasmissione effettiva di lettura assegnata (determinata dal numero di partizioni di cui disponi). In tal caso, non sarai in grado di leggere dal flusso e potrà essere avviato il backup del flusso.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: in genere le richieste sottoposte a limitazione della larghezza di banda della rete possono essere ritentate, quindi l'impostazione della soglia su zero rende l'allarme troppo sensibile. Tuttavia, la continua limitazione della larghezza di banda della rete può influire sulla lettura dal flusso e far scattare l'allarme. Imposta la soglia su una percentuale in base alle richieste sottoposte a limitazioni della larghezza di banda della rete per l'applicazione e ritenta le configurazioni.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

SubscribeToShardEvent.MillisBehindLatest

Dimensioni:StreamName, ConsumerName

Descrizione dell'allarme: questo allarme rileva quando il ritardo di elaborazione dei record nell'applicazione supera la soglia. Problemi transitori, come i malfunzionamenti delle API di un'applicazione a valle, possono causare un aumento improvviso del parametro. È necessario indagare se si verificano costantemente. Una causa comune è che il consumatore non elabora i record con una velocità sufficiente a causa delle risorse fisiche insufficienti o della logica di elaborazione dei record che non si è dimensionata all'aumento della velocità di trasmissione effettiva. Il blocco delle chiamate nel percorso critico è spesso causa di rallentamenti nell'elaborazione dei record. È possibile aumentare il parallelismo aumentando il numero di partizioni. Inoltre, è necessario verificare che i nodi di elaborazione sottostanti dispongano di risorse fisiche sufficienti durante i picchi di domanda.

Scopo: questo allarme può rilevare un ritardo nell'abbonamento all'evento di partizione del flusso. Ciò indica un ritardo di elaborazione e può contribuire a identificare potenziali problemi con le prestazioni dell'applicazione consumatore o l'integrità generale del flusso. Quando il ritardo di elaborazione diventa significativo, è necessario analizzare e risolvere eventuali colli di bottiglia o inefficienze delle applicazioni consumatore per garantire l'elaborazione dei dati in tempo reale e ridurre al minimo il backlog dei dati.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dal ritardo che l'applicazione è in grado di tollerare. Esamina i requisiti dell'applicazione e analizza le tendenze storiche, quindi seleziona una soglia di conseguenza. Quando la SubscribeToShard chiamata ha esito positivo, l'utente inizia a ricevere SubscribeToShardEvent eventi tramite la connessione persistente per un massimo di 5 minuti, dopodiché è necessario chiamare SubscribeToShard nuovamente per rinnovare l'abbonamento se si desidera continuare a ricevere registrazioni.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

WriteProvisionedThroughputExceeded

Dimensioni: StreamName

Descrizione dell'allarme: questo allarme rileva quando il numero di record con conseguente limitazione della larghezza di banda della rete della capacità effettiva di trasmissione di scrittura ha raggiunto la soglia. Quando superano la velocità di trasmissione effettiva di scrittura assegnata (determinata dal numero di partizioni di cui disponi), i produttori vengono sottoposti a limitazione della larghezza di banda della rete e non sarà possibile inviare record al flusso. Per ovviare alla continua limitazione della larghezza di banda della rete, dovresti prendere in considerazione l'aggiunta di partizioni al flusso. Ciò aumenta la velocità di trasmissione effettiva di scrittura assegnata e previene future limitazioni della larghezza di banda della rete. Inoltre, è necessario prendere in considerazione la scelta della chiave di partizione quando si importano i record. La chiave di partizione casuale è preferita perché distribuisce i record in modo uniforme tra le partizioni del flusso, quando possibile.

Scopo: questo allarme può rilevare se ai produttori viene rifiutata la scrittura dei record a causa della limitazione della larghezza di banda della rete del flusso o della partizione. Se il flusso è in modalità assegnata, l'impostazione di questo allarme consente di intervenire in modo proattivo quando il flusso di dati raggiunge i limiti, ottimizzando la capacità fornita o adottando le operazioni di dimensionamento appropriate per evitare la perdita di dati e garantire un'elaborazione dei dati fluida.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: in genere le richieste sottoposte a limitazione della larghezza di banda della rete possono essere ritentate, quindi l'impostazione della soglia su zero rende l'allarme troppo sensibile. Tuttavia, una continua limitazione della larghezza di banda della rete può influire sulla scrittura nel flusso ed è necessario impostare una soglia di allarme per rilevarla. Imposta la soglia su una percentuale in base alle richieste sottoposte a limitazioni della larghezza di banda della rete per l'applicazione e ritenta le configurazioni.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Lambda

ClaimedAccountConcurrency

Dimensions: nessuna

Descrizione dell'allarme: questo allarme aiuta a monitorare se la concorrenza delle funzioni Lambda si avvicina al limite di concorrenza a livello di regione del tuo account. La larghezza di banda della rete di una funzione inizia a essere limitata se raggiunge il limite di simultaneità. È possibile eseguire le seguenti operazioni per evitare la limitazione della larghezza di banda della rete.

1. Richiedi un aumento simultaneo della concorrenza in questa regione.
2. Identifica e riduci qualsiasi concorrenza riservata o accantonata non utilizzata.
3. Identifica i problemi di prestazioni nelle funzioni per migliorare la velocità di elaborazione e quindi migliorare la produttività.
4. Aumentate la dimensione del batch delle funzioni, in modo che vengano elaborati più messaggi a ogni chiamata di funzione.

Intento: questo allarme può rilevare in modo proattivo se la concorrenza delle funzioni Lambda si avvicina alla quota di concorrenza a livello regionale del tuo account, in modo che tu possa agire di conseguenza. Le funzioni vengono limitate se viene raggiunta la quota di concorrenza a livello regionale dell'account. `ClaimedAccountConcurrency` Se si utilizza Reserved Concurrency (RC) o Provisioned Concurrency (PC), questo allarme offre una maggiore visibilità sull'utilizzo della concorrenza rispetto a un allarme attivo. `ConcurrentExecutions`

Statistica: Maximum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: è necessario calcolare il valore di circa il 90% della quota di concorrenza impostata per l'account nella regione e utilizzare il risultato come valore di soglia. Per impostazione predefinita, l'account ha un limite di simultaneità pari a 1.000 per tutte le funzioni in una regione. Tuttavia, dovresti controllare la quota del tuo account dalla dashboard di Service Quotas.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_THRESHOLD

Errori

Dimensioni: FunctionName

Descrizione dell'allarme: questo allarme rileva un numero elevato di errori. Gli errori includono eccezioni generate dal codice e eccezioni generate dal runtime Lambda. È possibile controllare i log relativi alla funzione per diagnosticare il problema.

Scopo: l'allarme aiuta a rilevare un numero elevato di errori nelle invocazioni di funzioni.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia su un numero maggiore di zero. Il valore esatto può dipendere dalla tolleranza agli errori nell'applicazione. Comprendi la criticità delle chiamate gestite dalla funzione. Per alcune applicazioni, qualsiasi errore potrebbe essere inaccettabile, mentre altre applicazioni potrebbero consentire un certo margine di errore.

Periodo: 60

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: GREATER_THAN_THRESHOLD

Throttles

Dimensioni: FunctionName

Descrizione dell'allarme: questo allarme rileva un numero elevato di richieste di invocazione sottoposte a limitazione della larghezza di banda della rete. La limitazione della larghezza di banda della rete si verifica quando non è disponibile simultaneità per l'aumento. Esistono diversi approcci per risolvere il problema. 1) Richiedi un aumento simultaneo all' AWS assistenza in questa regione. 2) Identificare i problemi di prestazioni della funzione per migliorare la velocità di elaborazione e di conseguenza la velocità di trasmissione effettiva. 3) Aumentare la dimensione

del batch della funzione, in modo che a ogni invocazione della funzione vengano elaborati più messaggi.

Scopo: l'allarme aiuta a rilevare un numero elevato di richieste di invocazione con limitazione della larghezza di banda della rete per una funzione Lambda. È importante sapere se le richieste vengono costantemente rifiutate a causa della limitazione della larghezza di banda della rete e se è necessario migliorare le prestazioni della funzione Lambda o aumentare la capacità di simultaneità per evitare la limitazione costante.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia su un numero maggiore di zero. Il valore esatto della soglia può dipendere dalla tolleranza dell'applicazione. Imposta la soglia in base all'utilizzo e ai requisiti di dimensionamento della funzione.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Duration (Durata)

Dimensioni: FunctionName

Descrizione dell'allarme: questo allarme rileva tempi di elaborazione di un evento prolungati da parte di una funzione Lambda. Durate significative potrebbero essere dovute a modifiche nel codice della funzione che rendono più lunga l'esecuzione della funzione o delle rispettive dipendenze.

Scopo: questo allarme può rilevare una lunga durata di esecuzione di una funzione Lambda. Un'elevata durata di runtime indica che una funzione impiega più tempo per l'invocazione e può anche influire sulla capacità di chiamata simultanea nel caso in cui Lambda stia gestendo un numero maggiore di eventi. È fondamentale sapere se la funzione Lambda richiede costantemente tempi di esecuzione più lunghi del previsto.

Statistica: p90

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: la soglia per la durata dipende dall'applicazione e dai carichi di lavoro e dai requisiti di prestazioni. Per requisiti di prestazione elevati, imposta la soglia su un periodo di tempo più breve per verificare se la funzione soddisfa le aspettative. Puoi anche analizzare i dati storici dei parametri di durata per vedere se il tempo impiegato corrisponde alle aspettative di prestazioni della funzione e impostare la soglia su un periodo più lungo rispetto alla media storica. Assicurati di impostare la soglia al di sotto del timeout della funzione configurata.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

ConcurrentExecutions

Dimensioni: FunctionName

Descrizione dell'allarme: questo allarme aiuta a monitorare se la simultaneità della funzione si avvicina al limite di simultaneità a livello regionale dell'account. La larghezza di banda della rete di una funzione inizia a essere limitata se raggiunge il limite di simultaneità. È possibile eseguire le seguenti operazioni per evitare la limitazione della larghezza di banda della rete.

1. Richiedi un aumento concomitante in questa regione.
2. Identifica i problemi di prestazioni nelle funzioni per migliorare la velocità di elaborazione e quindi migliorare la produttività.
3. Aumentate la dimensione del batch delle funzioni, in modo che vengano elaborati più messaggi a ogni chiamata di funzione.

Per ottenere una migliore visibilità sulla concorrenza riservata e sull'utilizzo della concorrenza fornita, imposta invece un allarme sulla nuova metrica. `ClaimedAccountConcurrency`

Scopo: questo allarme può rilevare in modo proattivo se la simultaneità della funzione si avvicina alla quota di simultaneità a livello di regione per l'account, in modo che tu possa agire di conseguenza. La larghezza di banda della rete di una funzione viene limitata se raggiunge la quota di simultaneità a livello di regione per l'account.

Statistica: Maximum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia su circa il 90% della quota di simultaneità impostata per l'account nella regione. Per impostazione predefinita, l'account ha un limite di simultaneità pari a 1.000 per tutte le funzioni in una regione. Tuttavia, puoi verificare la quota del tuo account, in quanto può essere aumentata contattando l'assistenza. AWS

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_THRESHOLD

Lambda Insights

Come best practice, consigliamo di attivare allarmi basati sui seguenti parametri di Lambda Insights.

memory_utilization

Dimensioni: function_name

Descrizione dell'allarme: questo allarme viene utilizzato per rilevare se l'utilizzo della memoria di una funzione Lambda si avvicina al limite configurato. Per risolvere il problema, puoi provare a: 1) Ottimizzare il codice. 2) Dimensionare correttamente l'allocazione di memoria stimando accuratamente i requisiti di memoria. A tale scopo, puoi fare riferimento a Lambda Power Tuning. 3) Utilizzare il pooling delle connessioni. Per il pooling delle connessioni per i database RDS, consulta la pagina Using Amazon RDS Proxy with Lambda. 4) Puoi anche prendere in considerazione la possibilità di progettare le tue funzioni per evitare di archiviare grandi quantità di dati in memoria tra un'invocazione e l'altra.

Scopo: questo allarme viene utilizzato per rilevare se l'utilizzo della memoria per la funzione Lambda si avvicina al limite configurato.

Statistica: Average

Soglia raccomandata: 90,0

Giustificazione della soglia: imposta la soglia al 90% per ricevere un avviso quando l'utilizzo della memoria supera il 90% della memoria allocata. Se l'utilizzo della memoria da parte del carico

di lavoro rappresenta un fattore critico, è possibile impostarla su un valore inferiore. È inoltre possibile controllare i dati storici per questo parametro e impostare la soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

ComparisonOperator: GREATER_THAN_THRESHOLD

Amazon VPC (**`AWS/NATGateway`**)

ErrorPortAllocation

Dimensioni: NatGatewayId

Descrizione dell'allarme: questo allarme aiuta a rilevare quando il gateway NAT non è in grado di allocare le porte a nuove connessioni. Per risolvere questo problema, consulta la pagina Resolve port allocation errors on NAT Gateway.

Scopo: questo allarme viene utilizzato per rilevare se il gateway NAT non è in grado di allocare una porta di origine.

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: se il valore di ErrorPortAllocation è maggiore di zero, significa che tramite NatGateway sono aperte troppe connessioni simultanee verso una singola destinazione popolare.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

PacketsDropCount

Dimensioni: NatGatewayId

Descrizione dell'allarme: questo allarme aiuta a rilevare quando i pacchetti vengono eliminati dal Gateway NAT. Ciò potrebbe accadere a causa di un problema con NAT Gateway, quindi controllate lo stato di AWS NAT Gateway nella vostra regione nel pannello di controllo dello stato del AWS servizio. Questo può aiutarti a stabilire un collegamento con il problema di rete relativo al traffico che utilizza il gateway NAT.

Scopo: questo allarme viene utilizzato per rilevare se i pacchetti vengono eliminati dal gateway NAT.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: è necessario calcolare il valore dello 0,01% del traffico totale sul gateway NAT e utilizzare tale risultato come valore di soglia. Utilizza i dati storici del traffico sul gateway NAT per stabilire la soglia.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

AWS Link privato () **`AWS/PrivateLinkEndpoints`**

PacketsDropped

Dimensioni: ID VPC, ID endpoint VPC, tipo di endpoint, ID sottorete, nome del servizio

Descrizione dell'allarme: questo allarme aiuta a rilevare se l'endpoint o il servizio endpoint non è integro monitorando il numero di pacchetti rilasciati dall'endpoint. I pacchetti con dimensioni superiori a 8.500 byte che arrivano all'endpoint VPC vengono eliminati. Per la risoluzione dei problemi, consulta la pagina Connectivity problems between an interface VPC endpoint and an endpoint service.

Scopo: questo allarme viene utilizzato per rilevare se l'endpoint o il servizio endpoint non è integro.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base al caso d'uso. Per essere consapevole dello stato di non integrità dell'endpoint o del servizio endpoint, dovresti impostare la soglia su un valore basso in modo da avere la possibilità di risolvere il problema prima che si verifichi un'enorme perdita di dati. È possibile utilizzare i dati storici per comprendere la tolleranza in caso di perdita di pacchetti e impostare la soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

AWS Link privato (**`AWS/PrivateLinkServices`**)

RstPacketsSent

Dimensioni: ID servizio, ARN sistema di bilanciamento del carico, AZ

Descrizione dell'allarme: questo allarme consente di rilevare gli obiettivi non integri di un servizio endpoint in base al numero di pacchetti di ripristino inviati agli endpoint. Quando esegui il debug degli errori di connessione con un utente del tuo servizio, puoi verificare se il servizio sta ripristinando le connessioni con la RstPacketsSent metrica o se qualcos'altro non funziona sul percorso di rete.

Scopo: questo allarme viene utilizzato per rilevare gli obiettivi non integri di un servizio endpoint.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: a soglia dipende dal caso d'uso. Se il tuo caso d'uso può tollerare obiettivi non integri, puoi impostare la soglia su un valore alto. Se il caso d'uso non è in grado di tollerare obiettivi non integri, puoi impostare la soglia su un valore molto basso.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon RDS

CPUUtilization

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare un utilizzo costantemente elevato della CPU. L'utilizzo della CPU misura il tempo di non inattività. Prendi in considerazione l'utilizzo di Enhanced Monitoring o Performance Insights per verificare quale tempo di attesa consuma la maggior parte del tempo della CPU (`guest`, `irq`, `wait`, `nice` e così via) per MariaDB, MySQL, Oracle e PostgreSQL. Quindi valuta quali query consumano la maggiore quantità di CPU. Se non riesci a ottimizzare il carico di lavoro, valuta la possibilità di passare a una classe di istanze database più ampia.

Intento: questo allarme viene utilizzato per rilevare un utilizzo elevato e costante della CPU al fine di evitare tempi di risposta e timeout molto elevati. Se si desidera verificare il micro-bursting dell'utilizzo della CPU, è possibile impostare un tempo inferiore di valutazione degli allarmi.

Statistica: Average

Soglia raccomandata: 90,0

Giustificazione della soglia: i picchi casuali nel consumo di CPU potrebbero non influire sulle prestazioni del database, ma un utilizzo elevato e prolungato della CPU può ostacolare le richieste del database in arrivo. A seconda del carico di lavoro complessivo del database, una CPU elevata sull'istanza RDS/Aurora può ridurre le prestazioni complessive.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

DatabaseConnections

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme rileva un numero elevato di connessioni. Controlla le connessioni esistenti e interrompi quelle che sono in stato di “sospensione” o che non sono chiuse correttamente. Prendi in considerazione l'uso del pool di connessioni per limitare il numero di nuove connessioni. In alternativa, aumenta la dimensione dell'istanza database per utilizzare una classe con più memoria e quindi un valore predefinito più alto per “max_connections” oppure aumenta il valore “max_connections” in RDS e Aurora MySQL e PostgreSQL per la classe corrente se può supportare il tuo carico di lavoro.

Intento: questo allarme viene utilizzato per prevenire il rifiuto delle connessioni quando viene raggiunto il numero massimo di connessioni DB. Questo allarme non è consigliato nel caso in cui si cambi frequentemente la classe dell'istanza database, poiché così facendo si modificano la memoria e il numero massimo predefinito di connessioni.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il numero di connessioni consentite dipende dalla dimensione della classe dell'istanza database e dai parametri specifici del motore di database relativi a processi/connessioni. È necessario calcolare un valore compreso tra il 90 e il 95% del numero massimo di connessioni per il database e utilizzare tale risultato come valore di soglia.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

EBS% ByteBalance

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare una bassa percentuale di crediti della velocità di trasmissione effettiva rimanenti. Per la risoluzione dei problemi, controlla i problemi di latenza in RDS.

Intento: questo allarme è utilizzato per rilevare una bassa percentuale di crediti della velocità di trasmissione effettiva rimanenti nel burst bucket. Una bassa percentuale di saldo byte può causare problemi di velocità di trasmissione effettiva. Questo allarme non è consigliato per le istanze Aurora PostgreSQL.

Statistica: Average

Soglia raccomandata: 10,0

Giustificazione della soglia: un saldo a credito della velocità di trasmissione effettiva inferiore al 10% è considerato insufficiente ed è necessario impostare la soglia di conseguenza. È anche possibile impostare una soglia inferiore se l'applicazione è in grado di tollerare una velocità di trasmissione effettiva inferiore per il carico di lavoro.

Periodo: 60

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: LESS_THAN_THRESHOLD

EBSIOBalance%

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare una bassa percentuale di crediti IOPS rimanenti. Per la risoluzione dei problemi, consulta i problemi di latenza in RDS.

Intento: questo allarme è utilizzato per rilevare una bassa percentuale di crediti I/O rimanenti nel burst bucket. Una bassa percentuale di saldo IOPS può causare problemi di colli di bottiglia. Questo allarme non è consigliato per le istanze Aurora.

Statistica: Average

Soglia raccomandata: 10,0

Giustificazione della soglia: un saldo a credito IOPS inferiore al 10% è considerato insufficiente ed è possibile impostare la soglia di conseguenza. È anche possibile impostare una soglia inferiore, se l'applicazione è in grado di tollerare IOPS inferiore per il carico di lavoro.

Periodo: 60

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: LESS_THAN_THRESHOLD

FreeableMemory

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare una scarsa memoria liberabile, il che può indicare un picco nelle connessioni al database o che l'istanza potrebbe essere sottoposta a un'elevata pressione di memoria. Controlla la pressione della memoria monitorando le CloudWatch metriche relative `SwapUsage` a `in aggiunta a. `FreeableMemory` Se il consumo di memoria dell'istanza è spesso troppo elevato, è necessario controllare il carico di lavoro o aggiornare la classe di istanza. Per un'istanza database di lettura Aurora, valuta la possibilità di aggiungere più istanze database di lettura al cluster. Per ulteriori informazioni sulla risoluzione dei problemi di Aurora, consulta i problemi di memoria liberabile.

Intento: questo allarme viene utilizzato per evitare l'esaurimento della memoria, che può causare il rifiuto delle connessioni.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: a seconda del carico di lavoro e della classe di istanza, possono essere appropriati diversi valori per la soglia. Idealmente, la memoria disponibile non dovrebbe scendere al di sotto del 25% della memoria totale per periodi prolungati. Per Aurora, puoi impostare la soglia vicino al 5%, poiché un parametro vicino a 0 indica che l'istanza database ha eseguito il dimensionamento al valore massimo possibile. È possibile analizzare il comportamento storico di questo parametro per determinare livelli di soglia ragionevoli.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: LESS_THAN_THRESHOLD

FreeLocalStorage

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare uno scarso spazio di archiviazione locale. Aurora edizione compatibile con PostgreSQL utilizza l'archiviazione locale per

memorizzare i log degli errori e i file temporanei. Aurora MySQL utilizza l'archiviazione locale per archiviare i log degli errori, i log generali, i log delle query lente, i log di audit e le tabelle temporanee non InnoDB. Questi volumi di archiviazione locale sono supportati da Amazon EBS Store e possono essere estesi utilizzando una classe di istanza database più grande. Per la risoluzione dei problemi, verifica Aurora edizione compatibile con PostgreSQL ed edizione compatibile con MySQL.

Intento: questo allarme viene utilizzato per rilevare quanto manca all'istanza di database Aurora per raggiungere il limite di archiviazione locale, se non si usa Aurora Serverless v2 o versione successiva. L'archiviazione locale può raggiungere il limite di capacità quando si archiviano dati non persistenti, come tabelle e file di log temporanei, nella memoria locale. Questo allarme può prevenire un out-of-space errore che si verifica quando l'istanza DB esaurisce lo spazio di archiviazione locale.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: è necessario calcolare circa il 10%-20% dello spazio di archiviazione disponibile in base alla velocità e all'andamento dell'utilizzo del volume, quindi utilizzare tale risultato come valore di soglia per intervenire in modo proattivo prima che il volume raggiunga il limite.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: LESS_THAN_THRESHOLD

FreeStorageSpace

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme rileva una scarsa quantità di spazio di archiviazione disponibile. Se ti avvicini spesso ai limiti di capacità di archiviazione, prendi in considerazione la possibilità di aumentare lo spazio di archiviazione del database. Includi un buffer di memoria per soddisfare gli aumenti imprevisti della domanda delle tue applicazioni. In alternativa, valuta la possibilità di abilitare il dimensionamento automatico dell'archiviazione RDS. Puoi anche

valutare la possibilità di liberare più spazio eliminando dati e log obsoleti o inutilizzati. Per ulteriori informazioni, consulta il documento sulla mancanza di spazio per RDS e il documento sui problemi di archiviazione di PostgreSQL.

Intento: questo allarme aiuta a prevenire problemi di esaurimento dell'archiviazione. Può aiutare a prevenire il tempo di inattività che si verifica quando l'istanza di database esaurisce lo spazio di archiviazione. Si sconsiglia di utilizzare questo allarme se è abilitato il dimensionamento automatico dell'archiviazione o se si modifica frequentemente la capacità di archiviazione dell'istanza di database.

Statistica: Minimum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore della soglia dipenderà dallo spazio di archiviazione attualmente allocato. In genere, è necessario calcolare il 10% dello spazio di archiviazione allocato e utilizzare tale risultato come valore di soglia.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: LESS_THAN_THRESHOLD

MaximumUsedTransactionID

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a impedire il wraparound dell'ID delle transazioni per PostgreSQL. Fai riferimento alla procedura di risoluzione dei problemi in questo blog per esaminare e risolvere il problema. Puoi anche fare riferimento a questo blog per acquisire maggiore familiarità con i concetti relativi all'autovacuum, i problemi più comuni e le best practice.

Intento: questo allarme viene utilizzato per impedire il wraparound dell'ID delle transazioni per PostgreSQL.

Statistica: Average

Soglia raccomandata: 1,0E9

Giustificazione della soglia: l'impostazione di questa soglia a 1 miliardo dovrebbe darti il tempo di esaminare il problema. Il valore predefinito di autovacuum_freeze_max_age è 200 milioni. Se l'età della transazione più vecchia è 1 miliardo, l'autovacuum ha problemi a mantenere questa soglia al di sotto dell'obiettivo di 200 milioni di ID di transazione.

Periodo: 60

Punti dati su cui attivare allarmi: 1

Periodi di valutazione: 1

Operatore di confronto: GREATER_THAN_THRESHOLD

ReadLatency

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare una latenza di lettura elevata. Se la latenza dell'archiviazione è elevata, il carico di lavoro supera i limiti delle risorse. È possibile esaminare l'utilizzo dell'I/O in relazione alla configurazione dell'istanza e dell'archiviazione allocata. Fai riferimento alla risoluzione dei problemi di latenza dei volumi Amazon EBS causata da un collo di bottiglia IOPS. Per Aurora, puoi passare a una classe di istanza con una configurazione di archiviazione con ottimizzazione per I/O. Per informazioni, consulta Pianificazione dell'I/O in Aurora.

Intento: questo allarme viene utilizzato per rilevare una latenza di lettura elevata. I dischi del database hanno normalmente una bassa latenza di lettura/scrittura, ma possono presentare problemi che causano operazioni a latenza elevata.

Statistica: p90

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dal caso d'uso. Le latenze di lettura superiori a 20 millisecondi richiedono probabilmente un'analisi. È inoltre possibile impostare una soglia più alta se l'applicazione può avere una latenza più elevata per le operazioni di lettura. Esamina la criticità e i requisiti della latenza di lettura e analizza il comportamento storico di questo parametro per determinare livelli di soglia ragionevoli.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

ReplicaLag

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme ti aiuta a capire il numero di secondi di ritardo di una replica di lettura rispetto all'istanza principale. Una replica di lettura PostgreSQL segnala un ritardo di replica fino a cinque minuti se non ci sono transazioni di utenti sull'istanza database di origine. Quando la ReplicaLag metrica raggiunge 0, la replica ha raggiunto l'istanza DB principale. Se la ReplicaLag metrica restituisce -1, la replica non è attualmente attiva. Per indicazioni relative a RDS PostgreSQL, consulta le migliori pratiche di replica e per `ReplicaLag` la risoluzione dei problemi e gli errori correlati, consulta Risoluzione dei problemi. ReplicaLag

Intento: questo allarme è in grado di rilevare il ritardo di replica che riflette la perdita di dati che potrebbe verificarsi in caso di errore dell'istanza principale. Se la replica è troppo indietro rispetto alla principale e la principale riscontra errori, alla replica mancheranno i dati presenti nell'istanza principale.

Statistica: Maximum

Soglia raccomandata: 60,0

Giustificazione della soglia: in genere, il ritardo accettabile dipende dall'applicazione. Si consiglia di non superare i 60 secondi.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: GREATER_THAN_THRESHOLD

WriteLatency

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare una latenza di scrittura elevata. Se la latenza dell'archiviazione è elevata, il carico di lavoro supera i limiti delle risorse. È possibile

esaminare l'utilizzo dell'I/O in relazione alla configurazione dell'istanza e dell'archiviazione allocata. Fai riferimento alla risoluzione dei problemi di latenza dei volumi Amazon EBS causata da un collo di bottiglia IOPS. Per Aurora, puoi passare a una classe di istanza con una configurazione di archiviazione con ottimizzazione per I/O. Per informazioni, consulta Pianificazione dell'I/O in Aurora.

Intento: questo allarme viene utilizzato per rilevare una latenza di scrittura elevata. Sebbene i dischi del database abbiano in genere una bassa latenza di lettura/scrittura, possono presentare problemi che causano operazioni a latenza elevata. Il monitoraggio della latenza di scrittura garantirà che la latenza del disco sia bassa come previsto.

Statistica: p90

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dal caso d'uso. Le latenze di scrittura superiori a 20 millisecondi richiedono probabilmente un'analisi. È inoltre possibile impostare una soglia più alta se l'applicazione può avere una latenza più elevata per le operazioni di scrittura. Esamina la criticità e i requisiti della latenza di scrittura e analizza il comportamento storico di questo parametro per determinare livelli di soglia ragionevoli.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

DBLoad

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare un carico del database elevato. Se il numero di processi supera il numero di vCPUs, i processi vengono messi in coda. Quando la coda aumenta, le prestazioni diminuiscono. Se il carico è spesso sopra la vCPU massima e lo stato di attesa primario è CPU, la CPU è sovraccarica. In questo caso, puoi monitorare `CPUUtilization`, `DBLoadCPU` e attività in coda in Performance Insights/Enhanced Monitoring. Si potrebbero limitare le connessioni all'istanza, ottimizzare le eventuali query SQL con un elevato carico CPU o valutare la possibilità di una classe istanza di maggiori dimensioni. Istanze elevate

e costanti di qualsiasi stato di attesa indicano che possono verificarsi colli di bottiglia o problemi di conflitto delle risorse da risolvere.

Scopo: questo allarme viene utilizzato per rilevare un elevato carico del database. Un carico del database elevato può causare problemi di prestazioni nell'istanza di database. Questo allarme non è applicabile alle istanze di database serverless.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore vCPU massimo è determinato dal numero di core vCPU (CPU virtuale) per l'istanza database. A seconda della vCPU massima, possono essere appropriati valori diversi per la soglia. Idealmente, il carico del database non dovrebbe superare la linea vCPU.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

AuroraVolumeBytesLeftTotal

Dimensioni: DB ClusterIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare uno scarso livello di volume totale residuo. Quando il volume totale rimasto raggiunge il limite di dimensione, il cluster segnala un out-of-space errore. L'archiviazione di Aurora viene automaticamente dimensionata con i dati del volume cluster e si espande fino a 128 TiB o 64 TiB a seconda della versione del motore del database. Valuta la possibilità di eliminare le tabelle e i database che non sono più necessari. Per ulteriori informazioni, controlla il dimensionamento dell'archiviazione.

Intento: questo allarme viene utilizzato per rilevare quanto manca al cluster Aurora per raggiungere il limite delle dimensioni del volume. Questo allarme può prevenire un out-of-space errore che si verifica quando lo spazio del cluster si esaurisce. Questo allarme è consigliato solo per Aurora MySQL.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: è necessario calcolare il 10%-20% del limite delle dimensioni corrente in base alla velocità e all'andamento dell'aumento dell'utilizzo del volume, quindi usare tale risultato come valore di soglia per intervenire in modo proattivo prima che il volume raggiunga il limite.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: LESS_THAN_THRESHOLD

AuroraBinlogReplicaLag

Dimensioni: DBClusterIdentifier, role=Writer

Descrizione dell'allarme: questo allarme aiuta a monitorare lo stato di errore della replica dell'istanza di scrittura di Aurora. Per ulteriori informazioni, consulta Replica dei cluster DB Aurora MySQL tra le regioni. AWS Per la risoluzione dei problemi, consulta i roblemi di replica di Aurora MySQL.

Intento: questo allarme viene utilizzato per rilevare se l'istanza di scrittura è in uno stato di errore e non è in grado di replicare l'origine. Questo allarme è consigliato solo per Aurora MySQL.

Statistica: Average

Soglia raccomandata: -1,0

Giustificazione della soglia: si consiglia di utilizzare -1 come valore di soglia perché Aurora MySQL pubblica questo valore se la replica è in uno stato di errore.

Periodo: 60

Punti dati su cui attivare allarmi: 2

Periodi di valutazione: 2

Operatore di confronto: LESS_THAN_OR_EQUAL_TO_THRESHOLD

BlockedTransactions

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare un elevato numero di transazioni bloccate in un'istanza di database Aurora. Le transazioni bloccate possono terminare con un rollback o un commit. L'elevata simultaneità, le transazioni inattive o le transazioni di lunga durata possono portare al blocco delle transazioni. Per la risoluzione dei problemi, consulta la documentazione di Aurora MySQL.

Intento: questo allarme viene utilizzato per rilevare un numero elevato di transazioni bloccate in un'istanza di database Aurora al fine di prevenire i rollback delle transazioni e il peggioramento delle prestazioni.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: è necessario calcolare il 5% di tutte le transazioni dell'istanza utilizzando il parametro `ActiveTransactions` e utilizzare tale risultato come valore di soglia. Puoi anche esaminare la criticità e i requisiti delle transazioni bloccate e analizzare il comportamento storico di questo parametro per determinare livelli di soglia ragionevoli.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

BufferCacheHitRatio

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme consente di monitorare un rapporto di riscontri nella cache costantemente basso del cluster Aurora. Una percentuale di riscontri bassa indica che le query su questa istanza database vengono spesso trasferite su disco. Per la risoluzione del problemi, esamina il carico di lavoro per vedere quali query causano questo comportamento e controlla il documento Suggerimenti relativi alla RAM per un'istanza di database.

Intento: questo allarme viene utilizzato per rilevare un rapporto di riscontri nella cache costantemente basso al fine di prevenire una riduzione sostenuta delle prestazioni nell'istanza Aurora.

Statistica: Average

Soglia raccomandata: 80,0

Giustificazione della soglia: è possibile impostare la soglia per la percentuale di riscontri nella cache del buffer all'80%. Tuttavia, puoi regolare questo valore in base al livello di prestazioni e alle caratteristiche del carico di lavoro accettabili.

Periodo: 60

Punti dati su cui attivare allarmi: 10

Periodi di valutazione: 10

Operatore di confronto: LESS_THAN_THRESHOLD

EngineUptime

Dimensioni: DBClusterIdentifier, Role=Writer

Descrizione dell'allarme: questo allarme aiuta a monitorare tempi di inattività ridotti dell'istanza database di scrittura. L'istanza database di scrittura può interrompersi a causa di un riavvio, una manutenzione, un aggiornamento o un failover. Se l'uptime raggiunge 0 a causa di un failover nel cluster e il cluster ha una o più repliche Aurora, una replica Aurora Replica viene promossa all'istanza di scrittura principale durante un evento di errore. Per aumentare la disponibilità del cluster database, valuta la possibilità di creare almeno una o più repliche Aurora in due o più due zone di disponibilità. Per ulteriori informazioni, controlla i fattori che influenzano i tempi di inattività di Aurora.

Intento: questo allarme viene utilizzato per rilevare se l'istanza database di scrittura di Aurora è in fase di inattività. In questo modo è possibile evitare errori di lunga durata nell'istanza di scrittura dovuti a un arresto anomalo o a un failover.

Statistica: Average

Soglia raccomandata: 0,0

Giustificazione della soglia: un evento di errore ha come conseguenza una breve interruzione, durante la quale le operazioni di lettura e scrittura falliscono con un'eccezione. Tuttavia, il servizio viene in genere ripristinato in meno di 60 secondi e spesso in meno di 30 secondi.

Periodo: 60

Punti dati su cui attivare allarmi: 2

Periodi di valutazione: 2

Operatore di confronto: LESS_THAN_OR_EQUAL_TO_THRESHOLD

RollbackSegmentHistoryListLength

Dimensioni: DB InstanceIdentifier

Descrizione dell'allarme: questo allarme aiuta a monitorare una costante elevata lunghezza della cronologia dei segmenti di rollback di un'istanza Aurora. Una lunghezza dell'elenco della cronologia di InnoDB elevata indica che un grande numero di precedenti versioni di riga, la chiusura delle query e l'arresto del database sono diventati più lenti. Per ulteriori informazioni e risoluzione dei problemi, consulta la documentazione relativa all'aumento significativo della lunghezza dell'elenco della cronologia di InnoDB.

Intento: questo allarme viene utilizzato per rilevare una costante elevata lunghezza della cronologia dei segmenti di rollback. Può aiutarti a impedire un peggioramento sostenuto delle prestazioni e un elevato utilizzo della CPU nell'istanza Aurora. Questo allarme è consigliato solo per Aurora MySQL.

Statistica: Average

Soglia raccomandata: 1000000,0

Giustificazione della soglia: l'impostazione di questa soglia a 1 milione dovrebbe darti il tempo di esaminare il problema. Tuttavia, puoi regolare questo valore in base al livello di prestazioni e alle caratteristiche del carico di lavoro accettabili.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

StorageNetworkThroughput

Dimensioni: DBClusterIdentifier, Role=Writer

Descrizione dell'allarme: questo allarme aiuta a monitorare l'elevata velocità di trasmissione effettiva della rete di archiviazione. Se la velocità di trasmissione effettiva della rete di archiviazione supera la larghezza di banda di rete totale dell'istanza EC2, la latenza di lettura e

scrittura può essere elevata e causare un peggioramento delle prestazioni. Puoi controllare il tipo di istanza EC2 dalla Console. AWS Per la risoluzione dei problemi, controlla eventuali modifiche alle latenze di scrittura/lettura e valuta se hai attivato un allarme anche per questo parametro. In tal caso, esamina lo schema del carico di lavoro durante i momenti in cui è stato attivato l'allarme. Questo può aiutarti a capire se puoi ottimizzare il tuo carico di lavoro per ridurre la quantità totale di traffico di rete. Se ciò non è possibile, potresti dover valutare la possibilità di ridimensionare l'istanza.

Intento: questo allarme viene utilizzato per rilevare un'elevata velocità di trasmissione effettiva della rete di archiviazione. Il rilevamento di una elevata velocità di trasmissione effettiva può prevenire perdite di pacchetti di rete e un peggioramento delle prestazioni.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: è necessario calcolare circa l'80%-90% della larghezza di banda della rete totale del tipo di istanza EC2 e quindi utilizzare tale risultato come valore di soglia, in modo da agire proattivamente prima che i pacchetti di rete siano coinvolti. Puoi anche esaminare la criticità e i requisiti della velocità di trasmissione effettiva della rete di archiviazione e analizzare il comportamento storico di questo parametro per determinare livelli di soglia ragionevoli.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon Route 53 Public Data Plane

HealthCheckStatus

Dimensioni: HealthCheckId

Descrizione dell'allarme: questo allarme aiuta a rilevare gli endpoint non integri secondo gli strumenti di controllo dello stato. Per comprendere il motivo di un errore che causa uno stato di malfunzionamento, utilizza la scheda Strumenti di controllo dell'integrità nella Console di controllo dell'integrità di Route 53 per visualizzare lo stato di ciascuna regione e l'ultimo errore del controllo

dell'integrità. La scheda di stato mostra anche il motivo per cui l'endpoint viene segnalato come non integro. Consulta la procedura per la risoluzione dei problemi.

Scopo: questo allarme utilizza gli strumenti di controllo dell'integrità di Route53 per rilevare endpoint non integri.

Statistica: Average

Soglia raccomandata: 1,0

Giustificazione della soglia: lo stato dell'endpoint è riportato come 1 quando è integro. Tutti i valori inferiori a 1 equivalgono a non integro.

Periodo: 60

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: LESS_THAN_THRESHOLD

Amazon S3

4xxErrors

Dimensioni:BucketName, FilterId

Descrizione dell'allarme: questo allarme contribuisce a segnalare il numero totale di codici di stato di errore 4xx creati in risposta alle richieste dei client. I codici di errore 403 possono indicare una policy IAM errata, mentre i codici di errore 404 possono segnalare, ad esempio, un comportamento errato dell'applicazione client. L'abilitazione temporanea della registrazione degli accessi al server S3 ti aiuterà a individuare l'origine del problema utilizzando i campi Stato HTTP e Codice errore. Per ulteriori informazioni sul codice di errore, consulta la pagina Error Responses.

Scopo: questo allarme viene utilizzato per creare un valore di base per i tassi di errore 4xx tipici, in modo da poter esaminare eventuali anomalie che potrebbero indicare un problema di configurazione.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: la soglia raccomandata consiste nel rilevare se più del 5% del totale delle richieste presenta errori 4XX. È consigliabile attivare gli allarmi per gli errori 4XX che si verificano di frequente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile. Inoltre, è possibile regolare la soglia in base al carico delle richieste, tenendo conto del livello accettabile di errori 4XX. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

5xxErrors

Dimensioni:BucketName, FilterId

Descrizione dell'allarme: questo allarme aiuta a rilevare un numero elevato di errori lato server. Questi errori indicano che un client ha effettuato una richiesta che il server non è riuscito a completare. Questo può aiutarti a stabilire un collegamento con il problema che la tua applicazione sta riscontrando a causa di S3. Per ulteriori informazioni su come gestire o ridurre in modo efficiente gli errori, consulta la pagina Optimizing performance design patterns. Gli errori potrebbero essere causati anche da un problema con S3; controlla lo stato di Amazon S3 nella tua regione nel Pannello di controllo per lo stato dei servizi AWS.

Scopo: questo allarme può aiutare a rilevare se l'applicazione presenta problemi dovuti a errori 5xx.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: consigliamo di impostare la soglia per rilevare se più del 5% del totale delle richieste riceve 5XXError. Tuttavia, puoi regolare la soglia in base al traffico relativo alle richieste e ai tassi di errore accettabili. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

OperationsFailedReplication

Dimensioni:SourceBucket, DestinationBucket, RuleId

Descrizione dell'allarme: questo allarme aiuta a comprendere un errore di replica. Questo parametro tiene traccia dello stato dei nuovi oggetti replicati utilizzando S3 CRR o S3 SRR così come degli oggetti esistenti replicati utilizzando la replica in batch S3. Per ulteriori dettagli, consulta la sezione Replication troubleshooting.

Scopo: questo allarme viene utilizzato per rilevare se è presente un'operazione di replica non riuscita.

Statistica: Maximum

Soglia raccomandata: 0,0

Giustificazione della soglia: questo parametro emette un valore pari a 0 per le operazioni riuscite e nulla quando non vengono eseguite operazioni di replica per un minuto. Quando il parametro emette un valore maggiore di 0, l'operazione di replica non ha avuto esito positivo.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

S3ObjectLambda

4xxErrors

Dimensioni:AccessPointName, DataSource ARN

Descrizione dell'allarme: questo allarme aiuta a segnalare il numero totale di codici di stato di errore 4xx creati in risposta alle richieste dei client. L'abilitazione temporanea della registrazione

degli accessi al server S3 ti aiuterà a individuare l'origine del problema utilizzando i campi Stato HTTP e Codice errore.

Scopo: questo allarme viene utilizzato per creare un valore di base per i tassi di errore 4xx tipici, in modo da poter esaminare eventuali anomalie che potrebbero indicare un problema di configurazione.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: consigliamo di impostare la soglia per rilevare se più del 5% del totale delle richieste riceve 4XXError. È consigliabile attivare gli allarmi per gli errori 4XX che si verificano di frequente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile. Inoltre, è possibile regolare la soglia in base al carico delle richieste, tenendo conto del livello accettabile di errori 4XX. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

5xxErrors

Dimensioni:AccessPointName, DataSource ARN

Descrizione dell'allarme: questo allarme aiuta a rilevare un numero elevato di errori lato client. Questi errori indicano che un client ha effettuato una richiesta che il server non è riuscito a completare. Questi errori potrebbero essere causati anche da un problema con S3; controlla lo stato di Amazon S3 nella tua regione nel Pannello di controllo per lo stato dei servizi AWS. Questo può aiutarti a stabilire un collegamento con il problema che la tua applicazione sta riscontrando a causa di S3. Per informazioni su come gestire o ridurre in modo efficiente questi errori, consulta la pagina Optimizing performance design patterns.

Scopo: questo allarme può aiutare a rilevare se l'applicazione presenta problemi dovuti a errori 5xx.

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: consigliamo di impostare la soglia per rilevare se più del 5% del totale delle richieste riceve errori 5XX. Tuttavia, puoi regolare la soglia in base al traffico relativo alle richieste e ai tassi di errore accettabili. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

LambdaResponse4xx

Dimensioni:AccessPointName, DataSource ARN

Descrizione dell'allarme: questo allarme consente di rilevare e diagnosticare gli errori (500) nelle chiamate a S3 Object Lambda. Questi errori possono essere causati da errori o configurazioni errate nella funzione Lambda responsabile della risposta alle richieste. L'analisi dei flussi di CloudWatch log della funzione Lambda associata all'access point Object Lambda può aiutarti a individuare l'origine del problema in base alla risposta di S3 Object Lambda.

Intento: questo allarme viene utilizzato per rilevare gli errori del client 4xx per le chiamate. WriteGetObjectResponse

Statistica: Average

Soglia raccomandata: 0,05

Giustificazione della soglia: consigliamo di impostare la soglia per rilevare se più del 5% del totale delle richieste riceve 4XXError. È consigliabile attivare gli allarmi per gli errori 4XX che si verificano di frequente. Tuttavia, l'impostazione di un valore molto basso per la soglia può rendere l'allarme troppo sensibile. Inoltre, è possibile regolare la soglia in base al carico delle richieste, tenendo conto del livello accettabile di errori 4XX. Inoltre, è possibile analizzare i dati storici per determinare il tasso di errore accettabile per il carico di lavoro dell'applicazione e regolare la soglia di conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon SNS

NumberOfMessagesPublished

Dimensioni: TopicName

Descrizione dell'allarme: questo allarme può rilevare quando il numero di messaggi SNS pubblicati è troppo basso. Per la risoluzione dei problemi, controlla perché gli editori inviano una quantità minore di traffico.

Scopo: questo allarme aiuta a monitorare e rilevare in modo proattivo cali significativi nella pubblicazione delle notifiche. Ciò consente di identificare potenziali problemi con l'applicazione o i processi aziendali, in modo da intraprendere le operazioni appropriate per mantenere il flusso di notifiche previsto. È necessario creare questo allarme se si prevede che il sistema debba servire un traffico minimo.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il numero di messaggi pubblicati deve essere in linea con il numero previsto di messaggi pubblicati per l'applicazione. Inoltre, è possibile analizzare i dati storici, le tendenze e il traffico per individuare la soglia giusta.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: LESS_THAN_THRESHOLD

NumberOfNotificationsDelivered

Dimensioni: TopicName

Descrizione dell'allarme: questo allarme può rilevare quando il numero di messaggi SNS consegnati è troppo basso. Ciò potrebbe essere dovuto all'annullamento involontario dell'abbonamento di un endpoint o a un evento SNS che causa un ritardo nei messaggi.

Scopo: questo allarme consente di rilevare un calo del volume dei messaggi consegnati. È necessario creare questo allarme se si prevede che il sistema debba servire un traffico minimo.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il numero di messaggi consegnati deve essere in linea con il numero previsto di messaggi prodotti e il numero di consumatori. Inoltre, è possibile analizzare i dati storici, le tendenze e il traffico per individuare la soglia giusta.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: LESS_THAN_THRESHOLD

NumberOfNotificationsFailed

Dimensioni: TopicName

Descrizione dell'allarme: questo allarme può rilevare quando il numero di messaggi SNS consegnati è troppo alto. Per risolvere i problemi relativi alle notifiche non riuscite, abilita la registrazione nei registri. CloudWatch L'esame dei log può aiutarti a scoprire quali abbonati non funzionano e i codici di stato che restituiscono.

Scopo: questo allarme ti aiuta a individuare in modo proattivo i problemi relativi alla consegna delle notifiche e a prendere le misure necessarie per risolverli.

Statistica: Sum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dall'impatto delle notifiche non riuscite. Esamina gli SLA forniti agli utenti finali, la tolleranza agli errori e la criticità delle notifiche, analizza i dati storici e seleziona una soglia di

conseguenza. Il numero di notifiche non riuscite dovrebbe essere 0 per gli argomenti che hanno solo abbonamenti SQS, Lambda o Firehose.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidAttributes

Dimensioni: TopicName

Descrizione dell'allarme: questo allarme aiuta a monitorare e risolvere potenziali problemi con l'editore o gli abbonati. Controlla se un editore pubblica messaggi con attributi non validi o se a un abbonato viene applicato un filtro inappropriato. Puoi anche analizzare CloudWatch i log per individuare la causa principale del problema.

Scopo: l'allarme viene utilizzato per rilevare se i messaggi pubblicati non sono validi o se a un abbonato sono stati applicati filtri inappropriati.

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: gli attributi non validi sono quasi sempre un errore dell'editore. Si consiglia di impostare la soglia su 0 perché in un sistema integro non sono previsti attributi non validi.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidMessageBody

Dimensioni: TopicName

Descrizione dell'allarme: questo allarme aiuta a monitorare e risolvere potenziali problemi con l'editore o gli abbonati. Controlla se un editore pubblica messaggi con corpi dei messaggi non validi o se a un abbonato viene applicato un filtro inappropriato. Puoi anche analizzare CloudWatch i log per individuare la causa principale del problema.

Scopo: l'allarme viene utilizzato per rilevare se i messaggi pubblicati non sono validi o se a un abbonato sono stati applicati filtri inappropriati.

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: i corpi dei messaggi non validi sono quasi sempre un errore dell'editore. Si consiglia di impostare la soglia su 0 perché in un sistema integro non sono previsti messaggi non validi.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

NumberOfNotificationsRedrivenToDlq

Dimensioni: TopicName

Descrizione dell'allarme: questo allarme aiuta a monitorare il numero di messaggi che vengono spostati in una coda DLQ.

Scopo: l'allarme viene utilizzato per rilevare i messaggi che vengono spostati in una coda DLQ. Ti consigliamo di creare questo allarme quando SNS è abbinato a SQS, Lambda o Firehose.

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: in un sistema integro, indipendentemente dal tipo di abbonato, i messaggi non devono essere spostati nella coda DLQ. Ti consigliamo di ricevere una notifica se qualche messaggio finisce nella coda, in modo da poter identificare e risolvere la causa principale e, potenzialmente, reindirizzare i messaggi nella coda DLQ per evitare la perdita di dati.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

NumberOfNotificationsFailedToRedriveToDlq

Dimensioni: TopicName

Descrizione dell'allarme: questo allarme aiuta a monitorare i messaggi che non possono essere spostati in una coda DLQ. Controlla se la tua coda DLQ esiste e che sia configurata correttamente. Inoltre, verifica che SNS disponga delle autorizzazioni per accedere alla coda DLQ. Per ulteriori informazioni, consulta la documentazione sulle code DLQ.

Scopo: l'allarme viene utilizzato per rilevare i messaggi che non possono essere spostati in una coda DLQ.

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: è quasi sempre un errore se i messaggi non possono essere spostati nella coda DLQ. La soglia raccomandata è 0, il che significa che tutti i messaggi che non vengono elaborati devono essere in grado di raggiungere la coda DLQ quando la coda è stata configurata.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

SMS MonthToDateSpent USD

Dimensioni: TopicName

Descrizione dell'allarme: l'allarme aiuta a controllare se nell'account è disponibile una quota sufficiente per consentire a SNS di consegnare i messaggi. Se raggiungi la quota, SNS non sarà in grado di consegnare messaggi SMS. Per informazioni sull'impostazione della quota di spesa

mensile per gli SMS o per informazioni sulla richiesta di un aumento della quota di spesa con AWS, consulta Impostazione delle preferenze per i messaggi SMS.

Scopo: questo allarme viene utilizzato per rilevare se la quota dell'account è sufficiente per garantire la corretta consegna dei tuoi messaggi SMS.

Statistica: Maximum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia in base alla quota (limite di spesa dell'account) per l'account. Scegli una soglia che ti informi con sufficiente anticipo del raggiungimento del limite di quota, in modo da avere il tempo di richiedere un aumento.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

SMS SuccessRate

Dimensioni: TopicName

Descrizione dell'allarme: questo allarme aiuta a monitorare il tasso di mancata consegna dei messaggi SMS. Puoi configurare File di log CloudWatch per comprendere la natura dell'errore e agire di conseguenza.

Scopo: questo allarme viene utilizzato per rilevare le mancate consegne di messaggi SMS.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: imposta la soglia per l'allarme in base alla tua tolleranza in caso di mancata consegna dei messaggi SMS.

Periodo: 60

Punti dati su cui attivare allarmi: 5

Periodi di valutazione: 5

Operatore di confronto: GREATER_THAN_THRESHOLD

Amazon SQS

ApproximateAgeOfOldestMessage

Dimensioni: QueueName

Descrizione dell'allarme: questo allarme rileva l'età del messaggio più vecchio nella coda. È possibile utilizzare questo allarme per verificare se i consumatori elaborano i messaggi SQS alla velocità desiderata. Valuta la possibilità di aumentare il numero di consumatori o la velocità di trasmissione effettiva degli stessi per ridurre l'età dei messaggi. Questo parametro può essere utilizzato in combinazione con `ApproximateNumberOfMessagesVisible` per determinare l'entità del backlog della coda e la velocità di elaborazione dei messaggi. Per evitare che i messaggi vengano eliminati prima dell'elaborazione, prendi in considerazione la possibilità di configurare la coda DLQ in modo da mettere da parte i potenziali messaggi avvelenati.

Intento: questo allarme viene utilizzato per rilevare se l'età del messaggio più vecchio nella QueueName coda è troppo alta. L'età elevata può indicare che i messaggi non vengono elaborati abbastanza velocemente o che alcuni messaggi avvelenanti sono bloccati in coda e non possono essere elaborati.

Statistica: Maximum

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dal tempo previsto per l'elaborazione del messaggio. È possibile utilizzare i dati storici per calcolare il tempo medio di elaborazione dei messaggi e quindi impostare la soglia su un valore del 50% superiore al tempo massimo di elaborazione dei messaggi SQS previsto dai consumatori in coda.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesNotVisible

Dimensioni: QueueName

Descrizione dell'allarme: questo allarme aiuta a rilevare un numero elevato di messaggi in transito relativamente a `QueueName`. Per la risoluzione dei problemi, consulta la sezione sulla riduzione del backlog dei messaggi.

Scopo: questo allarme viene utilizzato per rilevare un numero elevato di messaggi in transito nella coda. Se i consumatori non eliminano i messaggi entro il periodo di timeout di visibilità, quando avviene il polling della coda, i messaggi riappaiono nella coda. Per le code FIFO, possono esserci al massimo 20.000 messaggi in transito. Se raggiungi questa quota, SQS non restituisce alcun messaggio di errore. Una coda FIFO esamina i primi 20.000 messaggi per determinare i gruppi di messaggi disponibili. Ciò significa che se in un singolo gruppo di messaggi è presente un backlog, non è possibile utilizzare i messaggi provenienti da altri gruppi di messaggi inviati successivamente alla coda fino a quando non si gestisce correttamente il backlog.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: il valore di soglia raccomandato per questo allarme dipende in larga misura dal numero previsto di messaggi in transito. È possibile utilizzare i dati storici per calcolare il numero massimo previsto di messaggi in transito e impostare una soglia del 50% superiore a questo valore. Se i consumatori della coda elaborano i messaggi della coda ma non li eliminano, questo numero aumenterà improvvisamente.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesVisible

Dimensioni: QueueName

Descrizione dell'allarme: questo allarme rileva se il backlog della coda di messaggi è maggiore del previsto, indicando che i consumatori sono troppo lenti o che il loro numero è insufficiente. Se

l'allarme entra nello stato ALARM, valuta la possibilità di aumentare il numero di consumatori o di velocizzarli.

Scopo: questo allarme viene utilizzato per rilevare se il numero di messaggi della coda attiva è troppo elevato e i consumatori sono lenti nell'elaborazione dei messaggi o se non sono sufficienti per la loro elaborazione.

Statistica: Average

Soglia raccomandata: dipende dalla situazione

Giustificazione della soglia: un numero inaspettatamente elevato di messaggi visibili indica che i messaggi non vengono elaborati da un consumatore alla velocità prevista. È consigliabile analizzare i dati storici per impostare questa soglia.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: GREATER_THAN_OR_EQUAL_TO_THRESHOLD

NumberOfMessagesSent

Dimensioni: QueueName

Descrizione dell'allarme: questo allarme aiuta a rilevare se non ci sono messaggi inviati da un produttore in merito a `QueueName`. Per la risoluzione dei problemi, controlla il motivo per cui il produttore non invia messaggi.

Scopo: questo allarme viene utilizzato per rilevare quando un produttore interrompe l'invio di messaggi.

Statistica: Sum

Soglia raccomandata: 0,0

Giustificazione della soglia: se il numero di messaggi inviati è 0, il produttore non invia alcun messaggio. Se questa coda ha un TPS basso, aumenta il numero di EvaluationPeriods conseguenza.

Periodo: 60

Punti dati su cui attivare allarmi: 15

Periodi di valutazione: 15

Operatore di confronto: LESS_THAN_OR_EQUAL_TO_THRESHOLD

AWS VPN

TunnelState

Dimensioni: VpnId

Descrizione dell'allarme: questo allarme permette di capire se lo stato di uno o più tunnel è DOWN. Per la risoluzione dei problemi, consulta la pagina VPN tunnel troubleshooting.

Scopo: questo allarme viene utilizzato per rilevare se almeno un tunnel è in stato DOWN per questa VPN, in modo da poter risolvere i problemi relativi alla VPN interessata. Questo allarme sarà sempre nello stato ALARM per le reti in cui è configurato un solo tunnel.

Statistica: Minimum

Soglia raccomandata: 1,0

Giustificazione della soglia: un valore inferiore a 1 indica che almeno un tunnel è in stato DOWN.

Periodo: 300

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: LESS_THAN_THRESHOLD

TunnelState

Dimensioni: TunnelIpAddress

Descrizione dell'allarme: questo allarme permette di capire se lo stato di questo tunnel è DOWN. Per la risoluzione dei problemi, consulta la pagina VPN tunnel troubleshooting.

Scopo: questo allarme viene utilizzato per rilevare se il tunnel è in stato DOWN, in modo da poter risolvere i problemi relativi alla VPN interessata. Questo allarme sarà sempre nello stato ALARM per le reti in cui è configurato un solo tunnel.

Statistica: Minimum

Soglia raccomandata: 1,0

Giustificazione della soglia: un valore inferiore a 1 indica che il tunnel è in stato DOWN.

Periodo: 300

Punti dati su cui attivare allarmi: 3

Periodi di valutazione: 3

Operatore di confronto: LESS_THAN_THRESHOLD

Creazione di allarmi sui parametri

I passaggi delle sezioni seguenti spiegano come creare CloudWatch allarmi in base alle metriche.

Crea un CloudWatch allarme basato su una soglia statica

Scegli una CloudWatch metrica per la sveglia da guardare e la soglia per quella metrica. L'allarme passa nello stato `ALARM` quando il parametro supera la soglia per un numero specificato di periodi di valutazione.

Se stai creando un allarme in un account configurato come account di monitoraggio in modalità osservabilità CloudWatch tra account diversi, puoi impostare l'allarme in modo che controlli una metrica in un account di origine collegato a questo account di monitoraggio. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Per creare un allarme basato su un parametro singolo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All alarms (Tutti gli allarmi).
3. Scegli Crea allarme.
4. Scegli Select Metric (Seleziona parametro).
5. Esegui una di queste operazioni:
   - Scegli lo spazio dei nomi del servizio contenente il parametro desiderato. Continua scegliendo le opzioni così come vengono visualizzate per limitare le scelte. Quando viene

visualizzato un elenco di parametri, seleziona la casella di controllo accanto al parametro desiderato.

- Nella casella di ricerca immetti il nome di un parametro, l'ID account, l'etichetta dell'account, la dimensione o l'ID risorsa. Dopodiché, scegli uno dei risultati e continua finché non viene visualizzato un elenco di parametri. Seleziona la casella di controllo accanto al parametro desiderato.

6. Seleziona la scheda Graphed metrics (Parametri nel grafico).

   a. In Statistic (Statistiche), scegli una delle statistiche o percentili predefiniti oppure specifica un percentile personalizzato (ad esempio, **`p95.45`**).

   b. In Period (Periodo), scegli il periodo di valutazione per l'allarme. Durante la valutazione dell'allarme, ogni periodo è aggregato in un punto dati.

      Puoi anche scegliere se la legenda dell'asse Y viene visualizzata a sinistra o a destra durante la creazione dell'allarme. Questa preferenza viene utilizzata solo durante la creazione dell'allarme.

   c. Scegli Select Metric (Seleziona parametro).

      Viene visualizzata la pagina Specify metric and conditions (Specifica parametro e condizioni), contenente un grafico e altre informazioni sul parametro e le statistiche selezionate.

7. In Conditions (Condizioni), specifica quanto segue:

   a. Per Whenever ***`metric`*** is (Ogni volta che il parametro è), specifica se il parametro deve essere maggiore di, minore di o uguale alla soglia. In than... (che...), specifica il valore di soglia.

   b. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

      Per creare un allarme M di N, specifica un numero minore per il primo valore rispetto a quello specificato per il secondo valore. Per ulteriori informazioni, consulta la pagina Valutazione di un allarme.

   c. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .

d. Se l'allarme utilizza un percentile come statistica monitorata, viene visualizzata una casella Percentiles with low samples (Percentili con campioni ridotti). Utilizzala per scegliere se valutare o ignorare casi con bassa frequenza di campionamento. Se scegli ignore (maintain alarm state) (ignora (mantieni stato dell'allarme)), lo stato corrente dell'allarme viene sempre mantenuto quando la dimensione dell'esempio è troppo bassa. Per ulteriori informazioni, consulta la pagina CloudWatch Allarmi basati su percentili ed esempi di dati limitati.

8. Seleziona Next (Successivo).
9. In Notification (Notifica), seleziona un argomento SNS per segnalare quando l'allarme è nello stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

   Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

   Nell' CloudWatch osservabilità tra più account, puoi scegliere di inviare notifiche a più AWS account. Ad esempio, sia all'account di monitoraggio che all'account di origine.

   Per fare in modo che l'allarme non invii notifiche, scegli Remove (Rimuovi).
10. Per fare in modo che l'allarme esegua operazioni Auto Scaling, EC2, Lambda o Systems Manager scegli il pulsante appropriato e scegli lo stato di allarme e l'operazione da eseguire. Gli allarmi possono eseguire le operazioni Systems Manager solo quando entrano nello stato ALARM. Per ulteriori informazioni sulle azioni di Systems Manager, vedere Configurazione per la creazione CloudWatch OpsItems da allarmi e Creazione di incidenti.

    > **Note**
    >
    > Per creare un allarme che esegua un'operazione SSM Incident Manager, è necessario disporre di determinate autorizzazioni. Per ulteriori informazioni, vedere Esempi di policy basate sull'identità per AWS Systems Manager Incident Manager.

11. Al termine, scegli Apply (Applica).
12. Inserisci un nome e una descrizione per l'allarme. Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne. Quindi scegli Successivo.
13. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Puoi anche aggiungere allarmi a un pannello di controllo. Per ulteriori informazioni, consulta la pagina Aggiungere o rimuovere un widget di allarme da una CloudWatch dashboard .

Crea un CloudWatch allarme basato su un'espressione matematica metrica

Per creare un allarme basato su un'espressione matematica metrica, scegli una o più CloudWatch metriche da utilizzare nell'espressione. Quindi, specifica l'espressione, la soglia e i periodi di valutazione.

Non puoi creare un allarme in base all'espressione SEARCH. Questo perché le espressioni di ricerca restituiscono più serie temporali e un allarme basato su un'espressione matematica può osservare solo una serie temporale.

Creazione di un allarme basato su un'espressione matematica del parametro

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), quindi Create alarm (Tutti gli allarmi).
3. Scegli Crea allarme.
4. Scegli Select Metric (Seleziona parametro) ed esegui una delle operazioni seguenti:
   - Seleziona uno spazio dei nomi dal menu a discesa namespaces (Spazi dei nomi)AWS o dal menu a discesa Custom namespaces (Spazi dei nomi personalizzati). Dopo aver selezionato uno spazio dei nomi, continua a scegliere le opzioni finché non viene visualizzato un elenco di parametri; seleziona quindi la casella di controllo accanto al parametro corretto.
   - Utilizza la casella di ricerca per trovare un parametro, un ID account, una dimensione o un ID risorsa. Dopo aver inserito il parametro, la dimensione o l'ID risorsa, continua a scegliere le opzioni finché non viene visualizzato un elenco di parametri; seleziona quindi la casella di controllo accanto al parametro corretto.
5. (Facoltativo) Se si desidera aggiungere un altro parametro a un'espressione matematica del parametro, puoi utilizzare la casella di ricerca per trovarne uno specifico. È possibile aggiungere un massimo di dieci parametri a un'espressione matematica del parametro.
6. Seleziona la scheda Graphed metrics (Parametri nel grafico). Per ogni parametro aggiunto in precedenza, svolgi le seguenti operazioni:

   a. Nella colonna Statistic (Statistiche), seleziona il menu a discesa. Nel menu a discesa, scegli una delle statistiche o uno dei percentili predefiniti. Utilizza la casella di ricerca nel menu a discesa per definire un percentile personalizzato.

b. Nella colonna Period (Periodo), seleziona il menu a discesa. Nel menu a discesa, scegli uno dei periodi di valutazione predefiniti.

   Durante la creazione dell'allarme, puoi definire se la legenda dell'asse Y viene visualizzata a sinistra o a destra del grafico.

> **Note**
>
> Quando CloudWatch valuta gli allarmi, i periodi vengono aggregati in singoli punti dati.

7. Scegli il menu a discesa Add math (Aggiungi matematica), quindi seleziona Start with an empty expression (Inizia con un'espressione vuota) nell'elenco delle espressioni matematiche predefinite del parametro.

   Dopo aver scelto Start with an empty expression (Inizia con un'espressione vuota), viene visualizzata un riquadro in cui puoi applicare o modificare le espressioni matematiche.

8. Inserisci l'espressione matematica nel riquadro, quindi scegli Apply (Applica).

   Dopo aver scelto Apply (Applica), viene visualizzata una colonna ID accanto alla colonna Label (Etichetta).

   Per usare un parametro o il risultato di un'altra espressione matematica del parametro nella formula dell'espressione matematica attuale, utilizza il valore illustrato nella colonna ID. Per modificare il valore di ID, seleziona l' pen-and-paper icona accanto al valore corrente. Il nuovo valore deve iniziare con una lettera minuscola e può includere numeri, lettere e il trattino basso. Modificare il valore di ID con un nome più significativo può rendere il grafico di allarme più comprensibile.

   Per informazioni sulle funzioni disponibili per la matematica del parametro, consulta la pagina Sintassi e funzioni della matematica dei parametri.

9. (Facoltativo) Aggiungi altre espressioni matematiche, utilizzando i parametri e i risultati di altre espressioni matematiche nelle formule delle nuove espressioni matematiche.

10. Quando disponi dell'espressione da utilizzare per l'allarme, deseleziona le caselle di controllo a sinistra di ogni altra espressione e ogni parametro sulla pagina. Solo la casella di controllo accanto all'espressione da utilizzare per l'allarme deve essere selezionata. L'espressione scelta per l'allarme deve produrre una singola serie temporale e visualizzare una sola linea sul grafico. Quindi, scegli Select metric (Seleziona parametro).

Viene visualizzata la pagina Specify metric and conditions (Specifica parametro e condizioni), contenente un grafico e altre informazioni relative all'espressione matematica selezionata.

11. Per Whenever ***`expression`*** is (Ogni volta che espressione è), specifica se l'espressione deve essere maggiore di, minore di o uguale alla soglia. In than... (che...), specifica il valore di soglia.
12. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

    Per creare un allarme M di N, specifica un numero minore per il primo valore rispetto a quello specificato per il secondo valore. Per ulteriori informazioni, consulta la pagina Valutazione di un allarme.
13. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .
14. Seleziona Next (Successivo).
15. In Notification (Notifica), seleziona un argomento SNS per segnalare quando l'allarme è nello stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

    Per fare in modo che l'allarme non invii notifiche, scegli Remove (Rimuovi).
16. Per fare in modo che l'allarme esegua operazioni Auto Scaling, EC2, Lambda o Systems Manager scegli il pulsante appropriato e scegli lo stato di allarme e l'operazione da eseguire. Se scegli una funzione Lambda come operazione di allarme, specifichi il nome della funzione o l'ARN e, facoltativamente, puoi scegliere una versione specifica della funzione.

    Gli allarmi possono eseguire le operazioni Systems Manager solo quando entrano nello stato ALARM. Per ulteriori informazioni sulle azioni di Systems Manager, vedere Configurazione per la creazione CloudWatch OpsItems da allarmi e Creazione di incidenti.

> **Note**
>
> Per creare un allarme che esegua un'operazione SSM Incident Manager, è necessario disporre di determinate autorizzazioni. Per ulteriori informazioni, vedere Esempi di policy basate sull'identità per AWS Systems Manager Incident Manager.

17. Al termine, scegli Apply (Applica).
18. Inserisci un nome e una descrizione per l'allarme. Quindi scegli Successivo.

    Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.
19. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Puoi anche aggiungere allarmi a un pannello di controllo. Per ulteriori informazioni, consulta la pagina Aggiungere o rimuovere un widget di allarme da una CloudWatch dashboard .

Crea un CloudWatch allarme basato su una query di Metrics Insights

Puoi creare un allarme su qualsiasi query di Approfondimenti sulle metriche che restituisce una singola serie temporale. Ciò può essere particolarmente utile per creare allarmi dinamici che controllano le metriche aggregate su un parco istanze di infrastrutture o applicazioni. Crea l'allarme una volta e si regola man mano che le risorse vengono aggiunte o rimosse dal parco istanze. Ad esempio, puoi creare un allarme che controlla l'utilizzo della CPU di tutte le tue istanze e l'allarme si regola dinamicamente man mano che aggiungi o rimuovi istanze.

Per istruzioni complete, consulta Creazione di allarmi nelle query di Approfondimenti sulle metriche.

Creazione di un allarme basato su un'origine dati connessa

Puoi creare allarmi che controllano le metriche provenienti da fonti di dati che non sono presenti. CloudWatch Per ulteriori informazioni sulla creazione di connessioni a queste altre origini dati, consulta Recupero dei parametri da altre origini dati.

Per creare un allarme sui parametri da un'origine dati alla quale si è effettuata la connessione

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Metrics (Parametri), All metrics (Tutti i parametri).
3. Scegli la scheda Query da più origini.
4. Per Origine dati, seleziona l'origine dati che desideri utilizzare.
5. Il generatore di query richiede le informazioni necessarie affinché la query recuperi i parametri da utilizzare per l'allarme. Il flusso di lavoro è diverso per ogni origine dati ed è personalizzato in base a essa. Ad esempio, per le origini dati Amazon Managed Service for Prometheus e le origini dati Prometheus, viene visualizzata una casella di editor di query PromQL con una guida alle query.
6. Quando hai completato la creazione della query, scegli Query a grafo.
7. Se il grafico di esempio ha l'aspetto previsto, scegli Crea allarme.
8. Viene visualizzata la pagina Specifica parametro e condizioni. Se la query che stai usando produce più di una serie temporale, un banner di avviso viene visualizzato nella parte superiore della pagina. In tal caso, seleziona una funzione da utilizzare per aggregare le serie temporali nella funzione di aggregazione.
9. (Facoltativo) Aggiungi un'etichetta per l'allarme.
10. Per Whenever ***`your-metric-name`***is.. , scegli Maggiore, Maggiore/Uguale, Minore/Uguale o Inferiore. Per di . . ., specifica un numero per il valore di soglia.
11. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

    Per creare un allarme M di N, specifica un numero del primo valore inferiore a quello del secondo valore. Per ulteriori informazioni, consulta la pagina Valutazione di un allarme.
12. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .
13. Seleziona Next (Successivo).
14. Per Notifica, seleziona un argomento Amazon SNS per segnalare quando l'allarme passa allo stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

a. (Facoltativo) Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

> **Note**
>
> Ti consigliamo di impostare l'allarme in modo da intervenire quando entra in stato Dati insufficienti oltre a quando entra in stato Allarme. Questo perché molti problemi relativi alla funzione Lambda che si connette all'origine dati possono causare il passaggio dell'allarme a Dati insufficienti.

b. (Facoltativo) Per fare in modo che l'allarme non invii notifiche Amazon SNS, scegli Rimuovi.

15. Per fare in modo che l'allarme esegua operazioni Auto Scaling, EC2, Lambda o Systems Manager scegli il pulsante appropriato e scegli lo stato di allarme e l'operazione da eseguire. Se scegli una funzione Lambda come operazione di allarme, specifichi il nome della funzione o l'ARN e, facoltativamente, puoi scegliere una versione specifica della funzione.

    Gli allarmi possono eseguire le operazioni Systems Manager solo quando entrano nello stato ALARM. Per ulteriori informazioni sulle azioni di Systems Manager, vedere Configurazione per la creazione CloudWatch OpsItems da allarmi e Creazione di incidenti.

    > **Note**
    >
    > Per creare un allarme che esegua un'operazione SSM Incident Manager, è necessario disporre di determinate autorizzazioni. Per ulteriori informazioni, vedere Esempi di policy basate sull'identità per AWS Systems Manager Incident Manager.

16. Seleziona Successivo.

17. In Add a description (Aggiungere una descrizione), immetti un nome e una descrizione per l'allarme e scegli Next (Successivo). Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.

> **Tip**
>
> Il nome dell'allarme deve contenere solo caratteri UTF-8. Non può contenere caratteri di controllo ASCII.

18. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni dell'allarme sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Dettagli sugli allarmi per le origini dati connesse

- Quando CloudWatch valuta un allarme, lo fa ogni minuto, anche se il periodo dell'allarme è più lungo di un minuto. Affinché l'allarme funzioni, la funzione Lambda deve essere in grado di restituire un elenco di timestamp a partire da un minuto qualsiasi, non solo da multipli della durata del periodo. Questi timestamp devono essere distanziati di un periodo.

  Pertanto, se l'origine dati interrogata da Lambda può restituire solo timestamp multipli della durata del periodo, la funzione dovrebbe “ricampionare” i dati recuperati in modo che corrispondano ai timestamp previsti dalla richiesta. `GetMetricData`

  Ad esempio, un allarme con un periodo di cinque minuti viene valutato ogni minuto utilizzando finestre di cinque minuti che si spostano di un minuto ogni volta. In questo caso:

  - Per la valutazione dell'allarme alle 12:15:00, CloudWatch prevede punti dati con timestamp pari a, e. `12:00:00` `12:05:00` `12:10:00`
  - Quindi, per la valutazione dell'allarme alle 12:16, si CloudWatch aspetta punti dati con timestamp di, e. `12:01:00` `12:06:00` `12:11:00`
- Quando CloudWatch valuta un allarme, tutti i punti dati restituiti dalla funzione Lambda che non sono in linea con i timestamp previsti vengono eliminati e l'allarme viene valutato utilizzando i punti dati previsti rimanenti. Ad esempio, quando l'allarme viene valutato alle `12:15:00`, prevede dati con timestamp pari di `12:00:00`, `12:05:00` e `12:10:00`. Se riceve dati con timestamp pari a,, e `12:00:00` `12:05:00` `12:06:0012:10:00`, i dati da `12:06:00` vengono eliminati e valuta l'allarme utilizzando gli altri timestamp. CloudWatch

  Quindi, per la valutazione successiva alle `12:16:00`, prevede dati con timestamp di `12:01:00`, `12:06:00` e `12:11:00`. Se ha solo i dati con timestamp di `12:00:00`, `12:05:00` e `12:10:00`, tutti questi punti dati vengono ignorati alle 12:16:00 e l'allarme passa allo stato in base a come

è stato specificato che tratti i dati mancanti. Per ulteriori informazioni, consulta Valutazione di un allarme.

- Ti consigliamo di creare questi allarmi per intraprendere azioni durante la transizione allo stato `INSUFFICIENT_DATA`, poiché diversi casi d'uso di errori della funzione Lambda faranno passare l'allarme a `INSUFFICIENT_DATA` indipendentemente dal modo in cui è stato specificato che tratti i dati mancanti.
- Se la funzione Lambda restituisce un errore o dati parziali:
  - Se c'è un problema di autorizzazione con la chiamata alla funzione Lambda, l'allarme inizia ad presentare transizioni di dati mancanti in base a come è stato specificato che tratti i dati mancanti al momento della creazione.
  - Se la funzione Lambda restituisce `'StatusCode' = 'PartialData'`, la valutazione dell'allarme fallisce e l'allarme passa a `INSUFFICIENT_DATA` dopo tre tentativi. L'operazione richiede circa tre minuti.
  - Qualsiasi altro errore proveniente dalla funzione Lambda causa il passaggio dell'allarme a `INSUFFICIENT_DATA`.
- Se il parametro richiesto dalla funzione Lambda presenta un certo ritardo che provoca sempre la mancanza dell'ultimo punto dati, è necessario utilizzare una soluzione alternativa. È possibile creare un allarme M di N o aumentare il periodo di valutazione dell'allarme. Per ulteriori informazioni sugli allarmi M di N, consulta Valutazione di un allarme.

Crea un allarme basato sul rilevamento delle CloudWatch anomalie

È possibile creare un allarme basato sul rilevamento delle CloudWatch anomalie, che analizza i dati metrici passati e crea un modello di valori previsti. I valori previsti fanno riferimento ai pattern orari, giornalieri e settimanali standard a livello di parametri.

Si imposta un valore per la soglia di rilevamento delle anomalie e CloudWatch si utilizza tale soglia con il modello per determinare l'intervallo di valori «normale» per la metrica. Un valore più alto per la soglia produce un intervallo più ampio di valori "normali".

Puoi decidere se l'allarme viene attivato quando il valore del parametro è al di sopra dell'intervallo di valori previsti, si trova al di sotto di tale intervallo oppure è sopra o sotto l'intervallo.

È inoltre possibile creare allarmi di rilevamento delle anomalie su singole metriche e gli output delle espressioni matematiche dei parametri. È possibile utilizzare queste espressioni per creare grafici che visualizzano le bande di rilevamento delle anomalie.

In un account configurato come account di monitoraggio per l'osservabilità CloudWatch tra più account, puoi creare rilevatori di anomalie sulle metriche negli account di origine oltre alle metriche nell'account di monitoraggio.

Per ulteriori informazioni, consulta la pagina Utilizzo del CloudWatch rilevamento delle anomalie.

> **Note**
>
> Se si sta già utilizzando un rilevamento di anomalie a scopo di visualizzazione per un parametro nella console dei parametri , e su crea un allarme per il rilevamento delle anomali sullo stesso parametro, la soglia impostata per l'allarme non modifica la soglia già utilizzata per la visualizzazione. Per ulteriori informazioni, consulta la pagina Creazione di un grafico.

Per creare un allarme basato sul rilevamento di anomalie

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All alarms (Tutti gli allarmi).
3. Scegli Crea allarme.
4. Scegli Select Metric (Seleziona parametro).
5. Esegui una di queste operazioni:
   - Scegliere lo spazio dei nomi del servizio che contiene il parametro, quindi continuare a scegliere le opzioni man mano che appaiono per restringere le opzioni. Quando viene visualizzato un elenco di parametri, seleziona la casella di controllo accanto al parametro desiderato.
   - Nella casella di ricerca, inserisci il nome di un parametro, una dimensione o un ID risorsa . Quindi scegli uno dei risultati e continuare finché non viene visualizzato un elenco di parametri. Seleziona la casella di controllo accanto a un parametro.
6. Scegli Graphed Metric.
   a. (Facoltativo) Per Statistica, scegli il menu a discesa, quindi seleziona una delle statistiche o dei percentili predefiniti. È possibile utilizzare la casella di ricerca nel menu a discesa per definire un percentile personalizzato come **`p95.45`**.
   b. (Facoltativo) Per Periodo, scegli il menu a discesa, quindi seleziona uno dei periodi di valutazione predefiniti.

> **Note**
>
> Quando CloudWatch valuta l'allarme, aggrega il periodo in un singolo datapoint. Per gli allarmi basati sul rilevamento di anomalie, il valore deve essere maggiore o uguale a un minuto.

7. Seleziona Successivo.
8. In Conditions (Condizioni), specifica quanto segue:

   a. Scegli Anomaly detection (Rilevamento di anomalie).

   Se il modello per questa metrica e statistica esiste già, CloudWatch visualizza un'anteprima della banda di rilevamento delle anomalie nel grafico nella parte superiore dello schermo. Dopo aver creato l'allarme, affinché l'intervallo di rilevamento di anomalie effettivo venga visualizzato nel grafico possono essere necessari fino a 15 minuti. Prima di ciò, l'intervallo visualizzato è un'approssimazione dell'intervallo di rilevamento di anomalie.

   > **Tip**
   >
   > Per visualizzare il grafico nella parte superiore dello schermo in un lasso di tempo più lungo, scegli Edit (Modifica) in alto a destra della pagina.

   Se il modello per questa metrica e statistica non esiste già, CloudWatch genera la banda di rilevamento delle anomalie dopo aver completato la creazione dell'allarme. Per i nuovi modelli, possono essere necessarie fino a 3 ore affinché l'intervallo di rilevamento di anomalie effettivo venga visualizzato nel grafico. L'addestramento del nuovo modello può richiedere fino a due settimane, quindi l'intervallo di rilevamento delle anomalie mostra valori attesi più accurati.

   b. Per Ogni volta che il ***`parametro`*** è, specificare quando attivare l'allarme. Ad esempio, quando il parametro è maggiore, minore o esterno alla banda (in entrambe le direzioni).

   c. Per Anomaly detection threshold (Soglia di rilevamento anomalie), scegli il numero da utilizzare per la soglia di rilevamento anomalie. Un numero più alto crea una banda più spessa di valori "normali" che è più tollerante alle modifiche dei parametri. Un numero più basso crea una banda più sottile, che entra nello stato `ALARM` a fronte di deviazioni dei parametri più piccole. Il numero non deve essere un numero intero.

d. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

   Per creare un allarme M di N, specifica un numero del primo valore inferiore a quello del secondo valore. Per ulteriori informazioni, consulta la pagina Valutazione di un allarme.

e. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .

f. Se l'allarme utilizza un percentile come statistica monitorata, viene visualizzata una casella Percentiles with low samples (Percentili con campioni ridotti). Utilizzala per scegliere se valutare o ignorare casi con bassa frequenza di campionamento. Se scegli Ignore (maintain alarm state) (Ignora [mantieni stato dell'allarme]), lo stato corrente dell'allarme viene sempre mantenuto quando la dimensione del campione è troppo piccola. Per ulteriori informazioni, consulta la pagina CloudWatch Allarmi basati su percentili ed esempi di dati limitati.

9. Seleziona Next (Successivo).

10. In Notification (Notifica), seleziona un argomento SNS per segnalare quando l'allarme è nello stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

    Scegli Remove se non desideri che l'allarme invii notifiche.

11. È possibile configurare l'allarme per eseguire azioni EC2 o richiamare una funzione Lambda quando cambia stato, oppure per creare un Systems Manager OpsItem o un incidente quando passa allo stato ALARM. A tale scopo, seleziona il pulsante appropriato, quindi scegli lo stato di allarme e l'operazione da eseguire.

    Se scegli una funzione Lambda come operazione di allarme, specifichi il nome della funzione o l'ARN e, facoltativamente, puoi scegliere una versione specifica della funzione.

    Per ulteriori informazioni sulle azioni di Systems Manager, vedere Configurazione per la creazione CloudWatch OpsItems da allarmi e Creazione di incidenti.

> **Note**
>
> Per creare un allarme che esegua un'operazione AWS Systems Manager Incident Manager, devi disporre di determinate autorizzazioni. Per ulteriori informazioni, vedere Esempi di policy basate sull'identità per AWS Systems Manager Incident Manager.

12. Seleziona Successivo.
13. In Add a description (Aggiungere una descrizione), immetti un nome e una descrizione per l'allarme e scegli Next (Successivo). Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.

> **Tip**
>
> Il nome dell'allarme deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII

14. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni dell'allarme sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Modifica di un modello di rilevamento delle anomalie

Dopo aver creato un allarme, è possibile modificare il modello di rilevamento delle anomalie. Puoi escludere l'utilizzo di determinati periodi di tempo nella creazione del modello. È fondamentale escludere dai dati di formazione eventi insoliti quali interruzioni del sistema, distribuzioni e festività. E' inoltre possibile specificare se regolare il modello per le modifiche all'ora legale.

Per adattare il modello di rilevamento delle anomalie per un allarme

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All alarms (Tutti gli allarmi).
3. Scegli il nome dell'allarme. Se necessario, utilizza la casella di ricerca per trovare l'allarme.
4. Scegli Analizza, Nei parametri.

5. Nella colonna Dettagli, scegli ANOMALY_DETECTION_BAND, Modifica modello di rilevamento delle anomalie.
6. Per escludere l'utilizzo di un periodo di tempo per la produzione del modello, scegliete l'icona del calendario per Data di fine. Quindi, seleziona o inserisci i giorni e le ore da escludere dall'addestramento, quindi scegli Apply (Applica).
7. Se il parametro fa distinzione tra ora solare e ora legale, seleziona il fuso orario appropriato nella casella Metric timezone (Fuso orario parametro).
8. Scegli Aggiorna.

Eliminazione di un modello di rilevamento delle anomalie

L'uso del rilevamento delle anomalie per un allarme incrementa i costi addebitati da . Come best practice, se l'allarme non ha più bisogno di un modello di rilevamento delle anomalie, elimina prima l'allarme e poi il modello. Quando vengono valutati gli allarmi di rilevamento delle anomalie, vengono creati eventuali rilevatori di anomalie mancanti per tuo conto. Se si cancella il modello senza eliminare l'allarme, l'allarme ricrea automaticamente il modello.

Per eliminare un allarme

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All Alarms (Tutti gli allarmi).
3. Scegli il nome dell'allarme.
4. Scegli Operazioni > Elimina.
5. Nella casella di conferma, scegli Elimina.

Per eliminare un modello di rilevamento delle anomalie utilizzato per un allarme

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Parametri quindi scegli Tutti i parametri.
3. Scegli Browse (Sfoglia), quindi seleziona il parametro che include il modello di rilevamento delle anomalie. Puoi effettuare la ricerca del parametro tramite la casella di ricerca o selezionarne uno scegliendo tra le opzioni.
   - (Facoltativo) Se utilizzi l'interfaccia originale, scegli All metrics (Tutti i parametri), quindi seleziona il parametro che include il modello di rilevamento delle anomalie. Puoi effettuare

la ricerca del parametro tramite la casella di ricerca o selezionarne uno scegliendo tra le opzioni.

4. Seleziona Graphed metrics (Parametri nel grafico).
5. Nella scheda Graphed metrics (Parametri definiti), scegli il nome del modello di rilevamento delle anomalie che desideri rimuovere, quindi seleziona Delete anomaly detection model (Elimina il modello di rilevamento delle anomalie).
   - (Facoltativo) Se utilizzi l'interfaccia originale, scegli Edit model (Modifica modello). Viene visualizzata una nuova schermata. In questa pagina, scegli Delete model (Elimina modello), quindi Delete (Elimina).

Creazione di allarmi sui log

I passaggi descritti nelle sezioni seguenti spiegano come creare CloudWatch allarmi nei registri.

Crea un CloudWatch allarme basato su un filtro metrico del gruppo di log

La procedura in questa sezione illustra come creare un allarme basato su un filtro parametri del gruppo di log. Con i filtri metrici, puoi cercare termini e modelli nei dati di registro man mano che i dati vengono inviati. CloudWatch Per ulteriori informazioni, consulta Creare metriche dagli eventi di log utilizzando i filtri nella Amazon CloudWatch Logs User Guide. Prima di creare un allarme basato su un filtro parametri del gruppo di log, devi completare le azioni seguenti:

- Creazione di un gruppo di log. Per ulteriori informazioni, consulta Working with log groups and log stream nella Amazon CloudWatch Logs User Guide.
- Creazione di un filtro parametri. Per ulteriori informazioni, consulta Creare un filtro metrico per un gruppo di log nella Amazon CloudWatch Logs User Guide.

Creazione di un allarme basato su un filtro parametri del gruppo di log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Logs (Log), quindi Log groups (Gruppi di log).
3. Scegli il gruppo di log che include il filtro parametri.
4. Scegli Metric filters (Filtri parametri).
5. Nella scheda Metric filters (Filtri parametri), seleziona la casella relativa al filtro parametri su cui basare l'allarme.

6. Scegli Crea allarme.
7. (Facoltativo) In Metric (Parametro), modifica i campi Metric name (Nome parametro), Statistic (Statistica) e Period (Periodo).
8. In Conditions (Condizioni), specifica quanto segue:

   a. Per Threshold type (Tipo di soglia), scegli Static (Statico) o Anomaly detection (Rilevamento anomalie).

   b. Per Whenever ***`your-metric-name`***is.. , scegli Maggiore, Maggiore/Uguale, Minore/Uguale o Inferiore.

   c. Per than . . . (di), specifica un numero per il valore di soglia.
9. Scegli Additional configuration (Configurazione aggiuntiva).

   a. Per Datapoints to alarm (Punti dati per allarme), specifica quanti punti dati attivano l'allarme in modo da farlo entrare in uno stato `ALARM`. Se indichi dei valori corrispondenti, l'allarme passa nello stato `ALARM` nel caso in cui si verifichi una violazione durante tali periodi consecutivi. Per creare un allarme M di N, specifica un numero del primo valore inferiore a quello del secondo valore. Per ulteriori informazioni, consulta Usare gli CloudWatch allarmi Amazon.

   b. Per Missing data treatment (Trattamento dei dati mancanti), seleziona un'opzione dal menu a discesa per specificare come trattare i dati mancanti quando viene valutato l'allarme.
10. Seleziona Successivo.
11. In Notification (Notifica), seleziona un argomento Amazon SNS per segnalare quando l'allarme si trova nello stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    a. (Facoltativo) Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

    b. (Facoltativo) Per fare in modo che l'allarme non invii notifiche, scegli Remove (Rimuovi).
12. Per fare in modo che l'allarme esegua operazioni Auto Scaling, EC2, Lambda o Systems Manager scegli il pulsante appropriato e scegli lo stato di allarme e l'operazione da eseguire. Se scegli una funzione Lambda come operazione di allarme, specifichi il nome della funzione o l'ARN e, facoltativamente, puoi scegliere una versione specifica della funzione.

    Gli allarmi possono eseguire le operazioni Systems Manager solo quando entrano nello stato ALARM. Per ulteriori informazioni sulle azioni di Systems Manager, vedere Configurazione per la creazione CloudWatch OpsItems da allarmi e Creazione di incidenti.

> **Note**
>
> Per creare un allarme che esegua un'operazione SSM Incident Manager, è necessario disporre di determinate autorizzazioni. Per ulteriori informazioni, vedere Esempi di policy basate sull'identità per AWS Systems Manager Incident Manager.

13. Seleziona Successivo.
14. Per Name and description (Nome e descrizione), inserisci un nome e una descrizione per il tuo allarme. Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.
15. Per Preview and create (Anteprima e creazione), verifica che la configurazione sia corretta, quindi seleziona Create alarm (Crea allarme).

Combinazione di allarmi

Con CloudWatch, puoi combinare più allarmi in un unico allarme composito per creare un indicatore di stato riepilogato e aggregato su un'intera applicazione o gruppo di risorse. Gli allarmi compositi sono allarmi che determinano il loro stato monitorando gli stati di altri allarmi. Definisci le regole per combinare lo stato degli allarmi monitorati utilizzando la logica booleana.

È possibile utilizzare allarmi compositi per ridurre il rumore dell'allarme intraprendendo operazioni solo a livello aggregato. Ad esempio, puoi creare un allarme composito affinché il team del tuo server web riceva una notifica se si attiva un qualsiasi allarme relativo al server web. Quando uno di questi allarmi entra nello stato ALARM, l'allarme composito entra automaticamente nello stato ALARM e invia una notifica al team. Se anche altri allarmi relativi al server web entrano nello stato ALARM, il team non viene sovraccaricato di nuove notifiche poiché l'allarme composito li ha già informati della situazione.

Puoi anche utilizzare gli allarmi compositi per creare condizioni di allarme complesse e agire solo quando vengono soddisfatte molte condizioni diverse. Ad esempio, è possibile creare un allarme composito che combini un allarme CPU e un allarme di memoria e invii una notifica al team solo se si sono attivati sia gli allarmi relativi alla CPU sia quelli relativi alla memoria.

Utilizzo degli allarmi compositi

Quando utilizzi gli allarmi compositi, hai due possibilità:

- Configurare le operazioni che desideri intraprendere solo a livello di allarme composito e creare gli allarmi monitorati sottostanti senza operazioni
- Configurare un diverso insieme di operazioni a livello di allarme composito. Ad esempio, le operazioni di allarme composito potrebbero coinvolgere un team diverso in caso di un problema diffuso.

Gli allarmi compositi possono effettuare soltanto le operazioni seguenti:

- Notifica di argomenti Amazon SNS
- Richiamo delle funzioni Lambda
- Crea OpsItems in Systems Manager Ops Center
- Creazione di incidenti in Systems Manager Incident Manager

> **Note**
>
> Tutti gli allarmi sottostanti dell'allarme composito devono trovarsi nello stesso account e nella stessa regione dell'allarme composito. Tuttavia, se imposti un allarme composito in un account di monitoraggio CloudWatch dell'osservabilità tra più account, gli allarmi sottostanti possono controllare le metriche in diversi account di origine e nell'account di monitoraggio stesso. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.
> Un singolo allarme composito può monitorare 100 allarmi sottostanti e 150 allarmi compositi possono monitorare un singolo allarme sottostante.

Espressioni di regola

Tutti gli allarmi composti contengono espressioni di regole. Le espressioni delle regole indicano agli allarmi compositi quali altri allarmi monitorare e determinare i loro stati. Espressioni di regola può riferirsi ad allarmi dei parametri e ad altri allarmi compositi. Quando si fa riferimento a un allarme in un'espressione di regola, si designa una funzione all'allarme che determina in quale dei tre stati seguenti si troverà l'allarme:

- ALLARME

  ALARM ("alarm-name or alarm-ARN") è TRUE se l'allarme è in stato ALARM.

- OK

  OK ("alarm-name or alarm-ARN") è TRUE se l'allarme è in stato OK.

- INSUFFICIENT_DATA

  INSUFFICIENT_DATA ("alarm-name or alarm-ARN") è TRUE se l'allarme denominato è in stato INSUFFICIENT_DATA.

> **Note**
>
> TRUE restituisce sempre TRUE e FALSE restituisce sempre FALSE

Espressioni di esempio

Parametro di richiesta `AlarmRule` supporta l'uso degli operatori logici `AND`, `OR` e `NOT`, in modo da poter combinare più funzioni in un'unica espressione. Le seguenti espressioni di esempio mostrano come configurare gli allarmi sottostanti nell'allarme composito:

- `ALARM(CPUUtilizationTooHigh) AND ALARM(DiskReadOpsTooHigh)`

  L'espressione specifica che l'allarme composito entra in stato `ALARM` solo se `CPUUtilizationTooHigh` e `DiskReadOpsTooHigh` sono in stato `ALARM`.

- `ALARM(CPUUtilizationTooHigh) AND NOT ALARM(DeploymentInProgress)`

  L'espressione specifica che l'allarme composito entra in stato `ALARM` se `CPUUtilizationTooHigh` è in stato `ALARM` e `DeploymentInProgress` non è in stato `ALARM`. Questo è un esempio di allarme composito che riduce il rumore di allarme durante una finestra di implementazione.

- `(ALARM(CPUUtilizationTooHigh) OR ALARM(DiskReadOpsTooHigh)) AND OK(NetworkOutTooHigh)`

  L'espressione specifica che l'allarme composito entra in stato `ALARM` se `(ALARM(CPUUtilizationTooHigh)` o `(DiskReadOpsTooHigh)` è in stato `ALARM` e `(NetworkOutTooHigh)` è in stato `OK`. Questo è un esempio di allarme composito che riduce il rumore dell'allarme non inviandoti notifiche quando uno degli allarmi sottostanti non è in stato `ALARM` mentre si verifica un problema di rete.

Argomenti

Creazione di un allarme composito

I passaggi di questa sezione spiegano come utilizzare la CloudWatch console per creare un allarme composito. Puoi anche utilizzare l'API o AWS CLI creare un allarme composito. Per ulteriori informazioni, consulta PutCompositeAlarmo put-composite-alarm

Come creare un allarme composito

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), quindi Create alarm (Tutti gli allarmi).
3. Dall'elenco degli allarmi, seleziona la casella di controllo accanto a ciascuno degli allarmi esistenti a cui si desidera fare riferimento nell'espressione della regola, quindi scegli Crea allarme composito.
4. In Specifica le condizioni di allarme composito, specifica l'espressione della regola per il nuovo allarme composito.

   > **Note**
   >
   > Automaticamente, gli allarmi selezionati dall'elenco degli allarmi sono elencati nella casella Conditions (Condizioni). Per impostazione predefinita, la funzione `ALARM` è stata designata per ciascuno dei tuoi allarmi e ciascuno degli allarmi è affiancato dall'operatore logico `OR`.

   Puoi utilizzare le seguenti operazioni secondarie per modificare l'espressione della regola:

   a. Puoi modificare lo stato richiesto per ciascuno dei tuoi allarmi da `ALARM` a `OK` o `INSUFFICENT_DATA`.
   b. È possibile modificare l'operatore logico nell'espressione della regola da `OR` a `AND` o `NOT` e puoi aggiungere parentesi per raggruppare le tue funzioni.
   c. Puoi includere altri allarmi nell'espressione della tua regola o eliminare allarmi dall'espressione della regola.

Esempio: espressione della regola con condizioni

```
(ALARM("CPUUtilizationTooHigh") OR
ALARM("DiskReadOpsTooHigh")) AND
OK("NetworkOutTooHigh")
```

Nell'espressione della regola di esempio in cui l'allarme composito entra in funzione `ALARM` quando ALARM (DiskReadOpsTooHigh«CPUUtilizationTooHigh" o ALARM (» «) è attivo contemporaneamente `ALARM` a OK (» NetworkOutTooHigh «)`OK`.

5. Al termine, scegli Apply (Applica).
6. In Configurazione delle operazioni, puoi scegliere tra le seguenti opzioni:

   Per Notifica

   - Seleziona un argomento SNS esistente, Creazione di un nuovo argomento di SNS, oppure Utilizzare un argomento di ARN per definire l'argomento SNS che riceverà la notifica.
   - Aggiungi notifica per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi.
   - Rimuovi per impedire all'allarme di inviare notifiche o intraprendere operazioni.

   (Facoltativo) Per fare in modo che l'allarme richiami una funzione Lambda quando cambia stato, scegli Aggiungi operazione Lambda. Quindi specifica il nome della funzione o l'ARN e, facoltativamente, scegli una versione specifica della funzione.

   Per Operazione Systems Manager

   - Aggiungi operazione Systems Manager, in modo che l'allarme possa eseguire un'azione SSM quando entra in stato ALARM.

   Per ulteriori informazioni sulle azioni di Systems Manager, vedere Configurazione CloudWatch per creare a OpsItems partire dagli allarmi nella Guida per l'AWS Systems Manager utente e Creazione degli incidenti nella Guida per l'utente di Incident Manager. Per creare un allarme che esegua un'operazione SSM Incident Manager, è necessario disporre delle autorizzazioni corrette. Per ulteriori informazioni, vedere Esempi di policy basate sull'identità per AWS Systems Manager Incident Manager nella Guida per l'utente di Incident Manager.

7. Al termine, scegli Apply (Applica).
8. In Aggiungi nome e descrizione, inserisci un nome dell'allarme e opzionale descrizione del nuovo allarme composito. Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.
9. Al termine, scegli Apply (Applica).
10. In Visualizza l'anteprima e crea, conferma le informazioni e scegli Creazione di un allarme composito.

> **Note**
>
> È possibile creare un ciclo di allarmi compositi, in cui un allarme composito e un altro allarme composito dipendono l'uno dall'altro. Se ti trovi in questo scenario, i tuoi allarmi compositi smettono di essere valutati e non puoi eliminare i tuoi allarmi compositi perché dipendono l'uno dall'altro. Il modo più semplice per interrompere il ciclo di dipendenza tra i tuoi allarmi compositi è cambiare la funzione `AlarmRule` in uno dei tuoi allarmi compositi in `False`.

Soppressione delle operazioni degli allarmi compositi

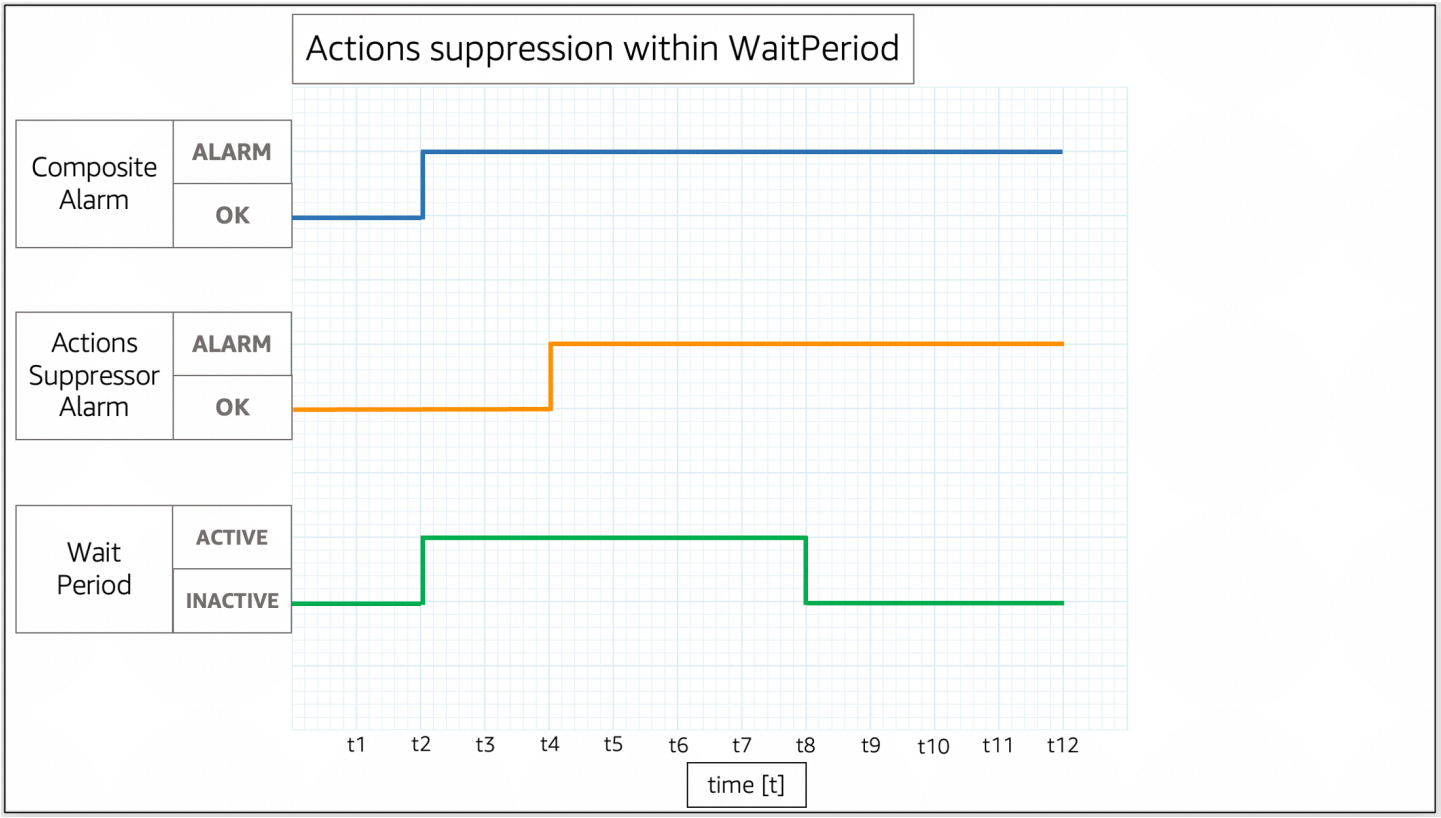
Poiché gli allarmi compositi consentono di ottenere una visione aggregata dello stato di integrità attraverso più allarmi, esistono alcune situazioni comuni in cui è previsto che tali allarmi si attivino, ad esempio, durante una finestra di manutenzione dell'applicazione o quando si indaga su un incidente in corso. In tali situazioni, potresti voler sopprimere le operazioni degli allarmi compositi, per evitare notifiche indesiderate o la creazione di nuove segnalazioni di incidenti.

Con la soppressione dell'operazione di allarme composita, si definiscono allarmi come allarmi soppressori. Gli allarmi soppressori impediscono agli allarmi compositi di agire. Ad esempio, è possibile specificare un allarme soppressore che rappresenta lo stato di una risorsa di supporto. Se la risorsa di supporto è inattiva, l'allarme soppressore impedisce all'allarme composito di inviare notifiche. La soppressione delle operazioni di allarmi compositi aiuta a ridurre il rumore degli allarmi, in modo da dedicare meno tempo alla gestione degli allarmi e più tempo a concentrarsi sulle operazioni.

Gli allarmi soppressori vengono specificati quando si configurano gli allarmi compositi. Qualsiasi allarme può funzionare come un allarme soppressore. Quando un allarme soppressore cambia stato da OK a ALARM, il suo allarme composito smette di agire. Quando un allarme soppressore cambia stato da ALARM a OK, il suo allarme composito riprende ad agire.

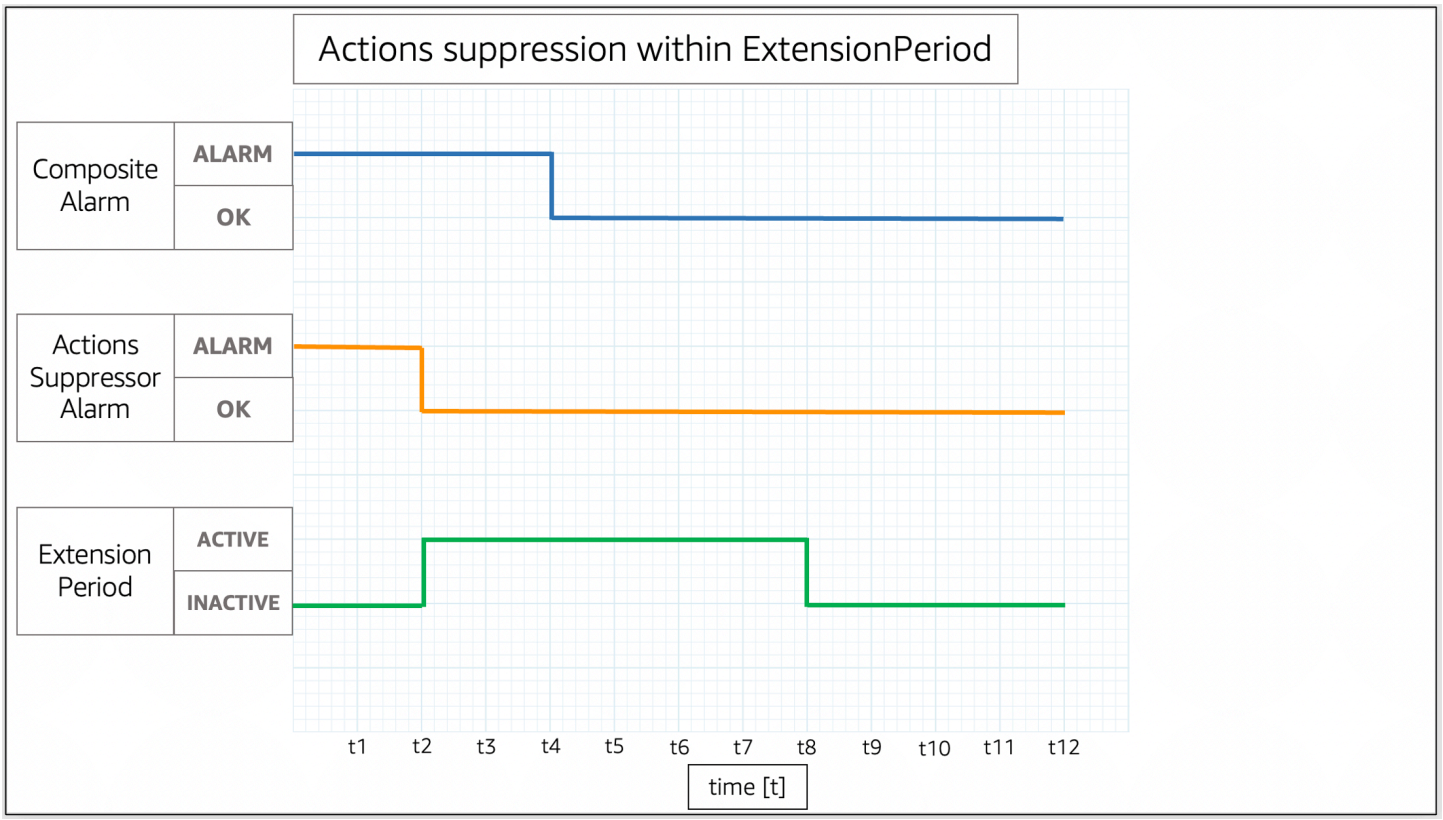
`WaitPeriod` e `ExtensionPeriod`

Quando si specifica un allarme soppressore, si impostano i parametri `WaitPeriod` e `ExtensionPeriod`. Questi parametri impediscono agli allarmi compositi di agire in modo imprevisto mentre gli allarmi soppressori cambiano stato. Utilizza `WaitPeriod` per compensare eventuali ritardi che possono verificarsi quando un allarme soppressore cambia da OK a ALARM. Ad esempio, se un allarme soppressore cambia da OK a ALARM entro 60 secondi, imposta `WaitPeriod` a 60 secondi.



Nell'immagine, l'allarme composito cambia da OK a ALARM a t2. Un `WaitPeriod` inizia a t2 e termina a t8. Questo dà all'allarme soppressore il tempo di cambiare stato da OK a ALARM a t4 prima di sopprimere le operazioni dell'allarme composito quando il `WaitPeriod` scade al t8.

Utilizza `ExtensionPeriod` per compensare eventuali ritardi che possono verificarsi quando un allarme composito cambia OK a seguito di un allarme soppressore che cambia in OK. Ad esempio, se un allarme composito cambia in OK entro 60 secondi dal passaggio di un allarme soppressore a OK, imposta `ExtensionPeriod` a 60 secondi.



Nell'immagine, l'allarme soppressore cambia da `ALARM` a `OK` a t2. Un `ExtensionPeriod` inizia a t2 e termina a t8. Questo dà all'allarme composito il tempo di passare da `ALARM` a `OK` prima della scadenza di `ExtensionPeriod` a t8.

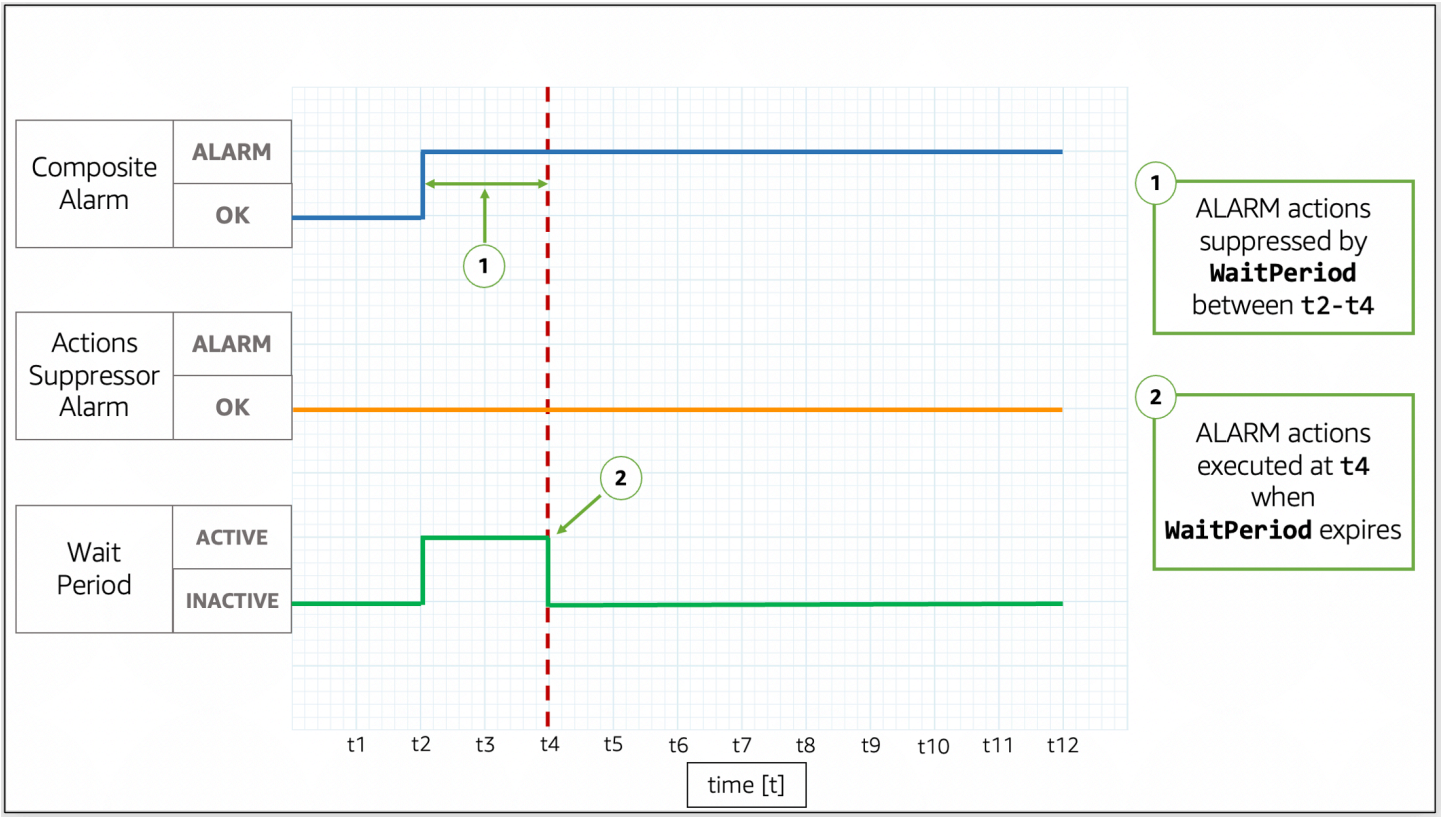
Gli allarmi compositi non intervengono quando `WaitPeriod` e `ExtensionPeriod` diventano attivi. Gli allarmi compositi eseguono operazioni basate sui loro stati correnti quando `ExtensionPeriod` e `WaitPeriod` diventano inattivi. Ti consigliamo di impostare il valore per ogni parametro su 60 secondi, poiché CloudWatch valuta gli allarmi metrici ogni minuto. È possibile impostare i parametri su qualsiasi numero intero in secondi.

Gli esempi seguenti descrivono in modo più dettagliato come `WaitPeriod` e `ExtensionPeriod` impediscono che gli allarmi compositi intraprendano operazioni impreviste.

> **Note**
>
> Negli esempi seguenti, `WaitPeriod` è configurato come 2 unità di tempo e `ExtensionPeriod` è configurato come 3 unità di tempo.

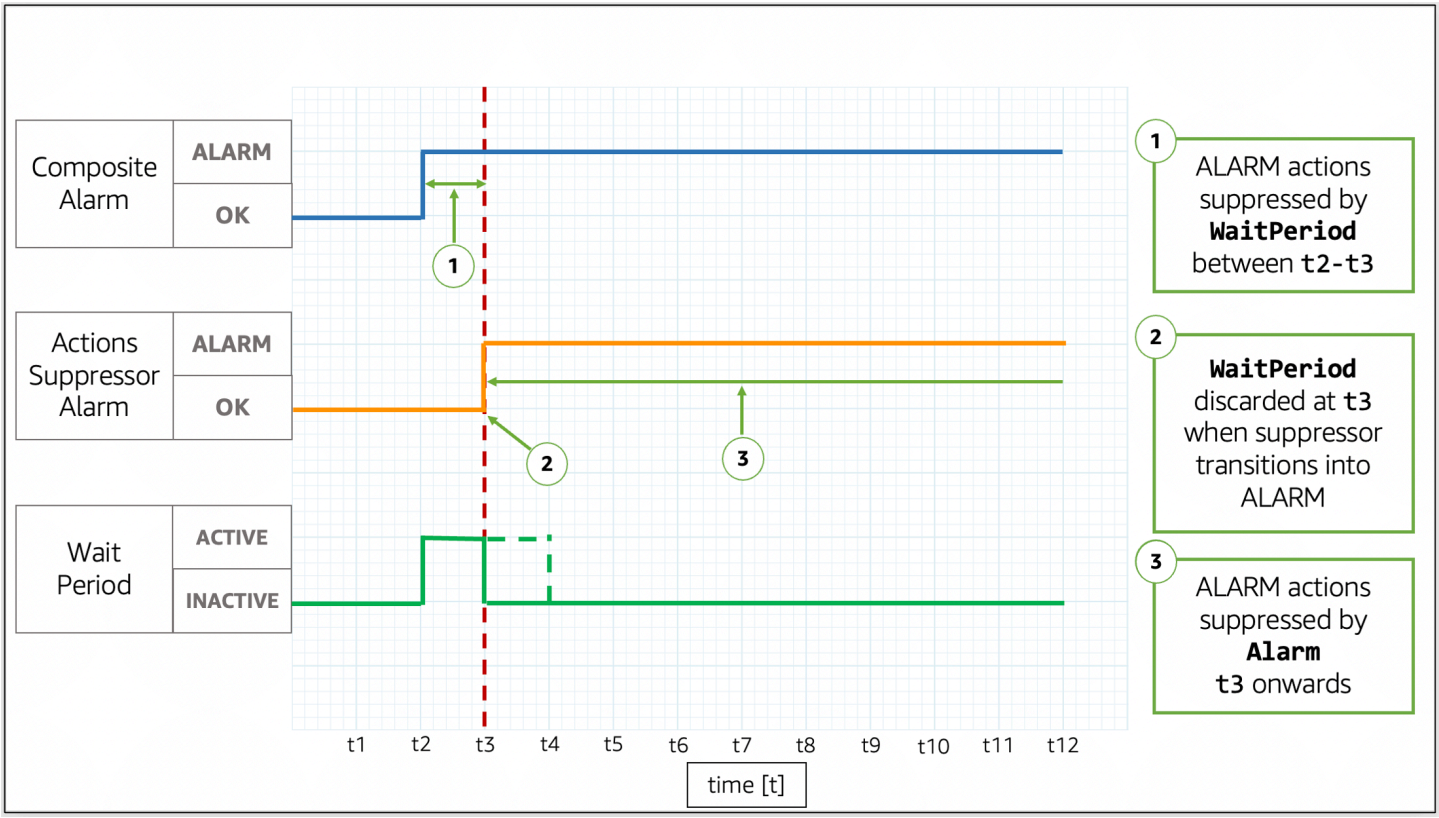
Esempi

Esempio 1: le operazioni non vengono soppresse dopo **`WaitPeriod`**



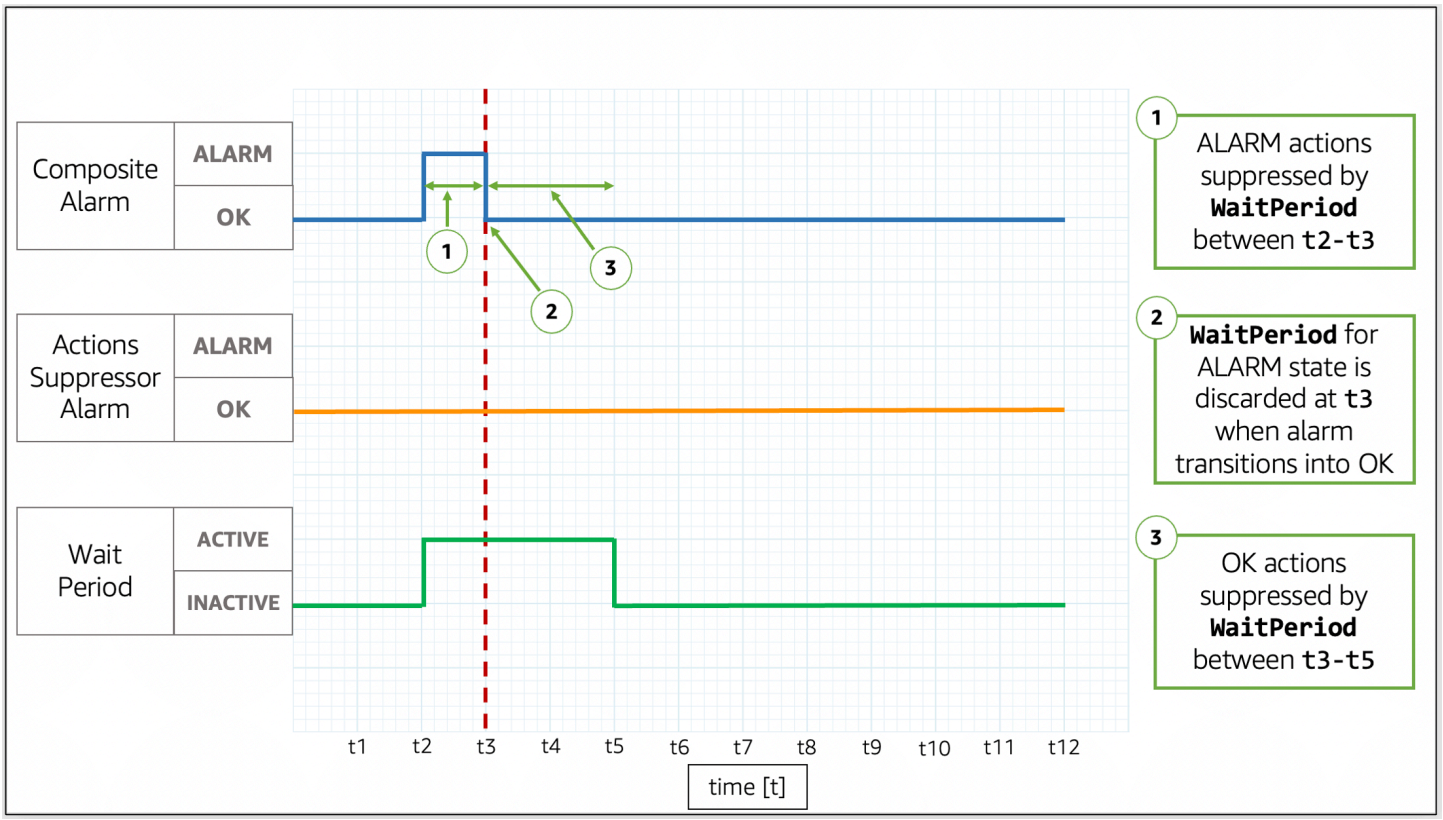
Nell'immagine, l'allarme composito cambia stato da OK a ALARM a t2. Un `WaitPeriod` inizia a t2 e termina a t4, in modo che possa impedire all'allarme composito di agire. Dopo la scadenza del `WaitPeriod`a t4, l'allarme composito compie le sue operazioni perché l'allarme soppressore è ancora attivo OK.

Esempio 2: le operazioni vengono soppresse dall'allarme prima della scadenza di **`WaitPeriod`**



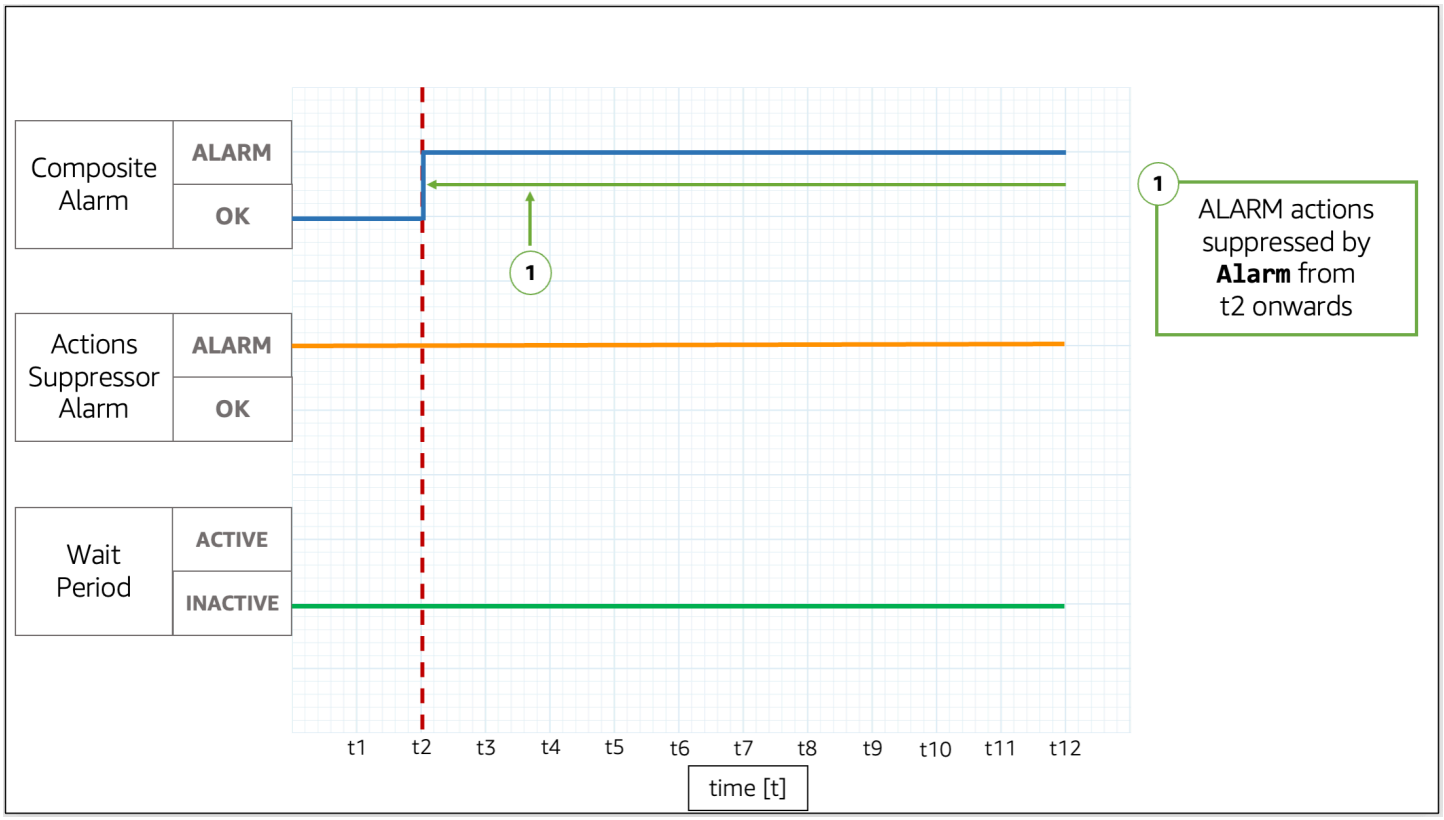
Nell'immagine, l'allarme composito cambia stato da OK a ALARM a t2. Un `WaitPeriod` inizia a t2 e termina a t4. Questo dà all'allarme soppressore il tempo di cambiare stato da OK a ALARM a t3. Perché l'allarme soppressore cambia stato OK a ALARM a t3, il `WaitPeriod` che è iniziato a t2 viene scartato e l'allarme soppressore ora impedisce all'allarme composito di agire.

Esempio 3: transizione di stato quando le operazioni vengono soppresse da **`WaitPeriod`**



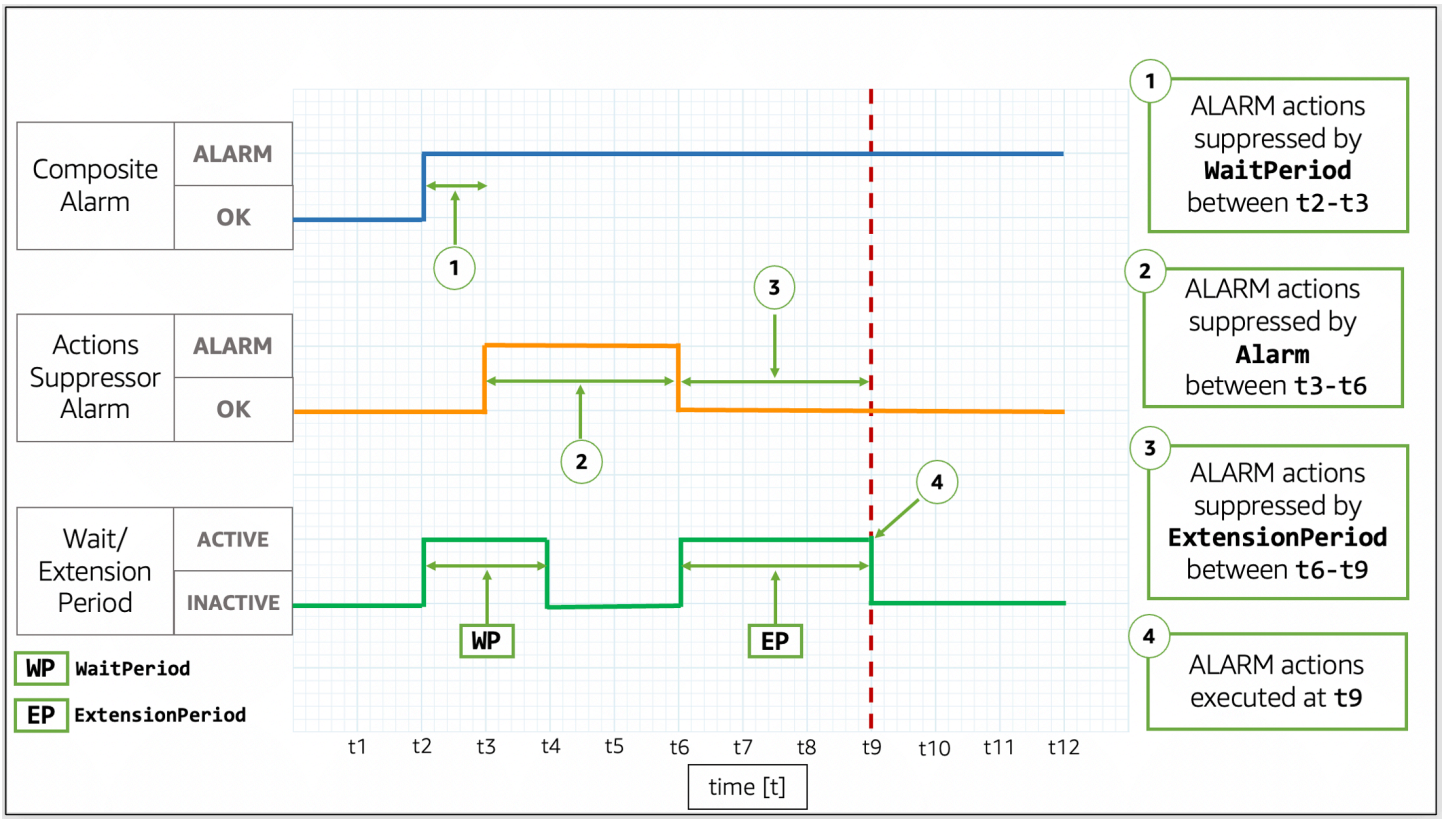
Nell'immagine, l'allarme composito cambia stato da `OK` a `ALARM` a t2. Un `WaitPeriod` inizia a t2 e termina a t4. Questo dà all'allarme soppressore il tempo di cambiare stato. L'allarme composito torna a `OK` a t3, quindi il `WaitPeriod` che è iniziato a t2 viene scartato. Un nuovo `WaitPeriod` inizia a t3 e termina a t5. Dopo la scadenza del nuovo `WaitPeriod` a t5, l'allarme composito compie le sue operazioni.

Esempio 4: transizione di stato quando le operazioni vengono soppresse da un allarme



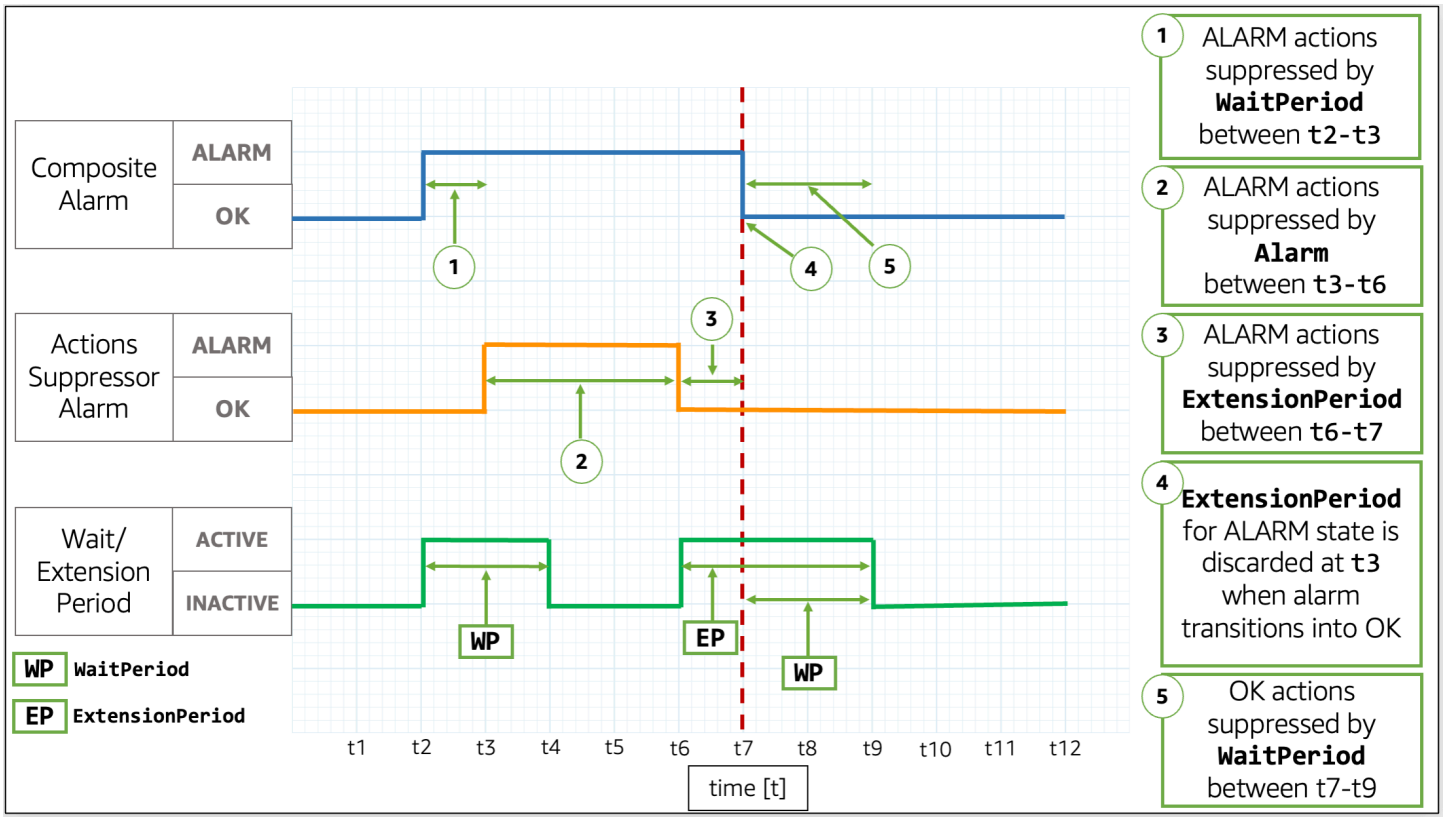
Nell'immagine, l'allarme composito cambia stato da `OK` a `ALARM` a t2. L'allarme soppressore è già in stato `ALARM`. L'allarme soppressore impedisce all'allarme composito di agire.

Esempio 5: le operazioni non vengono soppresse dopo **`ExtensionPeriod`**



Nell'immagine, l'allarme composito cambia stato da OK a ALARM a t2. Un `WaitPeriod` inizia a t2 e termina a t4. Questo dà all'allarme soppressore il tempo di cambiare stato da OK a ALARM a t3 prima che sopprime le operazioni dell'allarme composito fino a t6. Perché l'allarme soppressore cambia stato OK a ALARM a t3, il `WaitPeriod` che è iniziato a t2 viene scartato. A t6, l'allarme soppressore cambia in OK. Un `ExtensionPeriod` inizia a t6 e termina a t9. Dopo la scadenza di `ExtensionPeriod`, l'allarme composito esegue le sue operazioni.

Esempio 6: transizione di stato quando le operazioni vengono soppresse da **`ExtensionPeriod`**



Nell'immagine, l'allarme composito cambia stato da OK a ALARM a t2. Un `WaitPeriod` inizia a t2 e termina a t4. Questo dà all'allarme soppressore il tempo di cambiare stato da OK a ALARM a t3 prima che sopprime le operazioni dell'allarme composito fino a t6. Perché l'allarme soppressore cambia stato OK a ALARM a t3, il `WaitPeriod` che è iniziato a t2 viene scartato. A t6, l'allarme soppressore torna a OK. Un `ExtensionPeriod` inizia a t6 e termina a t9. Quando l'allarme composito torna a OK a t7, il `ExtensionPeriod` viene scartato e un nuovo `WaitPeriod` inizia a t7 e termina a t9.

> **Tip**
>
> Se si sostituisce l'operazione di allarme soppressore, qualsiasi `WaitPeriod` o `ExtensionPeriod` viene scartato.

Operazioni sulle modifiche degli allarmi

CloudWatch può notificare agli utenti due tipi di modifiche agli allarmi: quando un allarme cambia stato e quando la configurazione di un allarme viene aggiornata.

Quando un allarme viene valutato, potrebbe cambiare da uno stato all'altro, ad esempio ALARM, OK o INSUFFICIENT_DATA. Queste modifiche dello stato di allarme possono segnalare un possibile incidente, un ritorno alla normalità o l'indisponibilità di un parametro. In questi casi, potresti voler coinvolgere o notificare gli utenti utilizzando una delle seguenti opzioni:

- È possibile configurare l'allarme per inviare una notifica a un argomento SNS come parte delle azioni dell'avviso. Un argomento SNS può quindi essere configurato per la messaggistica application-to-application (A2A) e per le notifiche application-to-person (A2P), inclusi canali come notifiche e-mail e SMS. Tutte le destinazioni definite per l'argomento SNS ricevono la notifica di allarme. Per ulteriori informazioni, consulta le Destinazioni di eventi Amazon SNS.
- È possibile configurare le notifiche per gli eventi di modifica dello stato degli allarmi. AWS User Notifications offre un modo nativo per configurare tali notifiche ed è l'approccio consigliato.

Inoltre, CloudWatch invia eventi ad Amazon EventBridge ogni volta che gli allarmi cambiano di stato e quando gli allarmi vengono creati, eliminati o aggiornati. Puoi scrivere EventBridge regole per intraprendere azioni o ricevere notifiche quando ricevi questi EventBridge eventi.

Argomenti

Notifica agli utenti delle modifiche agli allarmi

Questa sezione spiega come utilizzare AWS User Notifications o Amazon Simple Notification Service per far sì che gli utenti vengano informati delle modifiche agli allarmi.

Configurazione delle notifiche AWS utente

È possibile utilizzare le notifiche AWS utente per configurare i canali di consegna per ricevere notifiche sulla modifica dello stato CloudWatch dell'allarme e sugli eventi di modifica della configurazione. L'utente riceverà una notifica quando un evento corrisponde a una regola specificata. È possibile ricevere notifiche per gli eventi tramite più canali, tra cui e-mail, notifiche chat di AWS Chatbot o notifiche push di AWS Console Mobile Application. È possibile visualizzare le notifiche anche nel Centro notifiche della console. La funzionalità Notifiche all'utente supporta l'aggregazione, che può ridurre il numero di notifiche ricevute durante eventi specifici.

Le configurazioni di notifica create con AWS User Notifications non vengono conteggiate ai fini del limite del numero di azioni che è possibile configurare per lo stato di allarme desiderato. Poiché AWS User Notificatinos corrisponde agli eventi trasmessi ad Amazon EventBridge, invia notifiche per tutti gli allarmi nel tuo account e nelle regioni selezionate, a meno che tu non specifichi un filtro avanzato per consentire o negare allarmi o schemi specifici.

L'esempio seguente di filtro avanzato corrisponde a una modifica dello stato dell'allarme da OK a ALARM sull'allarme denominato `ServerCpuTooHigh`.

```
{
"detail": {
    "alarmName": ["ServerCpuTooHigh"],
    "previousState": { "value": ["OK"] },
    "state": { "value": ["ALARM"] }
  }
}
```

Puoi utilizzare una qualsiasi delle proprietà pubblicate da un allarme negli eventi per creare un filtro. EventBridge Per ulteriori informazioni, consulta la pagina Eventi di allarme e EventBridge.

Impostazione delle notifiche Amazon SNS

Puoi utilizzare Amazon Simple Notification Service per inviare sia messaggi application-to-application (A2A) che messaggi application-to-person (A2P), inclusi messaggi di testo mobili (SMS) ed e-mail. Per ulteriori informazioni, consulta le Destinazioni di eventi Amazon SNS.

Per ogni stato in cui può verificarsi un allarme, puoi configurarlo per inviare un messaggio a un argomento SNS. Ogni argomento di Amazon SNS che configuri per uno stato di un determinato allarme conta ai fini del limite del numero di azioni che puoi configurare per tale allarme e stato. Puoi inviare messaggi allo stesso argomento Amazon SNS da qualsiasi allarme presente nel tuo account e utilizzare lo stesso argomento Amazon SNS per i consumer di applicazioni (A2A) e persone (A2P). Poiché questa configurazione viene eseguita a livello di allarme, solo gli allarmi che hai configurato invieranno messaggi all'argomento Amazon SNS selezionato.

Come prima cosa crea e sottoscrivi un argomento. Puoi anche pubblicare un messaggio di prova per l'argomento. Per vedere un esempio, consulta Configurazione di un argomento Amazon SNS utilizzando AWS Management Console. Per ulteriori informazioni, consulta Nozioni di base su Amazon SNS.

In alternativa, se prevedi di utilizzare il AWS Management Console per creare il tuo CloudWatch allarme, puoi saltare questa procedura perché puoi creare l'argomento quando crei l'allarme.

Quando crei un CloudWatch allarme, puoi aggiungere azioni per qualsiasi stato target in cui entra l'allarme. Aggiungi una notifica Amazon SNS per lo stato di cui desideri ricevere una notifica e seleziona l'argomento Amazon SNS che hai creato nel passaggio precedente per inviare una notifica e-mail quando l'allarme entra nello stato selezionato.

> **Note**
>
> Quando crei un argomento di Amazon SNS, scegli di renderlo un argomento standard o un argomento FIFO. CloudWatch garantisce la pubblicazione di tutte le notifiche di allarme relative a entrambi i tipi di argomenti. Tuttavia, anche se si utilizza un argomento FIFO, in rari casi CloudWatch invia le notifiche all'argomento in modo errato. Se si utilizza un argomento FIFO, l'allarme imposta l'ID del gruppo di messaggi delle notifiche di avviso come hash dell'ARN dell'allarme.

Prevenire i problemi di “confused deputy”

Per evitare problemi di sicurezza secondaria confusi tra servizi, ti consigliamo di utilizzare le chiavi `aws:SourceArn` e le chiavi di condizione `aws:SourceAccount` globali nella policy delle risorse di Amazon SNS che concede l'autorizzazione CloudWatch ad accedere alle tue risorse Amazon SNS.

Il seguente esempio di policy in materia di risorse utilizza la chiave `aws:SourceArn` condition per restringere l'`SNS:Publish`autorizzazione a essere utilizzata solo dagli CloudWatch allarmi nell'account specificato.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "cloudwatch.amazonaws.com"
        },
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:us-east-2:444455556666:MyTopic",
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": "arn:aws:cloudwatch:us-east-2:111122223333:alarm:*"
            },
            "StringEquals": {
```

```
                "aws:SourceAccount": "111122223333"
            }
        }
    }]
}
```

Se un ARN di allarme include caratteri non ASCII, utilizza solo la chiave di condizione globale `aws:SourceAccount` per limitare le autorizzazioni.

Configurazione di un argomento Amazon SNS utilizzando AWS Management Console

Come prima cosa crea e sottoscrivi un argomento. Puoi anche pubblicare un messaggio di prova per l'argomento.

Creazione di un argomento SNS

1. Apri la console Amazon SNS all'indirizzo https://console.aws.amazon.com/sns/v3/home.
2. Sul pannello di controllo Amazon SNS sotto Common actions (Operazioni comuni), scegli Create Topic (Crea argomento).
3. Nella finestra di dialogo Create new topic (Crea nuovo argomento) per Topic name (Nome argomento), digita un nome per l'argomento (ad esempio, **`my-topic`**).
4. Scegli Create topic (Crea argomento).
5. Copia il valore di Topic ARN (ARN argomento) per l'attività successiva (ad esempio, arn:aws:sns:us-east-1:111122223333:my-topic).

Abbonamento a un argomento SNS

1. Apri la console Amazon SNS all'indirizzo https://console.aws.amazon.com/sns/v3/home.
2. Nel pannello di navigazione, scegli Subscriptions (Abbonamenti), quindi Create subscription (Crea abbonamento).
3. Nella finestra di dialogo Create subscription (Crea abbonamento), per Topic ARN (ARN argomento), copia l'ARN dell'argomento creato durante la precedente attività.
4. Per Protocol, scegli Email.
5. Per Endpoint, digita un indirizzo e-mail che puoi utilizzare per ricevere la notifica, quindi scegli Create subscription (Crea abbonamento).
6. Dalla tua applicazione di posta elettronica, apri il messaggio da AWS Notifiche e conferma l'iscrizione.

Nel Web browser viene visualizzata una risposta di conferma di Amazon SNS.

Pubblicazione di messaggio test su un argomento SNS

1. Apri la console Amazon SNS all'indirizzo https://console.aws.amazon.com/sns/v3/home.
2. Nel pannello di navigazione, scegli Topics (Argomenti).
3. Nella pagina Topics (Argomenti), seleziona un argomento e scegli Publish to topic (Pubblica nell'argomento).
4. Nella pagina Publish a message (Pubblica un messaggio), per Subject (Oggetto), digita l'oggetto del messaggio e per Message (Messaggio) digita un breve messaggio.
5. Seleziona Publish Message (Pubblica messaggio).
6. Controlla la tua e-mail per la conferma di ricezione del messaggio.

Configurazione di un argomento SNS utilizzando il AWS CLI

In primo luogo crea un argomento SNS, quindi pubblica un messaggio direttamente sull'argomento per verificare di averlo configurato correttamente.

Per configurare un argomento SNS

1. Crea l'argomento usando il comando create-topic come segue.

```
aws sns create-topic --name my-topic
```

Amazon SNS restituisce un ARN dell'argomento con il seguente formato:

```
{
    "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic"
}
```

2. Sottoscrivi l'indirizzo e-mail per l'argomento utilizzando il comando subscribe. Se la richiesta di abbonamento va a buon fine, riceverai un'e-mail di conferma.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:111122223333:my-topic --
protocol email --notification-endpoint my-email-address
```

Amazon SNS restituisce quanto segue:

```
{
    "SubscriptionArn": "pending confirmation"
}
```

3. Dalla tua applicazione di posta elettronica, apri il messaggio da AWS Notifiche e conferma l'iscrizione.

   Nel Web browser viene visualizzata una risposta di conferma da Amazon Simple Notification Service.

4. Controlla l'abbonamento usando il list-subscriptions-by-topiccomando.

   ```
   aws sns list-subscriptions-by-topic --topic-arn arn:aws:sns:us-
   east-1:111122223333:my-topic
   ```

   Amazon SNS restituisce quanto segue:

   ```
   {
     "Subscriptions": [
       {
           "Owner": "111122223333",
           "Endpoint": "me@mycompany.com",
           "Protocol": "email",
           "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic",
           "SubscriptionArn": "arn:aws:sns:us-east-1:111122223333:my-topic:64886986-
   bf10-48fb-a2f1-dab033aa67a3"
       }
     ]
   }
   ```

5. (Facoltativo) Pubblica un messaggio test per l'argomento utilizzando il comando publish.

   ```
   aws sns publish --message "Verification" --topic arn:aws:sns:us-
   east-1:111122223333:my-topic
   ```

   Amazon SNS restituisce quanto segue:

   ```
   {
       "MessageId": "42f189a0-3094-5cf6-8fd7-c2dde61a4d7d"
   }
   ```

6. Controlla la tua e-mail per la conferma di ricezione del messaggio.

Eventi di allarme e EventBridge

CloudWatch invia eventi ad Amazon EventBridge ogni volta che un CloudWatch allarme viene creato, aggiornato, eliminato o cambia lo stato dell'allarme. Puoi utilizzare EventBridge questi eventi per scrivere regole che intraprendano azioni, come avvisarti quando un allarme cambia stato. Per ulteriori informazioni, consulta What is Amazon EventBridge?

CloudWatch garantisce la trasmissione degli eventi di modifica dello stato di allarme a EventBridge.

Eventi di esempio da CloudWatch

Questa sezione include eventi di esempio tratti da CloudWatch.

Modifica dello stato per un allarme di parametro singolo


```
{
    "version": "0",
    "id": "c4c1c1c9-6542-e61b-6ef0-8c4d36933a92",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:04:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "configuration": {
            "description": "Goes into alarm when server CPU utilization is too high!",
            "metrics": [
                {
                    "id": "30b6c6b2-a864-43a2-4877-c09a1afc3b87",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-1234567890123456 7"
                            },
                            "name": "CPUUtilization",
                            "namespace": "AWS/EC2"
```

```
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ]
        },
        "previousState": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [0.0666851903306472 (01/10/19 13:46:00)] was not greater than the threshold (50.0)
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-01T13:56:40.985+0000\",\"startDate\":\"2019-10-01T13:46:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[0.0666851903306472],
\"threshold\":50.0}",
            "timestamp": "2019-10-01T13:56:40.987+0000",
            "value": "OK"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [99.50160229693434 (02/10/19 16:59:00)] was greater than the threshold (50.0) (minimum
 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:04:40.985+0000\",\"startDate\":\"2019-10-02T16:59:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[99.50160229693434],
\"threshold\":50.0}",
            "timestamp": "2019-10-02T17:04:40.989+0000",
            "value": "ALARM"
        }
    }
}
```

Modifica dello stato per un allarme matematico di parametro

```
{
    "version": "0",
    "id": "2dde0eb1-528b-d2d5-9ca6-6d590caf2329",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:20:48Z",
    "region": "us-east-1",
```

```
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "configuration": {
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "metrics": [
                {
                    "expression": "SUM(METRICS())",
                    "id": "e1",
                    "label": "Total Network Traffic",
                    "returnData": true
                },
                {
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkIn",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkOut",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                }
```

```
            ]
        },
        "previousState": {
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2019-10-02T17:20:03.642+0000",
            "value": "INSUFFICIENT_DATA"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints [45628.0
 (02/10/19 17:10:00)] was greater than the threshold (10000.0) (minimum 1 datapoint for
 OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:20:48.551+0000\",\"startDate\":\"2019-10-02T17:10:00.000+0000\",
\"period\":300,\"recentDatapoints\":[45628.0],\"threshold\":10000.0}",
            "timestamp": "2019-10-02T17:20:48.554+0000",
            "value": "ALARM"
        }
    }
}
```

Modifica dello stato per un allarme di rilevamento delle anomalie

```
{
    "version": "0",
    "id": "daafc9f1-bddd-c6c9-83af-74971fcfc4ef",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-03T16:00:04Z",
    "region": "us-east-1",
    "resources": ["arn:aws:cloudwatch:us-east-1:123456789012:alarm:EC2 CPU Utilization
 Anomaly"],
    "detail": {
        "alarmName": "EC2 CPU Utilization Anomaly",
        "state": {
            "value": "ALARM",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints [0.0
 (03/10/19 15:58:00)] was less than the lower thresholds [0.020599444741798756] or
 greater than the upper thresholds [0.3006915352732461] (minimum 1 datapoint for OK ->
 ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T16:00:04.650+0000\",\"startDate\":\"2019-10-03T15:58:00.000+0000\",
```

```
\"period\":60,\"recentDatapoints\":[0.0],\"recentLowerThresholds\":
[0.020599444741798756],\"recentUpperThresholds\":[0.3006915352732461]}",
            "timestamp": "2019-10-03T16:00:04.653+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints
 [0.166666666664241 (03/10/19 15:57:00)] was not less than the lower thresholds
 [0.0206719426210418] or not greater than the upper thresholds [0.30076870222143803]
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T15:59:04.670+0000\",\"startDate\":\"2019-10-03T15:57:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.166666666664241],\"recentLowerThresholds\":
[0.0206719426210418],\"recentUpperThresholds\":[0.30076870222143803]}",
            "timestamp": "2019-10-03T15:59:04.672+0000"
        },
        "configuration": {
            "description": "Goes into alarm if CPU Utilization is out of band",
            "metrics": [{
                "id": "m1",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 60,
                    "stat": "Average"
                },
                "returnData": true
            }, {
                "id": "ad1",
                "expression": "ANOMALY_DETECTION_BAND(m1, 0.8)",
                "label": "CPUUtilization (expected)",
                "returnData": true
            }]
        }
    }
}
```

Cambio di stato per un allarme composito con un allarme soppressore

```
{
    "version": "0",
    "id": "d3dfc86d-384d-24c8-0345-9f7986db0b80",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-22T15:57:45Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "actionsSuppressedReason": "Actions suppressed by WaitPeriod",
            "value": "ALARM",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.FirstChild transitioned to ALARM
 at Friday 22 July, 2022 15:57:45 UTC",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"ALARM\",\"timestamp
\":\"2022-07-22T15:57:45.394+0000\"}}]}",
            "timestamp": "2022-07-22T15:57:45.394+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.Main was created and its alarm
 rule evaluates to OK",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:TotalNetworkTrafficTooHigh\",\"state\":{\"value\":\"OK\",
\"timestamp\":\"2022-07-14T16:28:57.770+0000\"}},{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"OK\",\"timestamp\":
\"2022-07-14T16:28:54.191+0000\"}}]}",
            "timestamp": "2022-07-22T15:56:14.552+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180
```

```
        }
    }
}
```

Creazione di un allarme composito

```
{

    "version": "0",
    "id": "91535fdd-1e9c-849d-624b-9a9f2b1d09d0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:22.289+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "alarmName": "ServiceAggregatedAlarm",
            "description": "Aggregated monitor for instance",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:22.289+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Creazione di un allarme composito con un allarme soppressore

```
{
    "version": "0",
```

```
    "id": "454773e1-09f7-945b-aa2c-590af1c3f8e0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:46Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```


Aggiornamento dell'allarme di un parametro


```
{

    "version": "0",
    "id": "bc7d3391-47f8-ae47-f457-1b4d06118d50",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:34Z",
    "region": "us-east-1",
```

```
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "operation": "update",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:13.757+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 80,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [
                {
                    "id": "86bfa85f-b14c-ebf7-8916-7da014ce23c0",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "CPUUtilization",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ],
            "alarmName": "ServerCpuTooHigh",
            "description": "Goes into alarm when server CPU utilization is too high!",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:34.267+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        },
        "previousConfiguration": {
            "evaluationPeriods": 1,
            "threshold": 70,
            "comparisonOperator": "GreaterThanThreshold",
```

```
            "treatMissingData": "ignore",
            "metrics": [
                {
                    "id": "d6bfa85f-893e-b052-a58b-4f9295c9111a",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "CPUUtilization",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ],
            "alarmName": "ServerCpuTooHigh",
            "description": "Goes into alarm when server CPU utilization is too high!",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:13.757+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Aggiornamento di un allarme composito con un allarme soppressore

```
{
    "version": "0",
    "id": "4c6f4177-6bd5-c0ca-9f05-b4151c54568b",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:56Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
```

```
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "update",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        },
        "previousConfiguration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Cancellazione di un allarme di un parametro matematico

```
{

    "version": "0",
    "id": "f171d220-9e1c-c252-5042-2677347a83ed",
    "detail-type": "CloudWatch Alarm Configuration Change",
```

```
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:07:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "operation": "delete",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:17.672+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 10000,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [{
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkIn",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkOut",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
```

```
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                },
                {
                    "id": "e1",
                    "expression": "SUM(METRICS())",
                    "label": "Total Network Traffic",
                    "returnData": true
                }
            ],
            "alarmName": "TotalNetworkTrafficTooHigh",
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:17.672+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []

        }
    }
}
```

Eliminazione di un allarme composito con un allarme soppressore

```
{
    "version": "0",
    "id": "e34592a1-46c0-b316-f614-1b17a87be9dc",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T14:00:01Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "delete",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
```

```
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Gestione degli allarmi

Modifica o elimina un allarme CloudWatch

È possibile modificare o eliminare un allarme esistente.

Il nome di un allarme esistente non può essere modificato. Puoi copiare un allarme e assegnare all'allarme un nome diverso. Per copiare un allarme, seleziona la casella di controllo accanto al nome di allarme nell'elenco di allarmi e scegli Action (Operazione), Copy (Copia).

Modifica di un allarme

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All Alarms (Tutti gli allarmi).
3. Scegli il nome dell'allarme.
4. Per aggiungere o rimuovere tag, scegli la scheda Tag e quindi Gestisci tag.
5. Per modificare altre parti dell'allarme, scegli Operazioni, Modifica.

   Viene visualizzata la pagina Specify metric and conditions (Specifica parametro e condizioni), contenente un grafico e altre informazioni sul parametro e le statistiche selezionate.
6. Per modificare il parametro, scegli Edit (Modifica), seleziona la scheda All metrics (Tutti i parametri) e procedi in uno dei seguenti modi:

- Scegli lo spazio dei nomi del servizio contenente il parametro desiderato. Continua scegliendo le opzioni così come vengono visualizzate per limitare le scelte. Quando viene visualizzato un elenco di parametri, seleziona la casella di controllo accanto al parametro desiderato.
- Nella casella di ricerca, inserisci il nome di un parametro, dimensione o ID risorsa e premi INVIO. Quindi scegli uno dei risultati e continua finché non viene visualizzato un elenco di parametri. Seleziona la casella di controllo accanto al parametro desiderato.

Scegli Select Metric (Seleziona parametro).

7. Per modificare altri aspetti dell'allarme, scegli le opzioni appropriate. Per modificare il numero di punti dati che devono essere violati affinché venga attivato lo stato `ALARM` dell'allarme o per modificare il modo in cui i dati mancanti vengono gestiti, scegli Additional configuration (Configurazione aggiuntiva).
8. Seleziona Successivo.
9. In Notification (Notifica), Auto Scaling action (Operazione Auto Scaling) e EC2 action (Operazione EC2), modifica facoltativamente le operazioni eseguite quando l'allarme viene attivato. Quindi scegli Successivo.
10. Modifica facoltativamente la descrizione dell'allarme.

    Il nome di un allarme esistente non può essere modificato. Puoi copiare un allarme e assegnare all'allarme un nome diverso. Per copiare un allarme, seleziona la casella di controllo accanto al nome di allarme nell'elenco di allarmi e scegli Action (Operazione), Copy (Copia).
11. Seleziona Successivo.
12. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Update alarm (Aggiorna allarme).

Aggiornamento di un elenco di notifiche e-mail creato usando la console di Amazon SNS

1. Apri la console Amazon SNS all'indirizzo https://console.aws.amazon.com/sns/v3/home.
2. Nel riquadro di navigazione, scegli Topics (Argomenti), quindi seleziona l'ARN per l'elenco di notifiche (argomento).
3. Esegui una di queste operazioni:

- Per aggiungere un indirizzo e-mail, scegli Create subscription (Crea abbonamento). Per Protocollo, scegli E-mail. Per Endpoint (Endpoint), immetti l'indirizzo e-mail del nuovo destinatario. Scegli Create Subscription (Crea sottoscrizione).
- Per rimuovere un indirizzo e-mail, scegli Subscription ID (ID abbonamento). Scegli Other subscription actions (Altre operazioni di abbonamento), Delete subscriptions (Elimina abbonamenti).

4. Scegli Publish to topic (Pubblica nell'argomento).

Per eliminare un allarme

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Alarms (Allarmi).
3. Seleziona la casella di controllo a sinistra del nome dell'avviso e scegli Azioni, Elimina.
4. Seleziona Delete (Elimina).

Nascondi gli allarmi di Auto Scaling

Quando visualizzi gli allarmi in AWS Management Console, puoi nascondere gli allarmi relativi ad Amazon EC2 Auto Scaling e Application Auto Scaling. Questa funzione è disponibile solo nella AWS Management Console.

Per nascondere temporaneamente gli allarmi Auto Scaling

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All alarms (Tutti gli allarmi) e seleziona Hide Auto Scaling alarms (Nascondi gli allarmi Auto Scaling).

Casi d'uso degli allarmi ed esempi

Le sezioni seguenti forniscono esempi e tutorial sugli allarmi per casi d'uso comuni.

Crea un allarme di fatturazione per monitorare gli addebiti stimati AWS

Puoi monitorare i AWS costi stimati utilizzando Amazon CloudWatch. Quando abiliti il monitoraggio degli addebiti stimati per il tuo AWS account, gli addebiti stimati vengono calcolati e inviati più volte al giorno CloudWatch come dati metrici.

I dati dei parametri di fatturazione sono archiviati nella regione Stati Uniti orientali (Virginia settentrionale) e rappresentano i costi a livello mondiale. Questi dati includono i costi stimati per ogni servizio utilizzato, oltre al totale stimato dei AWS costi. AWS

L'allarme scatta nel momento in cui la fatturazione dell'account supera il limite specificato. Viene attivato solo quando la fatturazione attuale supera la soglia. Non utilizza proiezioni in base all'utilizzo fino a un momento specifico nel mese.

Se si crea un allarme di fatturazione nel momento in cui i costi hanno già superato la soglia, l'allarme passa allo stato `ALARM` immediatamente.

> **Note**
>
> Per informazioni sull'analisi degli CloudWatch addebiti che ti sono già stati fatturati, consulta. CloudWatch fatturazione e costi

Attività

- Attivazione di avvisi di fatturazione
- Creazione di un allarme di fatturazione
- Eliminazione di un allarme di fatturazione

Attivazione di avvisi di fatturazione

Prima di poter creare un allarme per gli addebiti stimati, devi abilitare gli avvisi di fatturazione, in modo da poter monitorare gli AWS addebiti stimati e creare un allarme utilizzando i dati metrici di fatturazione. Dopo aver attivato gli avvisi di fatturazione, non è possibile disabilitare la raccolta di dati, ma è possibile eliminare qualsiasi allarme di fatturazione creato.

Dopo aver attivato gli avvisi di fatturazione per la prima volta, sono necessari circa 15 minuti prima di visualizzare i dati di fatturazione e di impostare gli allarmi di fatturazione.

Requisiti

- È necessario aver effettuato l'accesso utilizzando le credenziali dell'utente root dell'account o come utente IAM a cui è stata concessa l'autorizzazione per visualizzare le informazioni di fatturazione.
- Per gli account di fatturazione consolidata, puoi individuare i dati di fatturazione di ciascun account collegato accedendo come account di pagamento. Puoi visualizzare i dati di fatturazione per i costi stimati totali e i costi stimati per servizio per ogni account collegato aggiunto all'account consolidato.
- In un conto di fatturazione consolidato, le metriche degli account collegati ai membri vengono acquisite solo se il conto pagatore abilita la preferenza Ricevi avvisi di fatturazione. Se si modifica il conto gestione/pagante, è necessario abilitare gli avvisi di fatturazione nel nuovo conto gestione/pagante.
- L'account non deve far parte dell'Amazon Partner Network (APN) perché le metriche di fatturazione non vengono pubblicate sugli account APN. CloudWatch Per ulteriori informazioni, consulta la pagina Partner Network AWS.

Attivazione del monitoraggio dei costi stimati

1. Apri la console all'indirizzo https://console.aws.amazon.com/billing/ AWS Billing .
2. Nel riquadro di navigazione, scegli Billing preferences (Preferenze di fatturazione).
3. Per Preferenze di avviso, scegli Modifica.
4. Scegli Ricevi avvisi di CloudWatch fatturazione.
5. Scegli Save preferences (Salva preferenze).

Creazione di un allarme di fatturazione

> **Important**
>
> Prima di creare un allarme di fatturazione, imposta la regione su Stati Uniti orientali (Virginia settentrionale). I dati dei parametri di fatturazione sono archiviati in questa regione e rappresentano i costi a livello mondiale. Devi inoltre abilitare gli avvisi di fatturazione per il tuo l'account o per l'account gestione/pagante se utilizzi la fatturazione consolidata. Per ulteriori informazioni, consulta Attivazione di avvisi di fatturazione.

In questa procedura, crei un allarme che invia una notifica quando gli addebiti stimati AWS superano una soglia definita.

Per creare un allarme di fatturazione utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), quindi Create alarm (Tutti gli allarmi).
3. Scegli Crea allarme.
4. Scegli Select Metric (Seleziona parametro). In Browse (Sfoglia), scegli Billing (Fatturazione), quindi seleziona Total Estimated Charge (Addebito totale stimato).

   > **Note**
   >
   > Se non visualizzi il parametro Fatturazione/Addebito totale stimato, abilita gli avvisi di fatturazione e passa alla Regione Stati Uniti orientali (Virginia settentrionale). Per ulteriori informazioni, consulta Attivazione di avvisi di fatturazione.

5. Seleziona la casella per la EstimatedChargesmetrica, quindi scegli Seleziona metrica.
6. Per Statistic (Statistica), scegli Maximum (Massima).
7. Per Period (Periodo), scegli 6 hours (6 ore).
8. For Threshold type (Tipo di soglia), scegli Static (Statica).
9. Per EstimatedCharges Whenever is.. , scegli Maggiore.
10. Per than . . ., definisci il valore che desideri attivi l'allarme. Per un esempio, **200** USD.

    I valori delle EstimatedChargesmetriche sono solo in dollari USA (USD) e la conversione di valuta è fornita da Amazon Services LLC. Per ulteriori informazioni, consulta Cos'è AWS Billing? .

    > **Note**
    >
    > Dopo aver definito un valore di soglia, il grafico di anteprima mostra i costi stimati per il mese corrente.

11. Scegli Configurazione aggiuntiva e completa le seguenti operazioni:

    - In Datapoints to alarm (Data point per allarme), specifica 1 out of 1 (1 su 1).

- In Missing data treatment (Trattamento dei dati mancanti), scegli Treat missing data as missing (Tratta i dati mancanti come mancanti).

12. Seleziona Successivo.
13. In Notifica, assicurati che sia selezionata l'opzione In allarme. Quindi, specifica un argomento Amazon SNS per segnalare quando l'allarme si trova nello stato `ALARM`. L'argomento Amazon SNS può includere il tuo indirizzo e-mail in modo da ricevere un messaggio e-mail quando l'importo della fatturazione supera la soglia specificata.

    Puoi selezionare un argomento Amazon SNS esistente, crearne uno nuovo o utilizzare un ARN dell'argomento per inviare una notifica a un altro account. Se vuoi che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).
14. Seleziona Successivo.
15. In Name and description (Nome e descrizione), immetti un nome per l'allarme. Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII.

    - (Facoltativo) Immetti una descrizione per l'allarme. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella CloudWatch console. Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.
16. Seleziona Successivo.
17. In Preview and create (Anteprima e creazione), verifica che la configurazione sia corretta, quindi seleziona Create alarm (Crea allarme).

Eliminazione di un allarme di fatturazione

Puoi eliminare l'allarme di fatturazione quando non è più necessario.

Per eliminare un allarme di fatturazione

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Se necessario modifica la regione in Stati Uniti orientali (Virginia settentrionale). I dati dei parametri di fatturazione sono archiviati in questa regione e riflettono i costi mondiali.
3. Nel pannello di navigazione, scegli Alarms (Allarmi), All alarms (Tutti gli allarmi).
4. Seleziona la casella di controllo accanto all'allarme e scegli Actions (Operazioni), Delete (Elimina).

5. Quando viene richiesta la conferma, seleziona Yes, Delete (Sì, elimina).

Creazione di un allarme legato all'utilizzo della CPU

Puoi creare un CloudWatch allarme che invia una notifica utilizzando Amazon SNS quando lo stato dell'allarme cambia da a0K. ALARM

L'allarme modifica lo stato in ALARM quando l'utilizzo medio della CPU di un'istanza EC2 supera una specifica soglia per determinati periodi consecutivi.

Configurazione di un allarme relativo all'utilizzo della CPU utilizzando il AWS Management Console

Usa questi passaggi per AWS Management Console creare un allarme sull'utilizzo della CPU.

Per creare un allarme basato sull'utilizzo della CPU

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All Alarms (Tutti gli allarmi).
3. Scegli Crea allarme.
4. Scegli Select Metric (Seleziona parametro).
5. Nella scheda Tutte le metriche scegli Parametri EC2.
6. Scegli una categoria di parametri (ad esempio, Per-Instance Metrics).
7. Trova la riga con l'istanza che vuoi elencare nella InstanceIdcolonna e CPUUtilization nella colonna Metric Name. Seleziona la casella di controllo accanto a questa riga e scegli Seleziona metrica.
8. In Specifica metrica e condizioni, per Statistica scegli Media, scegli uno dei percentili predefiniti oppure specifichi un percentile personalizzato (ad esempio **`p95.45`**).
9. Seleziona un periodo (ad esempio, **`5 minutes`**).
10. In Conditions (Condizioni), specifica quanto segue:

    a. For Threshold type (Tipo di soglia), scegli Static (Statica).

    b. Per Ogniqualvolta CPUUtilization è, specificare Maggiore. In di..., specificare la soglia che deve attivare l'avviso per passare allo stato ALARM se l'utilizzo della CPU supera questa percentuale. Ad esempio, 70.

c. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

   Per creare un allarme M di N, specifica un numero minore per il primo valore rispetto a quello specificato per il secondo valore. Per ulteriori informazioni, consulta la pagina Valutazione di un allarme.

d. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .

e. Se l'allarme utilizza un percentile come statistica monitorata, viene visualizzata una casella Percentiles with low samples (Percentili con campioni ridotti). Utilizzala per scegliere se valutare o ignorare casi con bassa frequenza di campionamento. Se scegli ignore (maintain alarm state) (ignora (mantieni stato dell'allarme)), lo stato corrente dell'allarme viene sempre mantenuto quando la dimensione dell'esempio è troppo bassa. Per ulteriori informazioni, consulta la pagina CloudWatch Allarmi basati su percentili ed esempi di dati limitati.

11. Seleziona Next (Successivo).
12. In Notifica, scegli In allarme e seleziona un argomento SNS per notificare quando l'avviso è in stato `ALARM`

    Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

    Per fare in modo che l'allarme non invii notifiche, scegli Remove (Rimuovi).
13. Al termine, scegli Apply (Applica).
14. Inserisci un nome e una descrizione per l'allarme. Quindi scegli Successivo.

    Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.
15. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Impostazione di un allarme relativo all'utilizzo della CPU utilizzando il AWS CLI

Usa questi passaggi per AWS CLI creare un allarme sull'utilizzo della CPU.

Per creare un allarme basato sull'utilizzo della CPU

1. Imposta un argomento SNS. Per ulteriori informazioni, consulta la pagina Impostazione delle notifiche Amazon SNS.
2. Crea un allarme utilizzando il put-metric-alarmcomando seguente.

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2 --statistic
 Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --
dimensions  Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-actions
 arn:aws:sns:us-east-1:111122223333:my-topic --unit Percent
```

3. Prova l'allarme forzando la modifica dello stato dell'allarme utilizzando il set-alarm-statecomando.

   a. Modifica lo stato di un allarme da `INSUFFICIENT_DATA` a `OK`.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Modifica lo stato di un allarme da `OK` a `ALARM`.

   ```
   aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
    "initializing" --state-value ALARM
   ```

   c. Verifica di aver ricevuto una notifica sull'allarme.

Creazione di un allarme di latenza per il sistema di bilanciamento del carico che invia e-mail

Puoi impostare una notifica Amazon SNS e configurare un allarme che monitora la latenza che supera i 100 ms per Classic Load Balancer.

Impostazione di un allarme di latenza utilizzando il AWS Management Console

Usa questi passaggi per creare un allarme AWS Management Console di latenza del load balancer.

Per creare un allarme di latenza per il load balancer

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All Alarms (Tutti gli allarmi).
3. Scegli Crea allarme.
4. In CloudWatch Metriche per categoria, scegli la categoria Metriche ELB.
5. Seleziona la riga con il sistema Classic Load Balancer e il parametro Latency (Latenza).
6. Per la statistica, scegli Average (Media), seleziona uno dei percentili predefiniti oppure specifica un percentile personalizzato (ad esempio, **`p95.45`**).
7. Per il periodo, scegli 1 Minute (1 minuto).
8. Seleziona Successivo.
9. In Alarm Threshold (Soglia di allarme), immetti un nome univoco per l'allarme (ad esempio, **`myHighCpuAlarm`**) e una descrizione dell'allarme (ad esempio, **`Alarm when Latency exceeds 100s`**). I nomi degli allarmi devono contenere solo caratteri UTF-8 e non possono contenere caratteri di controllo ASCII

   Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.
10. In Whenever (Ogni volta che), per is (è), scegli **>** e immetti **`0.1`**. In for (per), immetti **`3`**.
11. In Additional settings (Impostazioni aggiuntive), per Treat missing data as (Tratta i dati mancanti come), seleziona ignore (maintain alarm state) (ignora (mantieni lo stato dell'allarme)), in modo tale che i punti dati mancanti non attivino le modifiche di stato dell'allarme.

    In Percentiles with low samples (Percentili con campioni ridotti), scegli ignore (maintain the alarm state) (ignora (mantieni lo stato dell'allarme)) in modo che l'allarme consideri solo situazioni con un numero sufficiente di esempi di dati.
12. In Actions (Operazioni), in Whenever this alarm (Ogni volta che questo allarme), scegli State is ALARM (Lo stato è ALLARME). Per Send notification to (Invia notifica a), scegli un argomento SNS esistente o creane uno nuovo.

    Per creare un argomento SNS, seleziona New list (Nuovo elenco). Per Send notification to (Invia notifica a) immetti un nome per l'argomento SNS (ad esempio, **`myHighCpuAlarm`**) e per Email list (Elenco e-mail) immetti un elenco di indirizzi e-mail separati da virgola a cui inviare una notifica quando l'allarme passa nello stato `ALARM`. Viene inviato a ciascun indirizzo un'e-mail di

conferma della sottoscrizione all'argomento. È necessario confermare l'abbonamento prima del possibile invio delle notifiche.

13. Scegli Create Alarm (Crea allarme).

Impostazione di un allarme di latenza utilizzando il AWS CLI

Usa questi passaggi per creare un allarme AWS CLI di latenza del load balancer.

Per creare un allarme di latenza per il load balancer

1. Imposta un argomento SNS. Per ulteriori informazioni, consulta la pagina Impostazione delle notifiche Amazon SNS.
2. Crea l'allarme utilizzando il put-metric-alarmcomando seguente:

```
aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description "Alarm
 when Latency exceeds 100s" --metric-name Latency --namespace AWS/ELB --statistic
 Average --period 60 --threshold 100 --comparison-operator GreaterThanThreshold --
dimensions Name=LoadBalancerName,Value=my-server --evaluation-periods 3 --alarm-
actions arn:aws:sns:us-east-1:111122223333:my-topic --unit Seconds
```

3. Verifica l'allarme forzando la modifica dello stato dell'allarme utilizzando il set-alarm-statecomando.

   a. Modifica lo stato di un allarme da `INSUFFICIENT_DATA` a `OK`.

   ```
   aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Modifica lo stato di un allarme da `OK` a `ALARM`.

   ```
   aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
    "initializing" --state-value ALARM
   ```

   c. Verifica di aver ricevuto una notifica tramite e-mail sull'allarme.

Creazione di un allarme della velocità di trasmissione effettiva dell'archiviazione che invia e-mail

Puoi impostare una notifica di SNS e configurare un allarme che si attiva quando Amazon EBS supera 100 MB di throughput.

Configurazione di un allarme per il throughput dello storage tramite AWS Management Console

Utilizza questi passaggi per AWS Management Console creare un allarme basato sul throughput di Amazon EBS.

Per creare un allarme del throughput dello storage

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All Alarms (Tutti gli allarmi).
3. Scegli Crea allarme.
4. In EBS Metrics (Parametri di EBS), scegli la categoria di un parametro.
5. Seleziona la riga con il volume e la VolumeWriteBytesmetrica.
6. Per la statistica, scegli Average (Media). Per il periodo, scegli 5 Minute (5 minuto). Seleziona Successivo.
7. In Alarm Threshold (Soglia di allarme), immetti un nome univoco per l'allarme (ad esempio, **`myHighWriteAlarm`**) e una descrizione dell'allarme (ad esempio, **`VolumeWriteBytes exceeds 100,000 KiB/s`**). Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.
8. In Whenever (Ogni volta che), per is (è), scegli **>** e immetti **`100000`**. Per for (per), immetti **`15`** periodi consecutivi.

   Viene visualizzata una rappresentazione grafica della soglia in Alarm Preview (Anteprima allarme).
9. In Additional settings (Impostazioni aggiuntive), per Treat missing data as (Tratta i dati mancanti come), seleziona ignore (maintain alarm state) (ignora (mantieni lo stato dell'allarme)), in modo tale che i punti dati mancanti non attivino le modifiche di stato dell'allarme.

10. In Actions (Operazioni), in Whenever this alarm (Ogni volta che questo allarme), scegli State is ALARM (Lo stato è ALLARME). Per Send notification to (Invia notifica a), scegli un argomento SNS esistente o creane uno nuovo.

    Per creare un argomento SNS, seleziona New list (Nuovo elenco). Per Send notification to (Invia notifica a) immetti un nome per l'argomento SNS (ad esempio, **myHighCpuAlarm**) e per Email list (Elenco e-mail) immetti un elenco di indirizzi e-mail separati da virgola a cui inviare una notifica quando l'allarme passa nello stato `ALARM`. Viene inviato a ciascun indirizzo un'e-mail di conferma della sottoscrizione all'argomento. È necessario confermare la sottoscrizione prima che le notifiche possano essere inviate a un indirizzo e-mail.

11. Scegli Crea allarme.

Configurazione di un allarme di throughput di archiviazione utilizzando il AWS CLI

Utilizza questi passaggi per AWS CLI creare un allarme basato sul throughput di Amazon EBS.

Per creare un allarme del throughput dello storage

1. Creare un argomento SNS. Per ulteriori informazioni, consulta la pagina Impostazione delle notifiche Amazon SNS.
2. Crea l'allarme.

```
aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description "Alarm
 when EBS volume exceeds 100MB throughput" --metric-name VolumeReadBytes --
namespace AWS/EBS --statistic Average --period 300 --threshold 100000000 --
comparison-operator GreaterThanThreshold --dimensions Name=VolumeId,Value=my-
volume-id --evaluation-periods 3 --alarm-actions arn:aws:sns:us-
east-1:111122223333:my-alarm-topic --insufficient-data-actions arn:aws:sns:us-
east-1:111122223333:my-insufficient-data-topic
```

3. Testa l'allarme forzando la modifica dello stato dell'allarme utilizzando il set-alarm-statecomando.

    a. Modifica lo stato di un allarme da `INSUFFICIENT_DATA` a `OK`.

    ```
    aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
     "initializing" --state-value OK
    ```

    b. Modifica lo stato di un allarme da `OK` a `ALARM`.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value ALARM
```

c. Modifica lo stato di un allarme da `ALARM` a `INSUFFICIENT_DATA`.

```
aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
 "initializing" --state-value INSUFFICIENT_DATA
```

d. Verifica di aver ricevuto una notifica tramite e-mail sull'allarme.

Crea un allarme sulle metriche dei contatori di Performance Insights da un database AWS

CloudWatch include una funzione matematica di metrica DB_PERF_INSIGHTS che puoi utilizzare per importare i parametri dei contatori di Performance Insights da Amazon Relational Database CloudWatch Service e Amazon DocumentDB (con compatibilità MongoDB). DB_PERF_INSIGHTS introduce il parametro `DBLoad` anche a intervalli inferiori al minuto. Puoi CloudWatch impostare allarmi in base a questi parametri.

Per ulteriori informazioni su Approfondimenti sulle prestazioni di Amazon RDS, consulta Utilizzo di Approfondimenti sulle prestazioni di Amazon RDS.

Per ulteriori informazioni su Approfondimenti sulle prestazioni di Amazon DocumentDB, consulta Monitoraggio con Approfondimenti sulle prestazioni.

Il rilevamento delle anomalie non è supportato per gli allarmi basati sulla funzione DB_PERF_INSIGHTS.

> **Note**
>
> I parametri ad alta risoluzione con granularità inferiore al minuto recuperati da DB_PERF_INSIGHTS sono applicabili solo al parametro DBLoad o ai parametri del sistema operativo se è stato abilitato il monitoraggio avanzato a una risoluzione più elevata. Per ulteriori informazioni sul monitoraggio avanzato di Amazon RDS, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.
> È possibile creare un allarme ad alta risoluzione utilizzando la funzione DB_PERF_INSIGHTS. L'intervallo di valutazione massimo per un allarme ad alta risoluzione

è di tre ore. È possibile utilizzare la CloudWatch console per rappresentare graficamente le metriche recuperate con la funzione DB_PERF_INSIGHTS per qualsiasi intervallo di tempo.

Creazione di un allarme basato sui parametri di Approfondimenti sulle prestazioni

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch
2. Nel pannello di navigazione, scegli Alarms (Allarmi), quindi Create alarm (Tutti gli allarmi).
3. Scegli Crea allarme.
4. Scegli Select Metric (Seleziona parametro).
5. Scegli il menu a discesa Aggiungi formula, quindi seleziona Parametri delle prestazioni del database, DB_PERF_INSIGHTS dall'elenco.

   Dopo aver scelto DB_PERF_INSIGHTS, viene visualizzata una casella di espressione matematica in cui applicare o modificare le espressioni matematiche.
6. Inserisci l'espressione matematica DB_PERF_INSIGHTS nella relativa casella, quindi scegli Applica.

   Ad esempio, **`DB_PERF_INSIGHTS('RDS', 'db-ABCDEFGHIJKLMNOPQRSTUVWXY1', 'os.cpuUtilization.user.avg')`**

   > **Important**
   >
   > Quando si utilizza l'espressione matematica DB_PERF_INSIGHTS, è necessario specificare l'ID univoco della risorsa del database. È diverso dall'identificatore del database. Per trovare l'ID della risorsa del database nella console Amazon RDS, scegli l'istanza database per vederne i dettagli. Quindi seleziona la scheda Configurazione. ID risorsa è visualizzato nella sezione Configurazione.

   Per informazioni sulla funzione DB_PERF_INSIGHTS e su altre funzioni disponibili per la formula dei parametri, consulta Sintassi e funzioni della matematica dei parametri.
7. Scegli Select Metric (Seleziona parametro).

   Viene visualizzata la pagina Specify metric and conditions (Specifica parametro e condizioni), contenente un grafico e altre informazioni relative all'espressione matematica selezionata.

8. Per Whenever ***expression*** is (Ogni volta che espressione è), specifica se l'espressione deve essere maggiore di, minore di o uguale alla soglia. In than... (che...), specifica il valore di soglia.
9. Scegli Additional configuration (Configurazione aggiuntiva). In Datapoints to Alarm (Punti dati all'allarme), specifica il numero di periodi di valutazione (punti dati) che devono essere nello stato `ALARM` per attivare l'allarme. Se i due valori corrispondono, crea un allarme che passa nello stato `ALARM` se si verifica una violazione durante tali periodi consecutivi.

   Per creare un allarme M di N, specifica un numero minore per il primo valore rispetto a quello specificato per il secondo valore. Per ulteriori informazioni, consulta la pagina Valutazione di un allarme.
10. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni, consulta la pagina Configurazione del modo in cui gli allarmi trattano i dati mancanti CloudWatch .
11. Seleziona Next (Successivo).
12. In Notification (Notifica), seleziona un argomento SNS per segnalare quando l'allarme è nello stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.

    Per fare in modo che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).

    Per fare in modo che l'allarme non invii notifiche, scegli Remove (Rimuovi).
13. Per fare in modo che l'allarme esegua operazioni Auto Scaling, EC2, Lambda o Systems Manager scegli il pulsante appropriato e scegli lo stato di allarme e l'operazione da eseguire. Se scegli una funzione Lambda come operazione di allarme, specifichi il nome della funzione o l'ARN e, facoltativamente, puoi scegliere una versione specifica della funzione.

    Gli allarmi possono eseguire le operazioni Systems Manager solo quando entrano nello stato ALARM. Per ulteriori informazioni sulle azioni di Systems Manager, vedere Configurazione per la creazione CloudWatch OpsItems da allarmi e Creazione di incidenti.

    > **Note**
    >
    > Per creare un allarme che esegua un'operazione SSM Incident Manager, è necessario disporre di determinate autorizzazioni. Per ulteriori informazioni, vedere Esempi di policy basate sull'identità per AWS Systems Manager Incident Manager.
14. Al termine, scegli Apply (Applica).

15. Inserisci un nome e una descrizione per l'allarme. Quindi scegli Successivo.

    Il nome deve contenere solo caratteri UTF-8 e non può contenere caratteri di controllo ASCII. La descrizione può includere la formattazione del markdown, che viene visualizzata solo nella scheda Dettagli dell'allarme nella console. CloudWatch Il markdown può essere utile per aggiungere collegamenti ai runbook o ad altre risorse interne.
16. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Creazione di allarmi per arrestare, terminare, riavviare o recuperare un'istanza EC2

Utilizzando Amazon CloudWatch Alarm Actions, puoi creare allarmi che interrompono, terminano, riavviano o ripristinano automaticamente le tue istanze EC2. Puoi utilizzare le operazioni di arresto o termine per aiutarti a risparmiare denaro quando non necessiti più dell'esecuzione di un'istanza. Puoi utilizzare le operazioni di riavvio e recupero per riavviare automaticamente tali istanze o recuperarle in un nuovo hardware, se si verifica un danneggiamento del sistema.

Esistono diversi scenari in cui potresti voler arrestare o terminare automaticamente l'istanza. Ad esempio, potresti disporre di istanze dedicate a processi di elaborazione della retribuzione in batch o ad attività di calcolo scientifico che vengono eseguite per un periodo di tempo, dopodiché completano il proprio lavoro. Anziché lasciare tali istanze inattive (accumulando addebiti), puoi arrestarle o terminarle per risparmiare denaro. La differenza principale tra l'uso delle operazioni di allarme di arresto o di terminazione consiste nel poter riavviare comodamente un'istanza arrestata se è necessario eseguirla in un secondo momento, mantenendo gli stessi ID istanza e volume radice. Tuttavia, non puoi riavviare un'istanza terminata. Al contrario, è necessario avviare una nuova istanza.

Puoi aggiungere le azioni di arresto, terminazione o riavvio a qualsiasi allarme impostato su un parametro Amazon EC2 per istanza, inclusi i parametri di monitoraggio di base e dettagliati forniti da CloudWatch Amazon (nello spazio dei nomi AWS/EC2), oltre a qualsiasi metrica personalizzata che includa la dimensione InstanceId "=», purché il valore si riferisca a un'istanza Amazon EC2 valida in esecuzione. InstanceId Puoi anche aggiungere l'azione di ripristino agli allarmi impostati su qualsiasi parametro per istanza Amazon EC2, ad eccezione di `StatusCheckFailed_Instance`.

Per impostare un'azione di CloudWatch allarme in grado di riavviare, arrestare o terminare un'istanza, è necessario utilizzare un ruolo IAM collegato al servizio, AWSServiceRoleForCloudWatchEvents. Il

ruolo AWSServiceRoleForCloudWatchEvents IAM consente di AWS eseguire azioni di allarme per tuo conto.

Per creare il ruolo collegato al servizio per CloudWatch Events, usa il seguente comando:

```
aws iam create-service-linked-role --aws-service-name events.amazonaws.com
```

Supporto della console

Puoi creare allarmi utilizzando la CloudWatch console o la console Amazon EC2. Le procedure descritte in questa documentazione utilizzano la CloudWatch console. Per le procedure che utilizzano la console Amazon EC2, consulta la pagina relativa alla creazione di allarmi che arrestano, terminano, riavviano o ripristinano un'istanza nella Guida per l'utente di Amazon EC2 per istanze Linux.

Autorizzazioni

Se utilizzi un account AWS Identity and Access Management (IAM) per creare o modificare un allarme che esegue azioni EC2 o azioni Systems Manager OpsItem , devi disporre dell'`iam:CreateServiceLinkedRole`autorizzazione.

Indice

Aggiungere azioni di interruzione agli CloudWatch allarmi Amazon

Puoi creare un allarme per arrestare un'istanza Amazon EC2 al raggiungimento di una determinata soglia. Ad esempio, potresti eseguire istanze di sviluppo o di test e occasionalmente dimenticare di disattivarle. Puoi creare un allarme che viene attivato quando la percentuale di utilizzo medio della CPU è inferiore al 10% per 24 ore, segnalando che la CPU è inattiva e non più in uso. Puoi regolare la soglia, la durata e il periodo di tempo in base alle tue esigenze. È possibile inoltre aggiungere una notifica SNS, in modo da ricevere un'e-mail all'attivazione dell'allarme.

Le istanze Amazon EC2 che utilizzano un volume Amazon Elastic Block Store come dispositivo root possono essere arrestate o terminate, mentre le istanze che utilizzano l'instance store come dispositivo root possono solo essere terminate.

Per creare un allarme per interrompere un'istanza inattiva utilizzando la console Amazon CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All alarms (Tutti gli allarmi).
3. Scegli Crea allarme.
4. Scegli Select Metric (Seleziona parametro).
5. Nel campo Spazi dei nomi AWS , scegli EC2.
6. Esegui questa operazione:
    a. Scegli Per-Instance Metrics (Parametri per istanza).
    b. Seleziona la casella di controllo nella riga con l'istanza e il parametro CPUUtilization corretti.
    c. Seleziona la scheda Graphed metrics (Parametri nel grafico).
    d. Per la statistica, scegli Average (Media).
    e. Seleziona un periodo (ad esempio, **1 Hour**).
    f. Scegli Select Metric (Seleziona parametro).
7. Nella fase Define Alarm (Definisci allarme), effettua le operazioni seguenti:
    a. In Conditions (Condizioni), scegli Static (Statico).
    b. In Whenever CPUUtilization is (Ogniqualvolta CPUUtilization è), scegli Lower (Più basso).
    c. Per than (di), inserisci **10**.
    d. Seleziona Successivo.
    e. Sotto Notification (Notifica), in Send notification to (Invia notifica a), seleziona un argomento SNS esistente o creane uno nuovo.

       Per creare un argomento SNS, seleziona New list (Nuovo elenco). In Send notification to (Invia notifica a), digita un nome per l'argomento SNS (ad esempio, Stop_EC2_Instance). In Email list (Elenco e-mail), digita un elenco di indirizzi e-mail separati da virgola a cui inviare una notifica quando l'allarme passa allo stato `ALARM`. Viene inviato a ciascun indirizzo un'e-mail di conferma della sottoscrizione all'argomento. È necessario confermare la sottoscrizione prima che le notifiche possano essere inviate a un indirizzo e-mail.
    f. Seleziona Add EC2 action (Aggiungi operazione EC2).

g. Per Alarm state trigger (Attivazione stato allarme), scegli In alarm (In allarme). Per Take the following action (Esegui la seguente operazione), scegli Stop this instance (Arresta questa istanza).

h. Seleziona Successivo.

i. Inserisci un nome e una descrizione per l'allarme. Il nome deve contenere solo caratteri ASCII. Quindi scegli Successivo.

j. In Preview and create (Visualizza anteprima e crea), conferma che le informazioni e le condizioni sono quelle desiderate, quindi scegli Create alarm (Crea allarme).

Aggiungere azioni di terminazione agli allarmi Amazon CloudWatch

Puoi creare un allarme per terminare automaticamente un'istanza EC2 al raggiungimento di una determinata soglia, purché non sia abilitata la protezione da cessazione dell'istanza. Ad esempio, potresti voler terminare un'istanza una volta completato il lavoro e non averne più bisogno. Se intendessi utilizzare l'istanza in un secondo momento, sarebbe necessario arrestare l'istanza anziché terminarla. Per informazioni sull'abilitazione e la disabilitazione della protezione da cessazione delle istanze, consulta la pagina relativa all'abilitazione della protezione da cessazione delle istanze nella Guida per l'utente di Amazon EC2 per istanze Linux.

Per creare un allarme per terminare un'istanza inattiva utilizzando la console Amazon CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Alarms (Allarmi), Create Alarm (Crea allarme).
3. Nella fase Select Metric (Seleziona parametro), effettua le operazioni seguenti:

   a. In EC2 Metrics (Parametri di EC2), seleziona Per-Instance Metrics (Parametri per istanza).

   b. Seleziona la riga con l'istanza e il parametro CPUUtilization.

   c. Per la statistica, scegli Average (Media).

   d. Seleziona un periodo (ad esempio, **`1 Hour`**).

   e. Seleziona Successivo.
4. Nella fase Define Alarm (Definisci allarme), effettua le operazioni seguenti:

   a. In Alarm Threshold (Soglia allarme), digita un nome per l'allarme (ad esempio, "Termina istanza EC2") e una descrizione dell'allarme (ad esempio, "Termina istanza EC2 quando la CPU è inattiva troppo a lungo"). I nomi degli allarmi devono contenere solo caratteri ASCII.

b. In Whenever (Ogni volta che), in is (è), scegli **<** e digita **10**. Per for (per), digita **24** periodi consecutivi.

   Viene visualizzata una rappresentazione grafica della soglia in Alarm Preview (Anteprima allarme).

c. Sotto Notification (Notifica), in Send notification to (Invia notifica a), seleziona un argomento SNS esistente o creane uno nuovo.

   Per creare un argomento SNS, seleziona New list (Nuovo elenco). In Send notification to (Invia notifica a), digita un nome per l'argomento SNS (ad esempio, Terminate_EC2_Instance). In Email list (Elenco e-mail), digita un elenco di indirizzi e-mail separati da virgola a cui inviare una notifica quando l'allarme passa allo stato `ALARM`. Viene inviato a ciascun indirizzo un'e-mail di conferma della sottoscrizione all'argomento. È necessario confermare la sottoscrizione prima che le notifiche possano essere inviate a un indirizzo e-mail.

d. Seleziona EC2 Action (Operazione EC2).

e. In Whenever this alarm (Ogniqualvolta questo allarme), seleziona State is ALARM (Lo stato è ALLARME). In Take this action (Esegui questa operazione), seleziona Terminate this instance (Termina questa istanza).

f. Scegli Crea allarme.

Aggiungere azioni di riavvio agli allarmi Amazon CloudWatch

Puoi creare un CloudWatch allarme Amazon che monitora un'istanza Amazon EC2 e riavvia automaticamente l'istanza. L'operazione di allarme di riavvio è consigliata per gli errori di controllo dello stato dell'istanza (contrariamente all'operazione di allarme di recupero, adatta agli errori di controllo dello stato del sistema). Il riavvio di un'istanza equivale al riavvio di un sistema operativo. Nella maggior parte dei casi, sono necessari pochi minuti per riavviare l'istanza. Quando riavvii un'istanza, questa rimane sullo stesso host fisico, in modo che l'istanza conservi il proprio nome DNS pubblico, indirizzo IP privato e tutti i dati presenti nei volumi instance store.

Il riavvio di un'istanza non avvia una uovo ora di fatturazione di istanza, a differenza dell'arresto e del riavvio dell'istanza. Per ulteriori informazioni sul riavvio di un'istanza, consulta l'argomento relativo al riavvio di un'istanza nella Guida per l'utente di Amazon EC2 per le istanze Linux.

> **Important**
>
> Per evitare una race condition tra le operazioni di riavvio e di recupero, evita di impostare lo stesso periodo di valutazione per entrambi gli allarmi di riavvio e di recupero. È consigliabile impostare gli allarmi di riavvio su tre periodi di valutazione di un minuto ciascuno.

Per creare un allarme per riavviare un'istanza utilizzando la console Amazon CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Alarms (Allarmi), Create Alarm (Crea allarme).
3. Nella fase Select Metric (Seleziona parametro), effettua le operazioni seguenti:
   a. In EC2 Metrics (Parametri di EC2), seleziona Per-Instance Metrics (Parametri per istanza).
   b. Seleziona la riga con l'istanza e la metrica StatusCheckFailed_Instance.
   c. Per la statistica, seleziona Minimum (Minimo).
   d. Seleziona un periodo (ad esempio, **`1 Minute`**).
   e. Seleziona Successivo.
4. Nella fase Define Alarm (Definisci allarme), effettua le operazioni seguenti:
   a. In Alarm Threshold (Soglia allarme), digita un nome per l'allarme (ad esempio, "Riavvia istanza EC2") e una descrizione dell'allarme (ad esempio, "Riavvia istanza EC2 quando i controlli dello stato hanno esito negativo). I nomi degli allarmi devono contenere solo caratteri ASCII.
   b. In Whenever (Ogni volta che), in is (è) scegli **>** e digita **`0`**. Per for (per), digita **`3`** periodi consecutivi.

      Viene visualizzata una rappresentazione grafica della soglia in Alarm Preview (Anteprima allarme).
   c. Sotto Notification (Notifica), in Send notification to (Invia notifica a), seleziona un argomento SNS esistente o creane uno nuovo.

      Per creare un argomento SNS, seleziona New list (Nuovo elenco). In Send notification to (Invia notifica a), digita un nome per l'argomento SNS (ad esempio, Reboot_EC2_Instance). In Email list (Elenco e-mail), digita un elenco di indirizzi e-mail separati da virgola a cui inviare una notifica quando l'allarme passa allo stato `ALARM`. Viene inviato a ciascun

indirizzo un'e-mail di conferma della sottoscrizione all'argomento. È necessario confermare la sottoscrizione prima che le notifiche possano essere inviate a un indirizzo e-mail.

d. Seleziona EC2 Action (Operazione EC2).

e. In Whenever this alarm (Ogniqualvolta questo allarme), seleziona State is ALARM (Lo stato è ALLARME). In Take this action (Esegui questa operazione), seleziona Reboot this instance (Riavvia questa istanza).

f. Scegli Crea allarme.

Aggiungere azioni di ripristino agli CloudWatch allarmi Amazon

Puoi creare un CloudWatch allarme Amazon che monitora un'istanza Amazon EC2 e ripristina automaticamente l'istanza se viene danneggiata a causa di un guasto hardware sottostante o di un problema che AWS richiede la riparazione. Le istanze terminate non possono essere recuperate. Un'istanza recuperata è identica all'istanza originale, incluso l'ID istanza, gli indirizzi IP privati, gli indirizzi IP elastici e tutti i metadati dell'istanza.

Quando viene attivato l'allarme `StatusCheckFailed_System` e viene avviata l'operazione di ripristino, riceverai una notifica dall'argomento Amazon SNS selezionato al momento della creazione dell'allarme e dell'associazione dell'operazione di ripristino. Durante il recupero dell'istanza, l'istanza viene migrata durante un riavvio di istanza e tutti i dati in memoria andranno persi. Una volta completato il processo, l'informazione viene pubblicata nell'argomento SNS configurato per l'allarme. Tutti coloro che hanno eseguito la sottoscrizione a questo argomento SNS riceveranno una notifica e-mail che include lo stato del tentativo di recupero ed eventuali ulteriori istruzioni. Noterai un riavvio di istanza nell'istanza recuperata.

L'operazione di recupero può essere utilizzata solo con `StatusCheckFailed_System`, non con `StatusCheckFailed_Instance`.

Esempi di problemi che causano il mancato superamento dei controlli dello stato del sistema:

- Perdita di connettività di rete
- Perdita di alimentazione elettrica del sistema
- Problemi di software sull'host fisico
- Problemi hardware sull'host fisico che incidono sulla raggiungibilità della rete

L'azione di ripristino è supportata solo su alcuni tipi di istanza. Per ulteriori informazioni sui tipi di istanza supportati e su altri requisiti, consulta Ripristino dell'istanza e Requisiti.

> **Important**
>
> Per evitare una race condition tra le operazioni di riavvio e di recupero, evita di impostare lo stesso periodo di valutazione per entrambi gli allarmi di riavvio e di recupero. Consigliamo di impostare gli allarmi di recupero su periodi di valutazione di un minuto ciascuno e gli allarmi di riavvio su tre periodi di valutazione di un minuto ciascuno.

Per creare un allarme per ripristinare un'istanza utilizzando la CloudWatch console Amazon

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Alarms (Allarmi), Create Alarm (Crea allarme).
3. Nella fase Select Metric (Seleziona parametro), effettua le operazioni seguenti:
   a. In EC2 Metrics (Parametri di EC2), seleziona Per-Instance Metrics (Parametri per istanza).
   b. Seleziona la riga con l'istanza e la metrica StatusCheckFailed_System.
   c. Per la statistica, seleziona Minimum (Minimo).
   d. Seleziona un periodo (ad esempio, **`1 Minute`**).

      > **Important**
      >
      > Per evitare una race condition tra le operazioni di riavvio e di recupero, evita di impostare lo stesso periodo di valutazione per entrambi gli allarmi di riavvio e di recupero. È consigliabile impostare gli allarmi di recupero su due periodi di valutazione di un minuto ciascuno.

   e. Seleziona Successivo.
4. Nella fase Define Alarm (Definisci allarme), effettua le operazioni seguenti:
   a. In Alarm Threshold (Soglia allarme), digita un nome per l'allarme (ad esempio, "Recupera istanza EC2") e una descrizione dell'allarme (ad esempio, "Recupera istanza EC2 quando i controlli dello stato hanno esito negativo). I nomi degli allarmi devono contenere solo caratteri ASCII.
   b. In Whenever (Ogni volta che), in is (è) scegli **>** e digita **`0`**. Per for (per), digita **2** periodi consecutivi.

c. Sotto Notification (Notifica), in Send notification to (Invia notifica a), seleziona un argomento SNS esistente o creane uno nuovo.

   Per creare un argomento SNS, seleziona New list (Nuovo elenco). In Send notification to (Invia notifica a), digita un nome per l'argomento SNS (ad esempio, Recover_EC2_Instance). In Email list (Elenco e-mail), digita un elenco di indirizzi e-mail separati da virgola a cui inviare una notifica quando l'allarme passa allo stato `ALARM`. Viene inviato a ciascun indirizzo un'e-mail di conferma della sottoscrizione all'argomento. È necessario confermare la sottoscrizione prima che le notifiche possano essere inviate a un indirizzo e-mail.

d. Seleziona EC2 Action (Operazione EC2).

e. In Whenever this alarm (Ogniqualvolta questo allarme), seleziona State is ALARM (Lo stato è ALLARME). In Take this action (Esegui questa operazione), seleziona Recover this instance (Recupera questa istanza).

f. Scegli Crea allarme.

Visualizzazione della cronologia degli allarmi e delle operazioni attivati

Puoi visualizzare la cronologia degli allarmi e delle azioni nella CloudWatch console Amazon. Amazon CloudWatch conserva la cronologia degli ultimi 30 giorni di allarmi e azioni.

Visualizzazione della cronologia degli allarmi e delle operazioni attivati

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, selezionare Alarms (Allarmi), quindi selezionare un allarme.
3. Per visualizzare la transizione di stato più recente insieme ai valori di tempo e di parametro, seleziona Details (Dettagli).
4. Per visualizzare le voci più recenti della cronologia, selezionare History (Cronologia).

Allarmi e tag

I tag sono coppie chiave-valore che possono aiutarti a organizzare e classificare le tue risorse. Puoi utilizzarle anche per definire l'ambito delle autorizzazioni dell'utente, assegnandogli l'autorizzazione ad accedere o modificare solo le risorse con determinati valori di tag. Per informazioni più generali sull'etichettatura delle risorse, consulta Etichettare le risorse AWS

L'elenco seguente spiega alcuni dettagli su come funziona l'etichettatura con gli allarmi. CloudWatch

- Per poter impostare o aggiornare i tag per una CloudWatch risorsa, devi accedere a un account che dispone dell'`cloudwatch:TagResource`autorizzazione. Ad esempio, per creare un allarme e impostarne i tag, è necessario disporre dell'`cloudwatch:TagResource`autorizzazione oltre all'`the cloudwatch:PutMetricAlarm` autorizzazione. Ti consigliamo di assicurarti che tutti i membri dell'organizzazione che creeranno o aggiorneranno CloudWatch le risorse dispongano dell'`cloudwatch:TagResource`autorizzazione.
- I tag possono essere utilizzati per il controllo delle autorizzazioni basato sui tag. Ad esempio, le autorizzazioni degli utenti o dei ruoli IAM possono includere condizioni per limitare le CloudWatch chiamate a risorse specifiche in base ai relativi tag. Tuttavia, tieni presente quanto segue
  - I tag con nomi che iniziano con non `aws:` possono essere utilizzati per il controllo delle autorizzazioni basato sui tag.
  - Gli allarmi compositi non supportano il controllo delle autorizzazioni basato su tag.

Application Signals

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback @amazon .com.

Utilizzate CloudWatch Application Signals per strumentare automaticamente le vostre applicazioni AWS in modo da monitorare lo stato attuale delle applicazioni e tenere traccia delle prestazioni delle applicazioni a lungo termine rispetto agli obiettivi aziendali. Application Signals ti offre una visione unificata e incentrata sulle applicazioni di applicazioni, servizi e dipendenze e ti aiuta a monitorare e valutare lo stato delle applicazioni.

- Consenti ad Application Signals di raccogliere automaticamente parametri e tracce dalle tue applicazioni e di visualizzare parametri chiave come volume delle chiamate, disponibilità, latenza, guasti ed errori. Visualizza e valuta rapidamente l'integrità operativa attuale e se le tue applicazioni stanno raggiungendo i loro obiettivi di prestazione a lungo termine, senza scrivere codice personalizzato o creare pannelli di controllo.
- Crea e monitora gli obiettivi a livello di servizio (SLO) con Application Signals. Crea e monitora facilmente lo stato degli SLO relativi alle CloudWatch metriche, incluse le nuove metriche applicative standard raccolte da Application Signals. Visualizza e monitora lo stato dell'indicatore del livello di servizio (SLI) dei tuoi servizi applicativi all'interno di un elenco di servizi e di una mappa topologica. Crea allarmi per monitorare i tuoi SLO e monitora i nuovi parametri standard delle applicazioni che Application Signals raccoglie.
- Visualizza una mappa della topologia delle applicazioni che Application Signals rileva automaticamente, che offre una rappresentazione visiva delle applicazioni, delle dipendenze e della loro connettività.
- Application Signals funziona con CloudWatch RUM, CloudWatchSynthetics canaries Amazon EC2 Auto Scaling e per visualizzare le pagine dei clienti AWS Service Catalog AppRegistry, Synthetics canaries e i nomi delle applicazioni all'interno di dashboard e mappe.

Utilizza Application Signals per il monitoraggio quotidiano delle applicazioni

Usa Application Signals all'interno della CloudWatch console, come parte del monitoraggio quotidiano delle applicazioni:

1. Se hai creato obiettivi del livello di servizio (SLO) per i tuoi servizi, inizia dalla pagina Obiettivi del livello di servizio (SLO). In questo modo puoi avere una visione immediata dello stato dei servizi e delle operazioni più importanti. Scegli il nome del servizio o dell'operazione per uno SLO per aprire la pagina dei dettagli del servizio e visualizzare informazioni dettagliate sul servizio durante la risoluzione dei problemi.
2. Apri la pagina Servizi per vedere un riepilogo di tutti i tuoi servizi e visualizzare rapidamente i servizi con il tasso di errore o la latenza più elevati. Se hai creato degli SLO, consulta la tabella Servizi per vedere quali servizi presentano indicatori di livello di servizio (SLI) non integri. Se un particolare servizio non è integro, selezionalo per aprire la pagina dei dettagli del servizio e visualizzare le operazioni del servizio, le dipendenze, i canali Synthetics e le richieste client. Seleziona un punto in un grafico per visualizzare le tracce correlate in modo da poter risolvere e identificare la causa principale dei problemi operativi.
3. Se sono stati implementati nuovi servizi o le dipendenze sono cambiate, apri la Mappa dei servizi per esaminare la topologia dell'applicazione. Visualizza una mappa delle tue applicazioni che mostra la relazione tra client, canary Synthetics, servizi e dipendenze. Visualizza rapidamente lo stato dello SLI, visualizza i parametri chiave come il volume delle chiamate, la frequenza di errore e la latenza e approfondisci per visualizzare informazioni più dettagliate nella pagina dei dettagli del servizio.

L'uso di Application Signals comporta costi. Per informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

> **Note**
>
> Non è necessario abilitare Application Signals per utilizzare CloudWatch Synthetics CloudWatch , RUM o Evidently. CloudWatch Tuttavia, Synthetics CloudWatch e RUM funzionano con Application Signals per offrire vantaggi quando si utilizzano queste funzionalità insieme.

Linguaggi e architetture supportati

Attualmente, Application Signals supporta applicazioni Java e Python.

Application Signals è supportato e testato su Amazon EKS, Amazon ECS e Amazon EC2. Sui cluster Amazon EKS, rileva automaticamente i nomi dei tuoi servizi e cluster. Su altre architetture, devi fornire i nomi dei servizi e degli ambienti quando abiliti tali servizi per Application Signals.

Le istruzioni per abilitare Application Signals su Amazon EC2 dovrebbero funzionare su qualsiasi architettura che supporti l' CloudWatch agente e AWS Distro for. OpenTelemetry Tuttavia, le istruzioni non sono state testate su architetture diverse da Amazon ECS e Amazon EC2.

Regioni supportate

In questa versione di anteprima, Application Signals è supportato nelle seguenti regioni.

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- US West (Oregon)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Europa (Irlanda)

Anteprima SDK

Una versione di anteprima dell'SDK è disponibile per il download.

> **Warning**
>
> Le operazioni e i parametri delle API sono soggetti a modifiche prima che Application Signals sia disponibile a livello generale. Potrebbe trattarsi di modifiche importanti. Non utilizzare la versione di anteprima dell'SDK per scopi di produzione.

Per installare Preview SDK, installa o aggiorna prima l'ultima versione della AWS CLI versione 2. Per ulteriori informazioni, consulta Installazione o aggiornamento della versione più recente della AWS CLI.

Quindi utilizza i seguenti comandi per scaricare il file zip SDK dal bucket Amazon S3 e quindi estrarne il contenuto. Ogni file zip SDK contiene istruzioni SDK e documentazione API.

> **Note**
>
> L'SDK è fornito in più linguaggi di programmazione in modo da poter utilizzare le API di Application Signals con uno qualsiasi di questi linguaggi di programmazione. Tuttavia,

la strumentazione automatica dell'applicazione per inviare dati ad Application Signals è supportata solo per le applicazioni Java e Python.

- SDK Java V2: `aws s3 cp s3://application-signals-preview-sdk/awsJavaSdkV2.zip ./`
- JavaScript SDK V3: `aws s3 cp s3://application-signals-preview-sdk/jsSdkV3.zip ./`
- JavaScript SDK V2: `aws s3 cp s3://application-signals-preview-sdk/jsSdkV2.zip ./`
- SDK Python: `aws s3 cp s3://application-signals-preview-sdk/pythonSdk.zip ./`
- SDK Kotlin: `aws s3 cp s3://application-signals-preview-sdk/kotlin.zip ./`
- SDK Android: `aws s3 cp s3://application-signals-preview-sdk/andriod.zip ./`
- SDK C++: `aws s3 cp s3://application-signals-preview-sdk/awsCppSdk.zip ./`
- SDK PHP: `aws s3 cp s3://application-signals-preview-sdk/awsSdkPhp.zip ./`
- SDK Ruby: `aws s3 cp s3://application-signals-preview-sdk/awsSdkRuby.zip ./`
- SDK Go V2: `aws s3 cp s3://application-signals-preview-sdk/awsSdkGoV2.zip ./`
- SDK Go V1: `aws s3 cp s3://application-signals-preview-sdk/go.zip ./`
- SDK iOS: `aws s3 cp s3://application-signals-preview-sdk/iOS.zip ./`

Argomenti

Autorizzazioni necessarie per Application Signals

⚠ Application Signals è in versione di anteprima per Amazon CloudWatch ed è soggetta a modifiche.

Questa sezione illustra le autorizzazioni necessarie per abilitare, gestire e utilizzare Application Signals.

Autorizzazioni per abilitare e gestire Application Signals

Per gestire Application Signals, devi aver effettuato l'accesso con le seguenti autorizzazioni:

```
{
 "Version": "2012-10-17",
 "Statement": [
    {
        "Sid": "CloudWatchApplicationSignalsFullAccessPermissions",
        "Effect": "Allow",
        "Action": "application-signals:*",
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsAlarmsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsMetricsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:ListMetrics"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
```

```
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSyntheticsPermissions",
        "Effect": "Allow",
        "Action": [
            "synthetics:DescribeCanaries",
            "synthetics:DescribeCanariesLastRun",
            "synthetics:GetCanaryRuns"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsRumPermissions",
        "Effect": "Allow",
        "Action": [
            "rum:BatchCreateRumMetricDefinitions",
            "rum:BatchDeleteRumMetricDefinitions",
            "rum:BatchGetRumMetricDefinitions",
            "rum:GetAppMonitor",
            "rum:GetAppMonitorData",
            "rum:ListAppMonitors",
            "rum:PutRumMetricsDestination",
            "rum:UpdateRumMetricDefinition"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsXrayPermissions",
        "Effect": "Allow",
```

```
        "Action": [
            "xray:GetTraceSummaries"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsPutMetricAlarmPermissions",
        "Effect": "Allow",
        "Action": "cloudwatch:PutMetricAlarm",
        "Resource": [
            "arn:aws:cloudwatch:*:*:alarm:SLO-AttainmentGoalAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLO-WarningAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLI-HealthAlarm-*"
        ]
    },
    {
        "Sid": "CloudWatchApplicationSignalsCreateServiceLinkedRolePermissions",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
        "Condition": {
            "StringLike": {
                "iam:AWSServiceName": "application-signals.cloudwatch.amazonaws.com"
            }
        }
    },
    {
        "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
        "Effect": "Allow",
        "Action": "iam:GetRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSnsWritePermissions",
        "Effect": "Allow",
        "Action": [
            "sns:CreateTopic",
            "sns:Subscribe"
        ],
        "Resource": "arn:aws:sns:*:*:cloudwatch-application-signals-*"
    },
    {
```

```
        "Sid": "CloudWatchApplicationSignalsSnsReadPermissions",
        "Effect": "Allow",
        "Action": "sns:ListTopics",
        "Resource": "*"
    }
 ]
}
```

Per abilitare Application Signals su Amazon EC2 o architetture personalizzateKubernetes, consulta Abilitare Application Signals su altre piattaforme con una configurazione personalizzata. Per abilitare e gestire Application Signals su Amazon EKS utilizzando il componente aggiuntivo Amazon CloudWatch Observability EKS, sono necessarie le seguenti autorizzazioni.

> **Important**
>
> Queste autorizzazioni includono `iam:PassRole` con `Resource "*"` e `eks:CreateAddon` con `Resource "*"`. Si tratta di autorizzazioni avanzate e dovresti concederle con cautela.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
"Sid": "CloudWatchApplicationSignalsEksAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:AccessKubernetesApi",
                "eks:CreateAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonConfiguration",
                "eks:DescribeAddonVersions",
                "eks:DescribeCluster",
                "eks:DescribeUpdate",
                "eks:ListAddons",
                "eks:ListClusters",
                "eks:ListUpdates",
                "iam:ListRoles",
                "iam:PassRole"
            ],
            "Resource": "*"
        },
        {
```

```
"Sid":
 "CloudWatchApplicationSignalsEksCloudWatchObservabilityAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DeleteAddon",
                "eks:UpdateAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
        }
    ]
}
```

La dashboard di Application Signals mostra le AWS Service Catalog AppRegistry applicazioni a cui sono associati i tuoi SLO. Per visualizzare queste applicazioni nelle pagine SLO, è necessario disporre delle seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Utilizzo di Application Signals

Gli operatori di servizi che utilizzano Application Signals per monitorare servizi e SLO devono accedere a un account con le seguenti autorizzazioni di sola lettura:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-signals:BatchGet*",
                "application-signals:Get*",
```

```
                "application-signals:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
        },
        {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
        {
            "Sid": "CloudWatchApplicationSignalsAlarmsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsMetricsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsSyntheticsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:DescribeCanariesLastRun",
                "synthetics:GetCanaryRuns"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsRumReadPermissions",
            "Effect": "Allow",
            "Action": [
                "rum:BatchGetRumMetricDefinitions",
                "rum:GetAppMonitor",
                "rum:GetAppMonitorData",
                "rum:ListAppMonitors"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsXrayReadPermissions",
            "Effect": "Allow",
            "Action": [
                "xray:GetTraceSummaries"
            ],
            "Resource": "*"
        }
    ]
}
```

Per vedere a quali AWS Service Catalog AppRegistry applicazioni sono associate le SLO nella dashboard di Application Signals, devi disporre delle seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
```

```
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Per verificare se Application Signals su Amazon EKS che utilizza il componente aggiuntivo Amazon CloudWatch Observability EKS è abilitato, devi disporre delle seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsEksReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:ListAddons",
                "eks:ListClusters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsEksDescribeAddonReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DescribeAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
        }
    ]
}
```

Abilitazione di Application Signals

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Gli argomenti di questa sezione spiegano come abilitare CloudWatch Application Signals nel tuo ambiente. Application Signals è supportato sui cluster Amazon EKS con un flusso di lavoro di

configurazione tramite la console. È supportato anche su altre piattaforme, tra cui Amazon EC2, con un processo di configurazione personalizzato.

Argomenti

Sistemi supportati da Application Signals

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Application Signals è supportato e testato su Amazon EKS, Amazon ECS e Amazon EC2. Le istruzioni per abilitare Application Signals su Amazon EC2 dovrebbero funzionare su qualsiasi piattaforma che supporti l' CloudWatch agente e AWS Distro for OpenTelemetry, ma non sono state testate su altre piattaforme.

Compatibilità con Java

Application Signals supporta le applicazioni Java e supporta le stesse librerie e framework Java di Distro for. AWS OpenTelemetry Per ulteriori informazioni, consulta Librerie, framework, server di applicazioni e JVM supportati.

Sono supportate le versioni JVM 8, 11 e 17.

Compatibilità con Python

Application Signals supporta le stesse librerie e framework di AWS Distro for. OpenTelemetry Per ulteriori informazioni, consulta Pacchetti supportati all'indirizzo. opentelemetry-python-contrib

Sono supportate le versioni di Python 3.8 e successive.

Prima di abilitare Application Signals per le tue applicazioni Python, tieni presente le seguenti considerazioni.

- In alcune applicazioni containerizzate, una variabile di `PYTHONPATH` ambiente mancante a volte può impedire l'avvio dell'applicazione. Per risolvere questo problema, assicuratevi di impostare la variabile di `PYTHONPATH` ambiente sulla posizione della directory di lavoro dell'applicazione. Ciò è dovuto a un problema noto con la OpenTelemetry strumentazione automatica. Per ulteriori informazioni su questo problema, vedere L'impostazione della strumentazione automatica in Python di PYTHONPATH non è conforme.
- Per le applicazioni Django, ci sono configurazioni aggiuntive richieste, che sono descritte nella documentazione di Python OpenTelemetry .
  - Usa il `--noreload` flag per impedire il ricaricamento automatico.
  - Imposta la variabile di `DJANGO_SETTINGS_MODULE` ambiente sulla posizione del file dell'`settings.py`applicazione Django. Ciò garantisce che OpenTelemetry possa accedere e integrarsi correttamente con le impostazioni di Django.

OpenTelemetry considerazioni sulla compatibilità

> ⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Per integrare le tue applicazioni con CloudWatch Application Signals, ti consigliamo di rimuovere prima completamente qualsiasi soluzione di monitoraggio delle prestazioni delle applicazioni esistente dall'applicazione. Questo include la rimozione di qualsiasi codice e configurazione della strumentazione.

Anche se Application Signals utilizza la OpenTelemetry strumentazione, non è garantito che sia compatibile con la strumentazione o la configurazione esistente. OpenTelemetry Nella migliore delle ipotesi, potreste riuscire a mantenere alcune delle vostre OpenTelemetry funzionalità, come le metriche personalizzate. Tuttavia, assicurati di leggere le sezioni seguenti per ulteriori informazioni.

Considerazioni se lo utilizzi già OpenTelemetry

Se la utilizzi già OpenTelemetry con la tua applicazione, il resto di questa sezione contiene informazioni importanti per garantire la compatibilità con Application Signals.

- Prima di abilitare l'applicazione per Application Signals, è necessario rimuovere l'iniezione di qualsiasi altro agente di strumentazione automatica basato su OpenTelemetry dall'applicazione. Questo aiuta a evitare conflitti di configurazione. È possibile continuare a utilizzare la strumentazione manuale utilizzando OpenTelemetry API compatibili insieme ad Application Signals.
- Se utilizzi la strumentazione manuale per generare intervalli o parametri personalizzati dall'applicazione, a seconda della complessità della strumentazione, l'abilitazione di Application Signals potrebbe interrompere la generazione di dati o causare altri comportamenti indesiderati. Potresti essere in grado di utilizzare alcune delle configurazioni disponibili in OpenTelemetry (ad eccezione di quelle menzionate nella tabella riportata più avanti in questa sezione) per mantenere il comportamento desiderato delle metriche o degli intervalli esistenti. Per ulteriori informazioni su queste configurazioni, consulta Configurazione SDK nella documentazione. OpenTelemetry

  Ad esempio, utilizzando la `OTEL_EXPORTER_OTLP_METRICS_ENDPOINT` configurazione e un'istanza OpenTelemetry Collector autogestita, potresti essere in grado di continuare a inviare le metriche personalizzate alla destinazione desiderata.
- Alcune variabili di ambiente o proprietà di sistema non devono essere utilizzate con Application Signals, mentre è possibile utilizzarne altre purché si seguano le indicazioni riportate nella tabella. Per i dettagli, consulta la seguente tabella.

Variabile di ambiente	Raccomandazione con Application Signals
Variabili di ambiente generali	
`OTEL_SDK_DISABLED`	Deve essere impostato su `true`.
`OTEL_TRACES_EXPORTER`	Deve essere impostato su `otlp`.
`OTEL_EXPORTER_OTLP_ENDPOINT`	Non deve essere usato.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Non deve essere usato.
`OTEL_ATTRIBUTE_COUNT_LIMIT`	Se impostato, deve essere impostato a un valore sufficientemente alto da includere circa altri 10 attributi span aggiunti da CloudWatch Application Signals.

Variabile di ambiente	Raccomandazione con Application Signals
`OTEL_PROPAGATORS`	Se impostato, deve essere incluso `xray` per il tracciamento finale.
`OTEL_TRACES_SAMPLER`	Se impostato, deve essere `xray` a utilizzare il campionamento centralizzato di X-Ray. Per utilizzare il campionamento locale, impostalo su `parentbased_tracei dratio` e specifica la frequenza di campionamento in `OTEL_TRACES_SAMPLE R_ARG` .
`OTEL_TRACES_SAMPLER_ARG`	Se utilizzi l'impostazione predefinita del campione di traccia centralizzato di X-Ray, questa variabile non deve essere utilizzata. Se invece utilizzi il campionamento locale, imposta la frequenza di campionamento in questa variabile. Ad esempio, `0.05` per una frequenza di campionamento del 5%.
Variabili di ambiente specifiche per Java	
`OTEL_JAVA_ENABLED_RESOURCE_ PROVIDERS`	Se impostato, deve includere rilevatori di AWS risorse.
Variabili di ambiente specifiche per Python	
`OTEL_PYTHON_CONFIGURATOR`	Se usato, deve essere impostato su `aws_configurator`
`OTEL_PYTHON_DISTRO`	Se usato, deve essere impostato su `aws_distro`

Abilita Application Signals sui cluster Amazon EKS

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

CloudWatch Application Signals è supportato per le applicazioni Java e Python in esecuzione nei cluster Amazon EKS. Per abilitare Application Signals per applicazioni in un cluster Amazon EKS, hai due opzioni:

- Per abilitare Application Signals per le applicazioni su un cluster Amazon EKS esistente, utilizza la procedura riportata in Abilitazione di Application Signals su un cluster Amazon EKS con i tuoi servizi.
- Per provare Application Signals in un ambiente non di produzione con un'applicazione di esempio, usa le istruzioni contenute in Abilitazione di Application Signals su un nuovo cluster Amazon EKS con un'app di esempio. Questo flusso di lavoro utilizza gli script forniti da AWS per creare un nuovo cluster Amazon EKS e installare un'applicazione di esempio abilitata per Application Signals. Ciò consente di visualizzare e testare la end-to-end funzionalità di Application Signals.

Argomenti

Abilitazione di Application Signals su un cluster Amazon EKS con i tuoi servizi

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Per abilitare CloudWatch Application Signals sulle tue applicazioni su un cluster Amazon EKS esistente, utilizza le istruzioni in questa sezione.

> **Important**
>
> Se stai già utilizzando OpenTelemetry un'applicazione che intendi abilitare per Application Signals, consulta OpenTelemetry considerazioni sulla compatibilità prima di abilitare Application Signals.

Per abilitare Application Signals per le applicazioni su un cluster Amazon EKS esistente

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Servizi.
3. Se non hai ancora abilitato Application Signals in questo account, devi concedere ad Application Signals le autorizzazioni necessarie per scoprire i tuoi servizi. Per far ciò, completa le seguenti operazioni. È necessario eseguire questa operazione solo una volta per account.

   a. Scegli Inizia a scoprire i tuoi servizi.

   b. Seleziona la casella di controllo e scegli Inizia a scoprire i servizi.

      Il completamento di questo passaggio per la prima volta nel tuo account crea il ruolo AWSServiceRoleForCloudWatchApplicationSignalscollegato al servizio. Questo ruolo concede ad Application Signals le seguenti autorizzazioni:

      - `xray:GetServiceGraph`
      - `logs:StartQuery`
      - `logs:GetQueryResults`
      - `cloudwatch:GetMetricData`
      - `cloudwatch:ListMetrics`
      - `tag:GetResources`

      Per ulteriori informazioni su questo ruolo, consulta Autorizzazioni di ruolo collegate al servizio per Application Signals CloudWatch .
4. Scegli Abilita Application Signals.
5. Per Specifica piattaforma, scegli EKS.
6. Per Seleziona un cluster EKS, scegli il cluster in cui desideri abilitare Application Signals.

7. Se questo cluster non ha già il componente aggiuntivo Amazon CloudWatch Observability EKS abilitato, ti verrà richiesto di abilitarlo. In questo caso, puoi fare quanto segue:

   a. Scegli il componente aggiuntivo Add CloudWatch Observability EKS. Viene visualizzata la console Amazon EKS.

   b. Seleziona la casella di controllo per Amazon CloudWatch Observability e scegli Avanti.

      Il componente aggiuntivo CloudWatch Observability EKS abilita sia Application Signals che CloudWatch Container Insights con una migliore osservabilità per Amazon EKS. Per ulteriori informazioni su Container Insights, consulta Container Insights.

   c. Seleziona la versione più recente del componente aggiuntivo da installare.

   d. Seleziona un ruolo IAM da utilizzare per il componente aggiuntivo. Se scegli Eredita dal nodo, assegna le autorizzazioni corrette per il ruolo IAM utilizzato dai tuoi nodi worker. *my-worker-node-role*Sostituiscilo con il ruolo IAM utilizzato dai nodi di lavoro Kubernetes.

      ```
      aws iam attach-role-policy \
      --role-name my-worker-node-role \
      --policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
      --policy-arn arn:aws:iam::aws:policy/AWSXRayWriteOnlyAccess
      ```

   e. Per creare un ruolo di servizio per l'utilizzo del componente aggiuntivo, consulta Installa l' CloudWatch agente utilizzando il componente aggiuntivo Amazon CloudWatch Observability EKS.

   f. Scegli Successivo, conferma le informazioni sullo schermo e scegli Crea.

   g. Nella schermata successiva, scegli Enable CloudWatch Application Signals per tornare alla CloudWatch console e completare il processo.

8. Esistono due opzioni per abilitare le applicazioni per Application Signals. Per motivi di coerenza, consigliamo di scegliere un'opzione per cluster.

   - L'opzione Console è più semplice. L'utilizzo di questo metodo fa sì che i pod si riavviino immediatamente.
   - Il metodo Annotate Manifest File ti offre un maggiore controllo sul momento in cui i pod si riavviano e può anche aiutarti a gestire il monitoraggio in modo più decentralizzato se non desideri centralizzarlo.

Console

L'opzione Console utilizza la configurazione avanzata del componente aggiuntivo Amazon CloudWatch Observability EKS per configurare Application Signals per i tuoi servizi. Per ulteriori informazioni sul componente aggiuntivo, consulta. (Facoltativo) Configurazione aggiuntiva

Se non vedi un elenco di carichi di lavoro e namespace, assicurati di disporre delle autorizzazioni giuste per visualizzarli per questo cluster. Per ulteriori informazioni, consulta Autorizzazioni richieste.

Puoi monitorare singoli carichi di lavoro o interi namespace.

Per monitorare un singolo carico di lavoro:

1. Seleziona la casella di controllo in base al carico di lavoro che desideri monitorare.
2. Seleziona la lingua del carico di lavoro. Per le applicazioni Python, assicurati che l'applicazione soddisfi i prerequisiti richiesti prima di continuare. Per ulteriori informazioni, consulta L'applicazione Python non si avvia dopo l'attivazione di Application Signals.
3. Seleziona Fatto. Il componente aggiuntivo Amazon CloudWatch Observability EKS inietterà immediatamente gli SDK AWS Distro for OpenTelemetry autoinstrumentation (ADOT) nei tuoi pod e attiverà il riavvio del pod per consentire la raccolta di metriche e tracce dell'applicazione.

Per monitorare un intero namespace:

1. Seleziona la casella di controllo accanto allo spazio dei nomi che desideri monitorare.
2. Seleziona la lingua del carico di lavoro. Ciò si applica a tutti i carichi di lavoro in questo namespace, indipendentemente dal fatto che siano attualmente distribuiti o che verranno distribuiti in futuro. Per le applicazioni Python, assicurati che l'applicazione soddisfi i prerequisiti richiesti prima di continuare. Per ulteriori informazioni, consulta L'applicazione Python non si avvia dopo l'attivazione di Application Signals.
3. Seleziona Fatto. Il componente aggiuntivo Amazon CloudWatch Observability EKS inietterà immediatamente gli SDK AWS Distro for OpenTelemetry autoinstrumentation (ADOT) nei tuoi pod e attiverà il riavvio del pod per consentire la raccolta di metriche e tracce dell'applicazione.

Per abilitare Application Signals in un altro cluster Amazon EKS, scegli Abilita Application Signals dalla schermata Servizi.

Annotate manifest file

Nella CloudWatch console, la sezione Monitor Services spiega che è necessario aggiungere un'annotazione a un file YAML manifesto nel cluster. L'aggiunta di questa annotazione strumenta automaticamente l'applicazione per inviare parametri, tracciamenti e log ad Application Signals.

Sono disponibili due opzioni per l'annotazione:

- Annotazione del carico di lavoro strumenta automaticamente un singolo carico di lavoro nel cluster.
- Annotazione dello spazio dei nomi strumenta automaticamente tutti i carichi di lavoro distribuiti nello spazio dei nomi selezionato.

Scegli una di queste opzioni e segui i passaggi appropriati:

- Per annotare un singolo carico di lavoro:
  1. Scegli Annotazione del carico di lavoro.
  2. Incolla una delle seguenti righe nella `PodTemplate` sezione del file manifesto del carico di lavoro.
     - Per i carichi di lavoro Java: `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
     - Per i carichi di lavoro Python: `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

       Per le applicazioni Python, sono necessarie configurazioni aggiuntive. Per ulteriori informazioni, consulta L'applicazione Python non si avvia dopo l'attivazione di Application Signals.
  3. Nel tuo terminale, inserisci `kubectl apply -f` *`your_deployment_yaml`* per applicare la modifica.
- Per annotare tutti i carichi di lavoro in un namespace:
  1. Scegli Annotazione dello spazio dei nomi.

2. Incolla una delle seguenti righe nella sezione dei metadati del file manifest dello spazio dei nomi. Se lo spazio dei nomi include sia carichi di lavoro Java che Python, incolla entrambe queste righe nel file manifest dello spazio dei nomi.
   - Se nel namespace sono presenti carichi di lavoro Java: `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
   - Se ci sono carichi di lavoro Python nello spazio dei nomi: `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

     Per le applicazioni Python, sono necessarie configurazioni aggiuntive. Per ulteriori informazioni, consulta L'applicazione Python non si avvia dopo l'attivazione di Application Signals.
3. Nel tuo terminale, inserisci `kubectl apply -f` *`your_namespace_yaml`* per applicare la modifica.
4. Nel tuo terminale, inserisci un comando per riavviare tutti i pod nello spazio dei nomi. Un comando di esempio per riavviare i carichi di lavoro di implementazione è `kubectl rollout restart deployment -n` *`namespace_name`*

9. Scegli Al termine visualizza servizi. Accederai alla visualizzazione dei servizi di Application Signals, dove è possibile visualizzare i dati raccolti da Application Signals. Potrebbero essere necessari alcuni minuti prima che i dati vengano visualizzati.

   Per abilitare Application Signals in un altro cluster Amazon EKS, scegli Abilita Application Signals dalla schermata Servizi.

   Per ulteriori informazioni sulla visualizzazione Servizi, consulta Monitoraggio dell'integrità operativa delle applicazioni con Application Signals.

> **Note**
>
> Abbiamo identificato alcune considerazioni da tenere a mente quando si abilitano le applicazioni Python per Application Signals. Per ulteriori informazioni, consulta L'applicazione Python non si avvia dopo l'attivazione di Application Signals.

Abilitazione di Application Signals su un nuovo cluster Amazon EKS con un'app di esempio

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Per provare CloudWatch Application Signals su un'app di esempio prima di utilizzarla per utilizzare le proprie applicazioni, segui le istruzioni in questa sezione. Queste istruzioni utilizzano script per aiutarti a creare un cluster Amazon EKS, installare un'applicazione di esempio e strumentare l'applicazione di esempio affinché funzioni con Application Signals.

L'applicazione di esempio è un'applicazione Spring “Pet Clinic” composta da quattro microservizi. Questi servizi vengono eseguiti su Amazon EKS su Amazon EC2 e sfruttano gli script di abilitazione di Application Signals per abilitare il cluster con l'agente di strumentazione automatica Java o Python.

Requisiti

- Attualmente, Application Signals monitora solo le applicazioni Java e Python.
- È necessario che AWS CLI sia installato sull'istanza. Consigliamo AWS CLI la versione 2, ma dovrebbe funzionare anche la versione 1. Per ulteriori informazioni sull'installazione di AWS CLI, consulta Installare o aggiornare la versione più recente di AWS CLI.
- Gli script in questa sezione sono pensati per essere eseguiti in ambienti Linux e macOS. Per le istanze Windows, si consiglia di utilizzare un AWS Cloud9 ambiente per eseguire questi script. Per ulteriori informazioni su AWS Cloud9, vedi Cos'è? AWS Cloud9 .
- Installa una versione supportata di `kubectl`. Devi utilizzare la versione `kubectl` la cui differenza di versione secondaria è uno rispetto al piano di controllo del cluster Amazon EKS. Ad esempio, un client `kubectl` 1.26 deve funzionare con cluster Kubernetes 1.25, 1.26 e 1.27. Se disponi già di un cluster Amazon EKS, potrebbe essere necessario configurare AWS le credenziali per`kubectl`. Per ulteriori informazioni, consulta Creazione o aggiornamento di un file `kubeconfig` per un cluster Amazon EKS.
- Installa`eksctl`. `eksctl`utilizza il AWS CLI per interagire AWS, il che significa che utilizza AWS le stesse credenziali di AWS CLI. Per ulteriori informazioni, consulta Installazione o aggiornamento di `eksctl`.
- Installa`jq`. `jq` è necessario per eseguire gli script di abilitazione di Application Signals. Per ulteriori informazioni, consulta Scarica jq.

Fase 1: download degli script

Per scaricare gli script per configurare CloudWatch Application Signals con un'app di esempio, puoi scaricare e decomprimere il file di GitHub progetto compresso su un'unità locale oppure clonare il progetto. GitHub

Per clonare il progetto, apri una finestra terminale e inserisci il seguente comando Git in una determinata directory di lavoro.

```
git clone https://github.com/aws-observability/application-signals-demo.git
```

Fase 2: creazione ed esecuzione dell'applicazione di esempio

Per creare e inviare le immagini di esempio dell'applicazione, segui queste istruzioni.

Fase 3: implementazione e abilitazione di Application Signals e dell'applicazione di esempio

Assicurati di aver completato i requisiti indicati in Abilitazione di Application Signals su un nuovo cluster Amazon EKS con un'app di esempio prima di completare i seguenti passaggi.

Per implementare e abilitare Application Signals e l'applicazione di esempio

1. Immetti il seguente comando nel terminale locale in cui è stato decompresso lo script di onboarding. *new-cluster-name*Sostituiscilo con il nome che desideri utilizzare per il nuovo cluster. Sostituisci *region-name* con il nome della AWS regione, ad esempio. `us-west-1`

   Questo comando configura l'app di esempio in esecuzione in un nuovo cluster Amazon EKS con Application Signals abilitato.

   ```
   # assuming the current working directory is 'onboarding'
   # this script sets up a new cluster, enables Application Signals, and deploys the
   # sample application
   cd application-signals-demo/scripts/eks/appsignals/one-step && ./setup.sh new-
   cluster-name region-name
   ```

   L'esecuzione dello script di configurazione richiede circa 30 minuti ed esegue le seguenti operazioni:

   - Crea un nuovo cluster Amazon EKS nella regione specificata.

- Crea le autorizzazioni IAM necessarie per Application Signals (`arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess` e `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy`).
- Abilita Application Signals installando l' CloudWatch agente e autostrumentando l'applicazione di esempio per CloudWatch metriche e tracce X-Ray.
- Distribuisce l'applicazione di esempio PetClinic Spring nello stesso cluster Amazon EKS.
- Crea cinque canarini CloudWatch Synthetics, `pc-add-vist` `pc-create-owners` denominati,,,,. `pc-visit-pet` `pc-visit-vet` `pc-clinic-traffic` Questi canary funzioneranno a una frequenza di un minuto per generare traffico sintetico per l'app di esempio e dimostrare come vengono visualizzati i canary Synthetics in Application Signals.
- Crea quattro obiettivi di livello di servizio (SLO) per l' PetClinic applicazione con i seguenti nomi:
  - Disponibilità per Searching an Owner
  - Latenza per Searching an Owner
  - Disponibilità per Registering an Owner
  - Latenza per Registering an Owner
- Crea il ruolo IAM richiesto con una policy di attendibilità personalizzata che concede ad Application Signals le seguenti autorizzazioni:
  - `cloudwatch:PutMetricData`
  - `cloudwatch:GetMetricData`
  - `xray:GetServiceGraph`
  - `logs:StartQuery`
  - `logs:GetQueryResults`

2. (Facoltativo) Se desideri esaminare il codice sorgente dell'applicazione di PetClinic esempio, puoi trovarlo nella cartella principale.

```
- application-signals-demo
  - spring-petclinic-admin-server
  - spring-petclinic-api-gateway
  - spring-petclinic-config-server
  - spring-petclinic-customers-service
  - spring-petclinic-discovery-server
  - spring-petclinic-vets-service
  - spring-petclinic-visits-service
```

3. Per visualizzare l'applicazione di PetClinic esempio distribuita, esegui il comando seguente per trovare l'URL:

```
kubectl get ingress
```

Fase 4: distribuzione dell'applicazione di esempio

Dopo aver completato i passaggi nella sezione precedente per creare il cluster Amazon EKS e distribuire l'applicazione di esempio, puoi utilizzare Application Signals per monitorare l'applicazione.

> **Note**
>
> Affinché la console Application Signals inizi a popolarsi, una parte del traffico deve raggiungere l'applicazione di esempio. Parte dei passaggi precedenti ha creato i canari CloudWatch Synthetics che generano traffico verso l'applicazione di esempio.

Monitoraggio dei servizi

Dopo l'attivazione, CloudWatch Application Signals rileva e compila automaticamente un elenco di servizi senza richiedere alcuna configurazione aggiuntiva.

Per visualizzare l'elenco dei servizi rilevati e monitorarne lo stato

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Servizi.
3. Per visualizzare un servizio, le sue operazioni e le sue dipendenze, scegli il nome di uno dei servizi nell'elenco.

   Questa visualizzazione unificata e incentrata sull'applicazione aiuta a fornire una prospettiva completa del modo in cui gli utenti interagiscono con il servizio. Questo può aiutarti a diagnosticare i problemi in caso di anomalie nelle prestazioni. Per i dettagli completi sulla visualizzazione dei Servizi, consulta Monitoraggio dell'integrità operativa delle applicazioni con Application Signals.
4. Scegli la scheda Operazioni di servizio per visualizzare i parametri standard delle applicazioni per le operazioni di quel servizio. Le operazioni sono le operazioni API chiamate dal servizio, ad esempio.

Quindi, per visualizzare i grafici per una singola operazione di quel servizio, scegli il nome dell'operazione.

5. Scegli la scheda Dipendenze per visualizzare le dipendenze dell'applicazione, insieme ai parametri critici delle applicazioni per ciascuna dipendenza. Le dipendenze includono AWS servizi e servizi di terze parti richiamati dall'applicazione.
6. Per visualizzare le tracce correlate dalla pagina dei dettagli del servizio, scegli un punto dati in uno dei tre grafici sopra la tabella. Questa operazione popola un nuovo riquadro con le tracce filtrate del periodo di tempo. Queste tracce vengono ordinate e filtrate in base al grafico scelto. Ad esempio, se hai scelto il grafico della latenza, le tracce vengono ordinate in base al tempo di risposta del servizio.
7. Nel riquadro di navigazione CloudWatch della console, scegli SLO. Vengono visualizzati gli SLO creati dallo script per l'applicazione di esempio. Per ulteriori informazioni sugli SLO, consulta Obiettivi del livello di servizio (SLO).

(Facoltativo) Fase 5: eliminazione

Quando hai finito di testare Application Signals, puoi utilizzare uno script fornito da Amazon per ripulire ed eliminare gli artefatti creati nel tuo account per l'applicazione di esempio. Per eseguire la eliminazione, immetti il seguente comando. Sostituisci *new-cluster-name* con il nome del cluster che hai creato per l'app di esempio e sostituisci *region* -name con il nome della AWS regione, ad esempio. `us-west-1`

```
cd application-signals-demo/scripts/eks/appsignals/one-step && ./cleanup.sh new-
cluster-name region-name
```

Abilita Application Signals su altre piattaforme con una configurazione personalizzata

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Abilita CloudWatch Application Signals su piattaforme diverse da Amazon EKS utilizzando i passaggi di configurazione personalizzati descritti in queste sezioni. Su queste architetture, puoi installare e configurare autonomamente l' CloudWatch agente e AWS Distro. OpenTelemetry

Su queste architetture, Application Signals non rileva automaticamente i nomi dei tuoi servizi o dei relativi cluster o host. Devi specificare questi nomi durante la configurazione personalizzata e i nomi specificati sono quelli visualizzati nei pannelli di controllo di Application Signals.

Argomenti

Usa una configurazione personalizzata per abilitare Application Signals su Amazon ECS

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback @amazon .com.

Utilizza queste istruzioni di configurazione personalizzate per effettuare l'onboarding delle tue applicazioni su Amazon ECS su CloudWatch Application Signals. Installi e configuri tu stesso l' CloudWatch agente e AWS Distro. OpenTelemetry

Nei cluster Amazon ECS, Application Signals non rileva automaticamente i nomi dei tuoi servizi o i cluster in cui vengono eseguiti. Devi specificare questi nomi durante la configurazione personalizzata e i nomi specificati sono quelli visualizzati nei pannelli di controllo di Application Signals.

⚠ Important

Solo la modalità di rete `awsvpc` è supportata.

Fase 1: abilitazione di Application Signals nel tuo account

Se non hai ancora abilitato Application Signals in questo account, devi concedere ad Application Signals le autorizzazioni necessarie per scoprire i tuoi servizi. Per far ciò, completa le seguenti operazioni. È necessario eseguire questa operazione solo una volta per account.

Per abilitare Application Signals per le tue applicazioni

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione, scegli Servizi.
3. Scegli Inizia a scoprire i tuoi servizi.
4. Seleziona la casella di controllo e scegli Inizia a scoprire i servizi.

   Il completamento di questo passaggio per la prima volta nel tuo account crea il ruolo AWSServiceRoleForCloudWatchApplicationSignalscollegato al servizio. Questo ruolo concede ad Application Signals le seguenti autorizzazioni:

   - `xray:GetServiceGraph`
   - `logs:StartQuery`
   - `logs:GetQueryResults`
   - `cloudwatch:GetMetricData`
   - `cloudwatch:ListMetrics`
   - `tag:GetResources`

   Per ulteriori informazioni su questo ruolo, consulta Autorizzazioni di ruolo collegate al servizio per Application Signals CloudWatch .

Fase 2: creazione dei ruoli IAM

È necessario creare due ruoli IAM. Se questi ruoli sono già stati creati, potrebbe essere necessario aggiungervi delle autorizzazioni.

- Ruolo dell'attività ECS: i container utilizzano questo ruolo per l'esecuzione. Le autorizzazioni devono corrispondere a ciò di cui hanno bisogno le applicazioni, in più. CloudWatchAgentServerPolicyAWSXRayWriteOnlyAccess
- Ruolo di esecuzione dell'attività ECS: Amazon ECS utilizza questo ruolo per avviare ed eseguire i tuoi container. Se hai già creato questo ruolo, allega AmazonSSM, ReadOnlyAccess AmazonECS e le politiche. TaskExecutionRolePolicy CloudWatchAgentServerPolicy

  Se devi archiviare dati più riservati utilizzabili da Amazon ECS, consulta Specifica dei dati sensibili per ulteriori informazioni.

Per ulteriori informazioni sulla creazione dei ruoli IAM, consulta Creazione di ruoli IAM.

Fase 3: Preparare la configurazione dell'agente CloudWatch

Innanzitutto, prepara la configurazione dell'agente con Application Signals abilitato. A tale scopo, crea un file locale denominato `/tmp/ecs-cwagent.json`.

```
{
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
  },
  "logs": {
    "metrics_collected": {
      "app_signals": {}
    }
  }
}
```

Quindi carica questa configurazione nell'archivio parametri SSM. A questo scopo, immetti il comando seguente. Nel file, sostituisci *$REGION* con il nome effettivo della tua regione.

```
aws ssm put-parameter \
--name "ecs-cwagent" \
--type "String" \
--value "`cat /tmp/ecs-cwagent.json`" \
--region "$REGION"
```

Fase 4: Strumentate la vostra applicazione con l' CloudWatch agente

Il passo successivo è quello di strumentare la vostra CloudWatch applicazione per Application Signals.

Java

Per strumentare la tua applicazione su Amazon ECS con l'agente CloudWatch

1. Innanzitutto, specifica un montaggio vincolato. Il volume verrà utilizzato per condividere file tra container nei passaggi successivi. Dovrai utilizzare questo montaggio vincolato più avanti in questa procedura.

   ```
   "volumes": [
   ```

```
  {
    "name": "opentelemetry-auto-instrumentation"
  }
]
```

2. Aggiungi una definizione collaterale CloudWatch dell'agente. A tale scopo, aggiungi un nuovo container chiamato `ecs-cwagent` alla definizione dell'attività dell'applicazione. Sostituisci *`$REGION`* con il nome effettivo della tua regione. Sostituisci con il percorso dell'immagine del CloudWatch contenitore più recente su Amazon Elastic Container Registry. Per maggiori informazioni sull'utilizzo di CloudTrail con Amazon ECR, consulta cloudwatch-agent.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
}
```

3. Aggiungi un nuovo container `init` alla definizione dell'attività dell'applicazione. Sostituisci *`$IMAGE`* con l'immagine più recente dal repository di immagini AWS Distro for OpenTelemetry Amazon ECR.

```
{
  "name": "init",
  "image": "$IMAGE",
  "essential": false,
  "command": [
    "cp",
```

```
    "/javaagent.jar",
    "/otel-auto-instrumentation/javaagent.jar"
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
}
```

4. Aggiungi le seguenti variabili di ambiente al container dell'applicazione. Per ulteriori informazioni, consulta la pagina

Variabile di ambiente	Impostazione per abilitare Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Sostituisci `$SVC_NAME` con il nome della tua applicazione. Questo verrà visualizzato come nome dell'applicazione nei pannelli di controllo di Application Signals. Sostituisci `$HOST_ENV` con l'ambiente host in cui è in esecuzione l'applicazione. Questo verrà visualizzato come ambiente ospitato dell'applicazione nei pannelli di controllo di Application Signals.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Impostato per abilitare gli `true` Application Signals. SpanMetricsProcessor
`OTEL_METRICS_EXPORTER`	Imposta su `none` per disabilitare gli esportatori di altri parametri.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Imposta su `http://127.0.0.1:4315` per inviare le metriche al CloudWatch sidecar.

Variabile di ambiente	Impostazione per abilitare Application Signals
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Imposta su `http://127.0.0.1:4315` per inviare tracce al sidecar. CloudWatch
`OTEL_TRACES_SAMPLER`	Definisci X-Ray come campionatore di tracce.
`OTEL_PROPAGATORS`	Aggiungi X-Ray come uno dei propagatori.
`JAVA_TOOL_OPTIONS`	Inietta l'agente AWS Distro for OpenTelemetry Java.

5. Monta il volume `opentelemetry-auto-instrumentation` definito nella fase 1 di questa procedura.

   Per un'applicazione Java, utilizzare quanto segue.

```
{
  "name": "app",
   ...
  "environment": [
    {
      "name": "OTEL_RESOURCE_ATTRIBUTES",
      "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAME"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
      "value": "true"
    },
    {
      "name": "OTEL_METRICS_EXPORTER",
      "value": "none"
    },
    {
      "name": "JAVA_TOOL_OPTIONS",
      "value": " -javaagent:/otel-auto-instrumentation/javaagent.jar"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
      "value": "http://127.0.0.1:4315"
```

```
    },
    {
      "name": "OTEL_TRACES_SAMPLER",
      "value": "xray"
    },
    {
      "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
    {
      "name": "OTEL_PROPAGATORS",
      "value": "tracecontext,baggage,b3,xray"
    }
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
}
```

Python

Prima di abilitare Application Signals per le tue applicazioni Python, tieni presente le seguenti considerazioni.

- In alcune applicazioni containerizzate, una variabile di `PYTHONPATH` ambiente mancante a volte può impedire l'avvio dell'applicazione. Per risolvere questo problema, assicuratevi di impostare la variabile di `PYTHONPATH` ambiente sulla posizione della directory di lavoro dell'applicazione. Ciò è dovuto a un problema noto con la OpenTelemetry strumentazione automatica. Per ulteriori informazioni su questo problema, vedere L'impostazione della strumentazione automatica in Python di PYTHONPATH non è conforme.
- Per le applicazioni Django, ci sono configurazioni aggiuntive richieste, che sono descritte nella documentazione di Python OpenTelemetry .
  - Usa il `--noreload` flag per impedire il ricaricamento automatico.
  - Imposta la variabile di `DJANGO_SETTINGS_MODULE` ambiente sulla posizione del file dell'`settings.py`applicazione Django. Ciò garantisce che OpenTelemetry possa accedere e integrarsi correttamente con le impostazioni di Django.

Per strumentare la tua applicazione Python su Amazon ECS con l'agente CloudWatch

1. Innanzitutto, specifica un montaggio vincolato. Il volume verrà utilizzato per condividere file tra container nei passaggi successivi. Dovrai utilizzare questo montaggio vincolato più avanti in questa procedura.

```
"volumes": [
  {
    "name": "opentelemetry-auto-instrumentation-python"
  }
]
```

2. Aggiungi una definizione collaterale CloudWatch dell'agente. A tale scopo, aggiungi un nuovo container chiamato `ecs-cwagent` alla definizione dell'attività dell'applicazione. Sostituisci *$REGION* con il nome effettivo della tua regione. Sostituisci con il percorso dell'immagine del CloudWatch contenitore più recente su Amazon Elastic Container Registry. Per maggiori informazioni sull'utilizzo di CloudTrail con Amazon ECR, consulta cloudwatch-agent.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
}
```

3. Aggiungi un nuovo container `init` alla definizione dell'attività dell'applicazione. Sostituisci *$IMAGE* con l'immagine più recente dal repository di immagini AWS Distro for OpenTelemetry Amazon ECR.

```
{
    "name": "init",
    "image": "$IMAGE",
    "essential": false,
    "command": [
        "cp",
        "-a",
        "/autoinstrumentation/.",
        "/otel-auto-instrumentation-python"
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

4. Aggiungi le seguenti variabili di ambiente al container dell'applicazione. Per ulteriori informazioni, consulta la pagina

Variabile di ambiente	Impostazione per abilitare Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Sostituisci `$SVC_NAME` con il nome della tua applicazione. Questo verrà visualizzato come nome dell'applicazione nei pannelli di controllo di Application Signals. Sostituisci `$HOST_ENV` con l'ambiente host in cui è in esecuzione l'applicazione. Questo verrà visualizzato come ambiente ospitato dell'applicazione nei pannelli di controllo di Application Signals.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Impostato per abilitare gli `true` Application Signals. SpanMetricsProcessor

Variabile di ambiente	Impostazione per abilitare Application Signals
`OTEL_METRICS_EXPORTER`	Imposta su `none` per disabilitare gli esportatori di altri parametri.
`OTEL_EXPORTER_OTLP_PROTOCOL`	Imposta su `http/protobuf` per inviare metriche e tracce all' CloudWatch utilizzo di HTTP.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Imposta per `http://127.0.0.1:4316/v1/metrics` inviare le metriche al sidecar. CloudWatch
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Imposta su `http://127.0.0.1:4316/v1/traces` per inviare tracce al sidecar. CloudWatch
`OTEL_TRACES_SAMPLER`	Definisci X-Ray come campionatore di tracce.
`OTEL_PROPAGATORS`	Aggiungi X-Ray come uno dei propagatori.
`OTEL_PYTHON_DISTRO`	Impostato per `aws_distro` utilizzare la strumentazione ADOT Python.
`OTEL_PYTHON_CONFIGURATOR`	Impostato `aws_configuration` per utilizzare la configurazione ADOT Python.
`PYTHONPATH`	Sostituisci `$APP_PATH` con la posizione della directory di lavoro dell'applicazione all'interno del contenitore. Questo è necessario affinché l'interprete Python trovi i moduli dell'applicazione.
`DJANGO_SETTINGS_MODULE`	Richiesto solo per le applicazioni Django. Impostalo sulla posizione del file dell'applicazione Django. `settings.py` Sostituisci `$PATH_TO_SETTINGS` .

5. Monta il volume `opentelemetry-auto-instrumentation-python` definito nella fase 1 di questa procedura.

   Per un'applicazione Python, usa quanto segue.

```
{
    "name": "app",
    ...
    "environment": [
        {
            "name": "PYTHONPATH",
            "value": "/otel-auto-instrumentation-python/opentelemetry/
instrumentation/auto_instrumentation:$APP_PATH:/otel-auto-instrumentation-
python"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_PROTOCOL",
            "value": "http/protobuf"
        },
        {
            "name": "OTEL_TRACES_SAMPLER",
            "value": "xray"
        },
        {
            "name": "OTEL_TRACES_SAMPLER_ARG",
            "value": "endpoint=http://localhost:2000"
        },
        {
            "name": "OTEL_LOGS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_PYTHON_DISTRO",
            "value": "aws_distro"
        },
        {
            "name": "OTEL_PYTHON_CONFIGURATOR",
            "value": "aws_configurator"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
            "value": "http://localhost:4316/v1/traces"
        },
```

```
        {
            "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
            "value": "http://localhost:4316/v1/metrics"
        },
        {
            "name": "OTEL_METRICS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
            "value": "true"
        },
        {
            "name": "OTEL_RESOURCE_ATTRIBUTES",
            "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAM"
        },
        {
            "name": "DJANGO_SETTINGS_MODULE",
            "value": "$PATH_TO_SETTINGS.settings"
        }
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

Fase 5: distribuzione dell'applicazione

Crea una nuova revisione della definizione dell'attività e distribuiscila nel tuo cluster di applicazioni. Dovresti vedere tre container nell'attività appena creata:

- `init`
- `ecs-cwagent`
- `app`

Utilizzo di una configurazione personalizzata per l'abilitazione di Application Signals su Amazon EC2

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Per le applicazioni in esecuzione su Amazon EC2 e altre architetture diverse da Amazon EKS, puoi installare e configurare personalmente l' CloudWatch agente e AWS Distro. OpenTelemetry Su queste architetture abilitate con una configurazione personalizzata di Application Signals, Application Signals non rileva automaticamente i nomi dei tuoi servizi o dei cluster o host su cui vengono eseguiti. Devi specificare questi nomi durante la configurazione personalizzata e i nomi specificati sono quelli visualizzati nei pannelli di controllo di Application Signals.

I seguenti passaggi sono stati testati su istanze Amazon EC2, ma si prevede che funzionino anche su altre architetture che supportano Distro for. AWS OpenTelemetry

Requisiti

- Per ottenere supporto per Application Signals, è necessario utilizzare la versione più recente sia dell'agente che di Distro for CloudWatch agent. AWS OpenTelemetry
- È necessario che AWS CLI sia installato sull'istanza. Consigliamo AWS CLI la versione 2, ma dovrebbe funzionare anche la versione 1. Per ulteriori informazioni sull'installazione di AWS CLI, consulta Installare o aggiornare la versione più recente di AWS CLI.

⚠ Important

Se stai già utilizzando OpenTelemetry un'applicazione che intendi abilitare per Application Signals, consulta OpenTelemetry considerazioni sulla compatibilità prima di abilitare Application Signals.

Fase 1: abilitazione di Application Signals nel tuo account

Se non hai ancora abilitato Application Signals in questo account, devi concedere ad Application Signals le autorizzazioni necessarie per scoprire i tuoi servizi. Per far ciò, completa le seguenti operazioni. È necessario eseguire questa operazione solo una volta per account.

Per abilitare Application Signals per le tue applicazioni

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Servizi.
3. Scegli Inizia a scoprire i tuoi servizi.
4. Seleziona la casella di controllo e scegli Inizia a scoprire i servizi.

   Il completamento di questo passaggio per la prima volta nel tuo account crea il ruolo AWSServiceRoleForCloudWatchApplicationSignalscollegato al servizio. Questo ruolo concede ad Application Signals le seguenti autorizzazioni:

   - `xray:GetServiceGraph`
   - `logs:StartQuery`
   - `logs:GetQueryResults`
   - `cloudwatch:GetMetricData`
   - `cloudwatch:ListMetrics`
   - `tag:GetResources`

   Per ulteriori informazioni su questo ruolo, consulta Autorizzazioni di ruolo collegate al servizio per Application Signals CloudWatch .

Passaggio 2: scarica e avvia l'agente CloudWatch

Per installare l' CloudWatch agente come parte dell'abilitazione di Application Signals su un'istanza Amazon EC2

1. Scarica la versione più recente dell' CloudWatch agente sull'istanza. Se l' CloudWatch agente è già installato sull'istanza, potrebbe essere necessario aggiornarlo. Solo le versioni dell'agente rilasciate il 30 novembre 2023 o successive supportano CloudWatch Application Signals.

   Per informazioni sul download dell' CloudWatch agente, consultaScarica il pacchetto dell' CloudWatch agente.
2. Prima di avviare l' CloudWatch agente, configuralo per abilitare Application Signals. L'esempio seguente è una configurazione CloudWatch dell'agente che abilita Application Signals sia per le metriche che per le tracce su un host EC2.

Puoi creare questo file immettendo il seguente comando:

```
vim amazon-cloudwatch-agent.json
```

Aggiungi quanto segue come contenuto del file.

```
{
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
  },
  "logs": {
    "metrics_collected": {
      "app_signals": {}
    }
  }
}
```

3. Collega le AWSXrayWriteOnlyAccesspolicy CloudWatchAgentServerPolicye IAM al ruolo IAM della tua istanza Amazon EC2.

   a. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.

   b. Scegli Ruoli e trova il ruolo utilizzato dalla tua istanza Amazon EC2. Quindi scegli il nome del ruolo.

   c. Nella scheda Autorizzazioni, scegli Aggiungi autorizzazioni, quindi Collega policy.

   d. Trova CloudWatchAgentServerPolicy. Usa la casella di ricerca se necessario. Quindi seleziona la casella di controllo della policy e seleziona Aggiungi autorizzazioni.

   e. Trova AWSXrayWriteOnlyAccess. Usa la casella di ricerca se necessario. Quindi seleziona la casella di controllo della policy e seleziona Aggiungi autorizzazioni.

4. Avvia l' CloudWatch agente inserendo i seguenti comandi. Sostituisci *`agent-config-file-path`*con il percorso del file di configurazione CloudWatch dell'agente, ad esempio `./amazon-cloudwatch-agent.json`. È necessario includere il prefisso `file:` come mostrato.

```
export CONFIG_FILE_PATH=./amazon-cloudwatch-agent.json
```

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \
-a fetch-config \
-m ec2 -s -c file:$CONFIG_FILE_PATH
```

Fase 3: strumentazione e avvio dell'applicazione

Il passaggio successivo consiste nello strumentare l' CloudWatch applicazione per Application Signals.

Java

Per strumentare le tue applicazioni Java come parte dell'abilitazione di Application Signals su un'istanza Amazon EC2

1. Scarica l'ultima versione dell'agente di strumentazione automatica AWS Distro for OpenTelemetry Java. Puoi scaricare la versione più recente utilizzando questo link. È possibile visualizzare informazioni su tutte le versioni rilasciate in Releases. aws-otel-java-instrumentation
2. Per ottimizzare i vantaggi di Application Signals, utilizza le variabili di ambiente per fornire informazioni aggiuntive prima di avviare l'applicazione. Queste informazioni verranno visualizzate nei pannelli di controllo di Application Signals.

   a. Per la variabile `OTEL_RESOURCE_ATTRIBUTES`, specifica le seguenti informazioni come coppie chiave-valore:

      - `aws.hostedin.environment` imposta l'ambiente in cui viene eseguita l'applicazione. Questo verrà visualizzato come ambiente ospitato dell'applicazione nei pannelli di controllo di Application Signals. Questa chiave di attributo viene utilizzata solo da Application Signals e viene convertita in annotazioni di tracce a raggi X e CloudWatch dimensioni metriche. Se non si fornisce un valore per questa chiave, viene utilizzato il valore predefinito di `Generic`.
      - `service.name` imposta il nome del servizio. Questo verrà visualizzato come nome del servizio per l'applicazione nei pannelli di controllo di Application Signals. Se non si fornisce un valore per questa chiave, viene utilizzato il valore predefinito di `unknown_service`.

   b. Per la variabile `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`, specifica l'URL dell'endpoint di base in cui esportare le tracce. L' CloudWatch agente espone

4315 come porta OLTP. Su Amazon EC2, poiché le applicazioni comunicano con l' CloudWatch agente locale, è necessario impostare questo valore su `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315`

c. Per la variabile `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`, specifica l'URL dell'endpoint di base in cui esportare i parametri. L' CloudWatch agente espone 4315 come porta OLTP. Su Amazon EC2, poiché le applicazioni comunicano con l' CloudWatch agente locale, è necessario impostare questo valore su `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315`

d. Per la `JAVA_TOOL_OPTIONS` variabile, specificare il percorso in cui è archiviato l'agente di strumentazione automatica AWS Distro for OpenTelemetry Java.

```
export JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH'
```

Per esempio:

```
export ADOT_AGENT_PATH=./aws-opentelemetry-agent.jar
```

e. Per la variabile `OTEL_METRICS_EXPORTER`, si consiglia di impostare il valore su `none`. Questa operazione disabilita gli esportatori di altri parametri in modo che venga utilizzato solo l'esportatore Application Signals.

f. Per la `OTEL_AWS_APP_SIGNALS_ENABLED` variabile, abilitate SpanMetricProcessor (SMP) impostandolo su. `OTEL_AWS_APP_SIGNALS_ENABLED` `true` Questo genera i parametri di Application Signals a partire dalle tracce.

3. Avvia l'applicazione con le variabili di ambiente illustrate nel passaggio precedente. Di seguito è riportato un esempio di script di avvio.

```
JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH' \
OTEL_METRICS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315 \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315 \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$YOUR_HOST_ENV,service.name=
$YOUR_SVC_NAME \
java -jar $MY_JAVA_APP.jar
```

Python

Per strumentare le tue applicazioni Python come parte dell'abilitazione di Application Signals su un'istanza Amazon EC2

1. Scarica l'ultima versione dell'agente di AWS strumentazione automatica Distro for OpenTelemetry Python. Installarlo eseguendo il seguente comando .

   ```
   pip install aws-opentelemetry-distro
   ```

   È possibile visualizzare informazioni su tutte le versioni rilasciate nella AWS strumentazione Distro for OpenTelemetry Python.

2. Per ottimizzare i vantaggi di Application Signals, utilizza le variabili di ambiente per fornire informazioni aggiuntive prima di avviare l'applicazione. Queste informazioni verranno visualizzate nei pannelli di controllo di Application Signals.

   a. Per la variabile `OTEL_RESOURCE_ATTRIBUTES`, specifica le seguenti informazioni come coppie chiave-valore:

      - `aws.hostedin.environment` imposta l'ambiente in cui viene eseguita l'applicazione. Questo verrà visualizzato come ambiente ospitato dell'applicazione nei pannelli di controllo di Application Signals. Questa chiave di attributo viene utilizzata solo da Application Signals e viene convertita in annotazioni di tracce a raggi X e CloudWatch dimensioni metriche. Se non si fornisce un valore per questa chiave, viene utilizzato il valore predefinito di `Generic`.
      - `service.name` imposta il nome del servizio. Questo verrà visualizzato come nome del servizio per l'applicazione nei pannelli di controllo di Application Signals. Se non si fornisce un valore per questa chiave, viene utilizzato il valore predefinito di `unknown_service`.

   b. Per la `OTEL_EXPORTER_OTLP_PROTOCOL` variabile, specificate di `http/protobuf` esportare i dati di telemetria tramite HTTP negli endpoint dell' CloudWatch agente elencati nei passaggi seguenti.

   c. Per la variabile `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`, specifica l'URL dell'endpoint di base in cui esportare le tracce. L' CloudWatch agente espone 4316 come porta OLTP su HTTP. Su Amazon EC2, poiché le applicazioni comunicano con l' CloudWatch agente locale, è necessario impostare questo valore su

`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces`

d. Per la variabile `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`, specifica l'URL dell'endpoint di base in cui esportare i parametri. L' CloudWatch agente espone 4316 come porta OLTP su HTTP. Su Amazon EC2, poiché le applicazioni comunicano con l' CloudWatch agente locale, è necessario impostare questo valore su `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics`

e. Per la variabile `OTEL_METRICS_EXPORTER`, si consiglia di impostare il valore su `none`. Questa operazione disabilita gli esportatori di altri parametri in modo che venga utilizzato solo l'esportatore Application Signals.

f. Per la `OTEL_AWS_APP_SIGNALS_ENABLED` variabile, abilita l'impostazione SpanMetricProcessor by `OTEL_AWS_APP_SIGNALS_ENABLED` su`true`. Questo genera i parametri di Application Signals a partire dalle tracce.

3. Avvia l'applicazione con le variabili di ambiente illustrate nel passaggio precedente. Di seguito è riportato un esempio di script di avvio.

   - Sostituisci `$HOST_ENV` con l'ambiente host in cui è in esecuzione l'applicazione. Questo verrà visualizzato come ambiente Hosted In dell'applicazione, nei dashboard di Application Signals.
   - Sostituisci `$SVC_NAME` con il nome della tua applicazione. Questo verrà visualizzato come nome dell'applicazione, nei dashboard di Application Signals.
   - Sostituiscilo `$PYTHON_APP` con la posizione e il nome dell'applicazione.

```
OTEL_METRICS_EXPORTER=none \
OTEL_LOGS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_PYTHON_DISTRO=aws_distro \
OTEL_PYTHON_CONFIGURATOR=aws_configurator \
OTEL_EXPORTER_OTLP_PROTOCOL=http/protobuf \
OTEL_TRACES_SAMPLER=xray \
OTEL_TRACES_SAMPLER_ARG="endpoint=http://localhost:2000" \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$HOST_ENV,service.name=
$SVC_NAME \
```

```
opentelemetry-instrument python $PYTHON_APP.py
```

Prima di abilitare Application Signals per le tue applicazioni Python, tieni presente le seguenti considerazioni.

- In alcune applicazioni containerizzate, una variabile di `PYTHONPATH` ambiente mancante a volte può impedire l'avvio dell'applicazione. Per risolvere questo problema, assicuratevi di impostare la variabile di `PYTHONPATH` ambiente sulla posizione della directory di lavoro dell'applicazione. Ciò è dovuto a un problema noto con la OpenTelemetry strumentazione automatica. Per ulteriori informazioni su questo problema, vedere L'impostazione della strumentazione automatica in Python di PYTHONPATH non è conforme.
- Per le applicazioni Django, ci sono configurazioni aggiuntive richieste, che sono descritte nella documentazione di Python OpenTelemetry .
  - Usa il `--noreload` flag per impedire il ricaricamento automatico.
  - Imposta la variabile di `DJANGO_SETTINGS_MODULE` ambiente sulla posizione del file dell'`settings.py`applicazione Django. Ciò garantisce che OpenTelemetry possa accedere e integrarsi correttamente con le impostazioni di Django.

Risoluzione dei problemi relativi all'installazione di Application Signals

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Questa sezione contiene suggerimenti per la risoluzione dei problemi relativi a CloudWatch Application Signals.

Argomenti

L'applicazione non si avvia dopo l'abilitazione di Application Signals

Se l'applicazione su un cluster Amazon EKS non si avvia dopo aver abilitato Application Signals sul cluster, verifica quanto segue:

- Verifica se l'applicazione è stata strumentata da un'altra soluzione di monitoraggio. Application Signals non supporta la coesistenza con altre soluzioni di strumentazione.
- Verifica che l'applicazione soddisfi i requisiti di compatibilità per utilizzare Application Signals. Per ulteriori informazioni, consulta Sistemi supportati da Application Signals .
- Se l'applicazione non è riuscita a recuperare gli artefatti di Application Signals come l'agente AWS Distro for Java OpenTelemetery o Python e CloudWatch le immagini degli agenti, potrebbe trattarsi di un problema di rete.

Per mitigare il problema, rimuovete l'annotazione `instrumentation.opentelemetry.io/inject-java: "true"` o `instrumentation.opentelemetry.io/inject-python: "true"` dal manifesto di distribuzione dell'applicazione e ridistribuite l'applicazione. Quindi controlla se l'applicazione funziona.

L'applicazione Python non si avvia dopo l'attivazione di Application Signals

È noto che nella OpenTelemetry strumentazione automatica una variabile di `PYTHONPATH` ambiente mancante a volte può impedire l'avvio dell'applicazione. Per risolvere il problema, assicuratevi di impostare la variabile di `PYTHONPATH` ambiente sulla posizione della directory di lavoro dell'applicazione. Per ulteriori informazioni su questo problema, vedere L'impostazione della strumentazione automatica di Python di PYTHONPATH non è conforme al comportamento di risoluzione dei moduli di Python, interrompendo le applicazioni Django.

Per le applicazioni Django, ci sono configurazioni aggiuntive richieste, che sono descritte nella documentazione di Python OpenTelemetry .

- Usa il `--noreload` flag per impedire il ricaricamento automatico.
- Imposta la variabile di `DJANGO_SETTINGS_MODULE` ambiente sulla posizione del file dell'`settings.py`applicazione Django. Ciò garantisce che OpenTelemetry possa accedere e integrarsi correttamente con le impostazioni di Django.

Mancano i dati di telemetria in X-Ray CloudWatch

Se nei pannelli di controllo di Application Signals mancano parametri o tracce, le cause potrebbero essere le seguenti. Esamina queste cause solo se hai atteso per 15 minuti che Application Signals raccogliesse e visualizzasse i dati dall'ultimo aggiornamento.

- Assicurati che la libreria e il framework che stai utilizzando siano supportati dall'agente Java ADOT. Per ulteriori informazioni, consulta Librerie/framework.
- Assicurati che l'agente sia in esecuzione. CloudWatch Per prima cosa controlla lo stato dei pod degli CloudWatch agenti e assicurati che siano tutti a `Running` posto.

  ```
  kubectl -n amazon-cloudwatch get pods.
  ```

  Aggiungi quanto segue al file di configurazione dell' CloudWatch agente per abilitare i log di debug, quindi riavvia l'agente.

  ```
  "agent": {
  >>>>>>> streams
    "region": "${REGION}",
    "debug": true
  },
  ```

  Quindi verifica la presenza di errori nei pod dell'agente. CloudWatch
- Verifica la presenza di problemi di configurazione con l' CloudWatch agente. Verificate che quanto segue sia ancora presente nel file di configurazione dell' CloudWatch agente e che l'agente sia stato riavviato da quando è stato aggiunto.

  ```
  "agent": {
    "region": "${REGION}",
    "debug": true
  },
  ```

  Quindi controlla i registri di OpenTelemetry debug per verificare la presenza di messaggi di errore come. `ERROR io.opentelemetry.exporter.internal.grpc.OkHttpGrpcExporter - Failed to export ...` Questi messaggi potrebbero indicare il problema.

  Se questo non risolve il problema, scarica e controlla le variabili di ambiente con nomi che iniziano con `OTEL_` descrivendo il pod con il comando `kubectl describe pod`.

- Per abilitare la registrazione di debug in OpenTelemetry Python, imposta la variabile di ambiente su `debug` e ridistribuisci l'`OTEL_PYTHON_LOG_LEVEL`applicazione.
- Verifica la presenza di autorizzazioni errate o insufficienti per l'esportazione dei dati dall'agente. CloudWatch Se vedi `Access Denied` dei messaggi nei registri degli CloudWatch agenti, questo potrebbe essere il problema. È possibile che le autorizzazioni applicate durante l'installazione dell' CloudWatch agente siano state successivamente modificate o revocate.
- Verifica la presenza di un problema relativo a AWS Distro for OpenTelemetry (ADOT) durante la generazione di dati di telemetria.

  Assicurati che le annotazioni sulla strumentazione `instrumentation.opentelemetry.io/inject-java` e `sidecar.opentelemetry.io/inject-java` vengano applicate alla distribuzione dell'applicazione e che il valore sia `true`. Senza questi, i pod dell'applicazione non saranno dotati di strumenti anche se il componente aggiuntivo ADOT è installato correttamente.

  Quindi, controlla se il container `Init` è applicato all'applicazione e lo stato di `Ready` è `True`. Se il container `init` non è pronto, verifica lo stato del motivo.

  Se il problema persiste, procedi come segue per abilitare la registrazione di debug su Java SDK. OpenTelemetry Quindi cerca i messaggi che iniziano con `ERROR io.telemetry`.

  Per abilitare la registrazione di debug, imposta la variabile di ambiente `OTEL_JAVAAGENT_DEBUG` su true e ridistribuisci l'applicazione.
- È possibile che l'esportatore metric/span stia eliminando i dati. Per scoprirlo, controlla il log dell'applicazione per i messaggi che includono `Failed to export...`
- L' CloudWatch agente potrebbe subire delle limitazioni durante l'invio di metriche o intervalli ad Application Signals. Verifica la presenza di messaggi che indicano una limitazione nei registri degli agenti. CloudWatch

I valori dei parametri di dipendenza sono sconosciuti

Se visualizzi o UnknownOperationcerchi UnknownRemoteServiceun nome o UnknownRemoteOperationun'operazione di dipendenza nei dashboard di Application Signals, controlla se la presenza di punti dati per il servizio remoto sconosciuto e il funzionamento remoto sconosciuto coincidono con le relative implementazioni. È un problema noto su Application Signals e dovrebbe essere corretto in una versione futura.

Gestione di un ConfigurationConflict durante la gestione del componente aggiuntivo Amazon CloudWatch Observability EKS

Quando installi o aggiorni il componente aggiuntivo Amazon CloudWatch Observability EKS, se noti un errore causato da un `Health Issue` di tipo `ConfigurationConflict` con una descrizione che inizia con`Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, è probabile che l' CloudWatch agente e i relativi componenti associati, come il ServiceAccount, il ClusterRole e il, siano ClusterRoleBinding installati nel cluster. Quando il componente aggiuntivo tenta di installare l' CloudWatch agente e i componenti associati, se rileva modifiche nei contenuti, per impostazione predefinita fallisce l'installazione o l'aggiornamento per evitare di sovrascrivere lo stato delle risorse sul cluster.

Se stai tentando di effettuare l'onboarding al componente aggiuntivo Amazon CloudWatch Observability EKS e riscontri questo errore, ti consigliamo di eliminare una configurazione di CloudWatch agente esistente che avevi precedentemente installato sul cluster e quindi di installare il componente aggiuntivo EKS. Assicurati di eseguire il backup di tutte le personalizzazioni che potresti aver apportato alla configurazione originale dell' CloudWatch agente, ad esempio una configurazione personalizzata dell'agente, e di fornirle al componente aggiuntivo Amazon CloudWatch Observability EKS alla prossima installazione o aggiornamento. Se in precedenza avevi installato l' CloudWatch agente per l'onboarding su Container Insights, consulta per ulteriori informazioni. Eliminazione dell' CloudWatch agente e di Fluent Bit for Container Insights

In alternativa, il componente aggiuntivo supporta un'opzione di configurazione per la risoluzione dei conflitti che può specificare `OVERWRITE`. È possibile utilizzare questa opzione per procedere con l'installazione o l'aggiornamento del componente aggiuntivo sovrascrivendo i conflitti nel cluster. Se utilizzi la console Amazon EKS, trovi il metodo di risoluzione dei conflitti selezionando le impostazioni di configurazione facoltative quando crei o aggiorni il componente aggiuntivo. Se utilizzi il AWS CLI, puoi fornire il comando `--resolve-conflicts OVERWRITE` al tuo comando per creare o aggiornare il componente aggiuntivo.

Configurazione di Application Signals

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Questa sezione contiene informazioni sulla configurazione di CloudWatch Application Signals.

Velocità di campionamento della traccia

Per impostazione predefinita, quando si abilita Application Signals, il campionamento centralizzato X-Ray viene abilitato utilizzando le impostazioni di velocità di campionamento predefinite `reservoir=1/s` e `fixed_rate=5%`. Le variabili di ambiente per l'agente AWS Distro for OpenTelemetry (ADOT) SDK sono impostate come segue.

Variabile di ambiente	Valore	Nota
`OTEL_TRACES_SAMPLER`	`xray`	
`OTEL_TRACES_SAMPLER_ARG`	`endpoint=http://cloudwatch-agent.amazon-cloudwatch:2000`	Endpoint dell'agente CloudWatch

Per ulteriori informazioni sulla modifica della configurazione di campionamento, consulta:

- Per modificare il campionamento X-Ray, consulta Personalizzazione delle regole di campionamento
- Per modificare il campionamento ADOT, vedere Configurazione del OpenTelemetry collettore per il campionamento remoto a raggi X

Se desideri disabilitare il campionamento centralizzato X-Ray e utilizzare invece il campionamento locale, imposta i seguenti valori per l'agente Java SDK ADOT come indicato di seguito. L'esempio seguente imposta la velocità di campionamento al 5%.

Variabile di ambiente	Valore
`OTEL_TRACES_SAMPLER`	`parentbased_traceidratio`
`OTEL_TRACES_SAMPLER_ARG`	`0.05`

Per informazioni sulle impostazioni di campionamento più avanzate, consulta OTEL_TRACES_SAMPLER.

Gestisci le operazioni ad alta cardinalità

Application Signals include impostazioni nell' CloudWatch agente che puoi utilizzare per gestire la cardinalità delle tue operazioni e gestire l'esportazione delle metriche per ottimizzare i costi. Per impostazione predefinita, la funzione di limitazione delle metriche diventa attiva quando il numero di operazioni distinte per un servizio nel tempo supera la soglia predefinita di 500. È possibile ottimizzare il comportamento modificando le impostazioni di configurazione.

Determina se la limitazione metrica è attivata

Puoi utilizzare i seguenti metodi per scoprire se è in corso la limitazione metrica predefinita. In tal caso, dovresti prendere in considerazione l'ottimizzazione del controllo della cardinalità seguendo i passaggi nella sezione successiva.

- Nella CloudWatch console, scegli Application Signals, Services. Se vedi un'operazione denominata AllOtherOperationso RemoteOperationdenominata AllOtherRemoteOperations, è in corso una limitazione delle metriche.
- Se alcune metriche raccolte da Application Signals hanno il valore `AllOtherOperations` corrispondente alla loro `Operation` dimensione, allora si verifica una limitazione delle metriche.
- Se alcune metriche raccolte da Application Signals hanno il valore `AllOtherRemoteOperations` corrispondente alla loro `RemoteOperation` dimensione, allora si verifica una limitazione delle metriche.

Ottimizza il controllo della cardinalità

Per ottimizzare il controllo della cardinalità, puoi fare quanto segue:

- Crea regole personalizzate per aggregare le operazioni.
- Configura la tua politica di limitazione delle metriche.

Crea regole personalizzate per aggregare le operazioni

Le operazioni con elevata cardinalità possono a volte essere causate da valori univoci inappropriati estratti dal contesto. Ad esempio, l'invio di richieste HTTP/S che includono ID utente o ID di sessione nel percorso può portare a centinaia di operazioni diverse. Per risolvere questi problemi, si consiglia di configurare l' CloudWatch agente con regole di personalizzazione per riscrivere queste operazioni.

Nei casi in cui si verifichi un'impennata nella generazione di numerose metriche diverse tramite `RemoteOperation` chiamate individuali, ad esempio`PUT /api/customer/owners/123`, e richieste simili`PUT /api/customer/owners/456`, si consiglia di consolidare queste operazioni in un'unica operazione. `RemoteOperation` Un approccio consiste nello standardizzare tutte le `RemoteOperation` chiamate che iniziano con un formato uniforme, `PUT /api/customer/owners/` in particolare. `PUT /api/customer/owners/{ownerId}` Nell'esempio seguente viene descritto quanto segue. Per informazioni su altre regole di personalizzazione, consulta. Abilita CloudWatch Application Signals

```
{
   "logs":{
      "metrics_collected":{
         "app_signals":{
            "rules":[
               {
                  "selectors":[
                     {
                        "dimension":"RemoteOperation",
                        "match":"PUT /api/customer/owners/*"
                     }
                  ],
                  "replacements":[
                     {
                        "target_dimension":"RemoteOperation",
                        "value":"PUT /api/customer/owners/{ownerId}"
                     }
                  ],
                  "action":"replace"
               }
            ]
         }
      }
   }
}
```

In altri casi, potrebbero essere state aggregate metriche ad alta cardinalità e potrebbe non essere chiaro quali metriche specifiche siano incluse. `AllOtherRemoteOperations` L' CloudWatch agente è in grado di registrare le operazioni interrotte. Per identificare le operazioni interrotte, utilizzate la configurazione nell'esempio seguente per attivare la registrazione fino alla ricomparsa del problema. Quindi ispeziona i log degli CloudWatch agenti (accessibili tramite contenitore `stdout` o file di registro EC2) e cerca la parola chiave. `drop metric data`

```
{
  "agent": {
    "config": {
      "agent": {
        "debug": true
      },
      "traces": {
        "traces_collected": {
          "app_signals": {
          }
        }
      },
      "logs": {
        "metrics_collected": {
          "app_signals": {
            "limiter": {
              "log_dropped_metrics": true
            }
          }
        }
      }
    }
  }
}
```

Crea la tua politica di limitazione delle metriche

Se la configurazione di limitazione delle metriche predefinita non riguarda la cardinalità del servizio, puoi personalizzare la configurazione del limitatore metrico. A tale scopo, aggiungi una `limiter` sezione sotto la `logs/metrics_collected/app_signals` sezione del file di configurazione dell'agente. CloudWatch

L'esempio seguente riduce la soglia di limitazione delle metriche da 500 metriche distinte a 100.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "limiter": {
          "drop_threshold": 100
        }
      }
```

```
    }
  }
}
```

Obiettivi del livello di servizio (SLO)

> ⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

È possibile utilizzare Application Signals per creare obiettivi del livello di servizio per i servizi destinati alle operazioni aziendali critiche. Creando SLO su questi servizi, sarai in grado di tracciarli sulla dashboard SLO, offrendoti una at-a-glance panoramica delle tue operazioni più importanti.

Oltre a creare una panoramica che gli operatori possono utilizzare per visualizzare lo stato attuale delle operazioni critiche, puoi utilizzare gli SLO per monitorare le prestazioni a lungo termine dei tuoi servizi, per assicurarti che soddisfino le tue aspettative. Se hai stipulato contratti sul livello di servizio con i clienti, gli SLO sono un ottimo strumento per accertarti che vengano rispettati.

La valutazione dello stato dei servizi con gli SLO inizia con la definizione di obiettivi chiari e misurabili basati su parametri delle prestazioni chiave: gli indicatori del livello di servizio (SLI). Uno SLO tiene traccia delle prestazioni SLI rispetto alla soglia e all'obiettivo prefissati e riporta in che misura le prestazioni delle applicazioni si avvicinano alla soglia.

Application Signals ti aiuta a impostare gli SLO sui parametri delle prestazioni chiave. Application Signals raccoglie automaticamente parametri di `Latency` e `Availability` per ogni servizio e operazione che individua e questi parametri sono spesso ideali da utilizzare come SLI. Con la procedura guidata di creazione degli SLO, puoi utilizzare questi parametri per i tuoi SLO. Puoi quindi monitorare lo stato di tutti i tuoi SLO tramite i pannelli di controllo di Application Signals.

Puoi impostare gli SLO su operazioni specifiche che il tuo servizio chiama o utilizza. Puoi utilizzare qualsiasi CloudWatch metrica o espressione metrica come SLI, oltre a utilizzare le metriche and. `Latency` `Availability`

La creazione di SLO è molto importante per ottenere il massimo vantaggio da Application Signals. CloudWatch Dopo aver creato gli SLO, puoi visualizzarne lo stato nella console Application Signals per vedere rapidamente quali di questi servizi e operazioni critici stanno funzionando bene e quali no. La possibilità di monitorare gli SLO offre i seguenti principali vantaggi:

- Gli operatori di servizi vedere più facilmente l'integrità operativa attuale dei servizi critici confrontandoli con lo SLI. In questo modo possono controllare e identificare rapidamente servizi e operazioni non funzionanti.
- È possibile monitorare le prestazioni dei servizi rispetto a obiettivi aziendali misurabili per periodi di tempo più lunghi.

Scegliendo su cosa impostare gli SLO, dai la priorità a ciò che è importante per te. I pannelli di controllo di Application Signals mostrano automaticamente informazioni su ciò a cui hai dato priorità.

Quando crei uno SLO, puoi anche scegliere di creare CloudWatch allarmi contemporaneamente per monitorare gli SLO. Puoi impostare allarmi per monitorare le violazioni della soglia e anche i livelli di avviso. Questi allarmi possono avvisarti automaticamente se i parametri SLO superano la soglia che hai impostato o se si avvicinano a una soglia di avviso. Ad esempio, uno SLO che si avvicina alla soglia di avviso può avvisarti che il tuo team dovrebbe rallentare la frequenza di abbandono dell'applicazione per assicurarsi che gli obiettivi di prestazione a lungo termine vengano raggiunti.

Argomenti

Concetti di SLO

Uno SLO include i componenti seguenti:

- Un indicatore del livello di servizio (SLI), che è un parametro chiave delle prestazioni specificato dall'utente. Rappresenta il livello di prestazione desiderato per l'applicazione. Application Signals raccoglie automaticamente parametri chiave di `Latency` e `Availability` per i servizi e le operazioni che individua e questi parametri sono spesso ideali da utilizzare come SLI.

  Sei tu a scegliere la soglia da utilizzare per il tuo SLI. Ad esempio, 200 ms per la latenza.
- Un obiettivo o un obiettivo di raggiungimento, ovvero la percentuale di tempo in cui si prevede che lo SLI raggiunga la soglia in ogni intervallo di tempo. Gli intervalli di tempo possono essere brevi, come ore, o lunghi, come un anno.

Gli intervalli possono essere intervalli di calendario o intervalli ricorrenti.

- Gli intervalli del calendario sono allineati al calendario, ad esempio un SLO registrato mensilmente. CloudWatch regola automaticamente i dati relativi a salute, budget e risultati scolastici in base al numero di giorni in un mese. Gli intervalli di calendario sono più adatti agli obiettivi aziendali che sono misurati in base al calendario.
- Gli intervalli ricorrenti sono calcolati su base sequenziale. Gli intervalli ricorrenti sono più adatti per monitorare l'esperienza utente recente della tua applicazione.

- Il periodo è un periodo di tempo più breve e più periodi costituiscono un intervallo. Le prestazioni dell'applicazione vengono confrontate allo SLI durante ogni periodo compreso nell'intervallo. Per ogni periodo, si stabilisce che l'applicazione ha raggiunto o non ha raggiunto le prestazioni previste.

Ad esempio, un obiettivo del 99% con un intervallo di calendario di un giorno e un periodo di 1 minuto significa che l'applicazione deve soddisfare o raggiungere la soglia di successo nel 99% dei periodi di 1 minuto durante il giorno. In caso affermativo, lo SLO è stato raggiunto per quel giorno. Il giorno successivo è previsto un nuovo intervallo di valutazione e l'applicazione deve soddisfare o raggiungere la soglia di successo nel 99% dei periodi di 1 minuto durante il secondo giorno per soddisfare lo SLO per il secondo giorno.

Uno SLI può essere basato su uno dei nuovi parametri dell'applicazione standard raccolte da Application Signals. In alternativa, può essere qualsiasi espressione metrica o CloudWatch metrica. I parametri dell'applicazione standard che è possibile utilizzare per una SLI sono `Latency` e `Availability`. `Availability` rappresenta le risposte andate a buon fine divise per il totale delle richieste. Viene calcolata come (1 - frequenza di errore)*100, dove le risposte di errore sono `5xx` errori. Le risposte andate a buon fine sono risposte prive di errori `5XX`. Le risposte `4XX` vengono considerate come andate a buon fine.

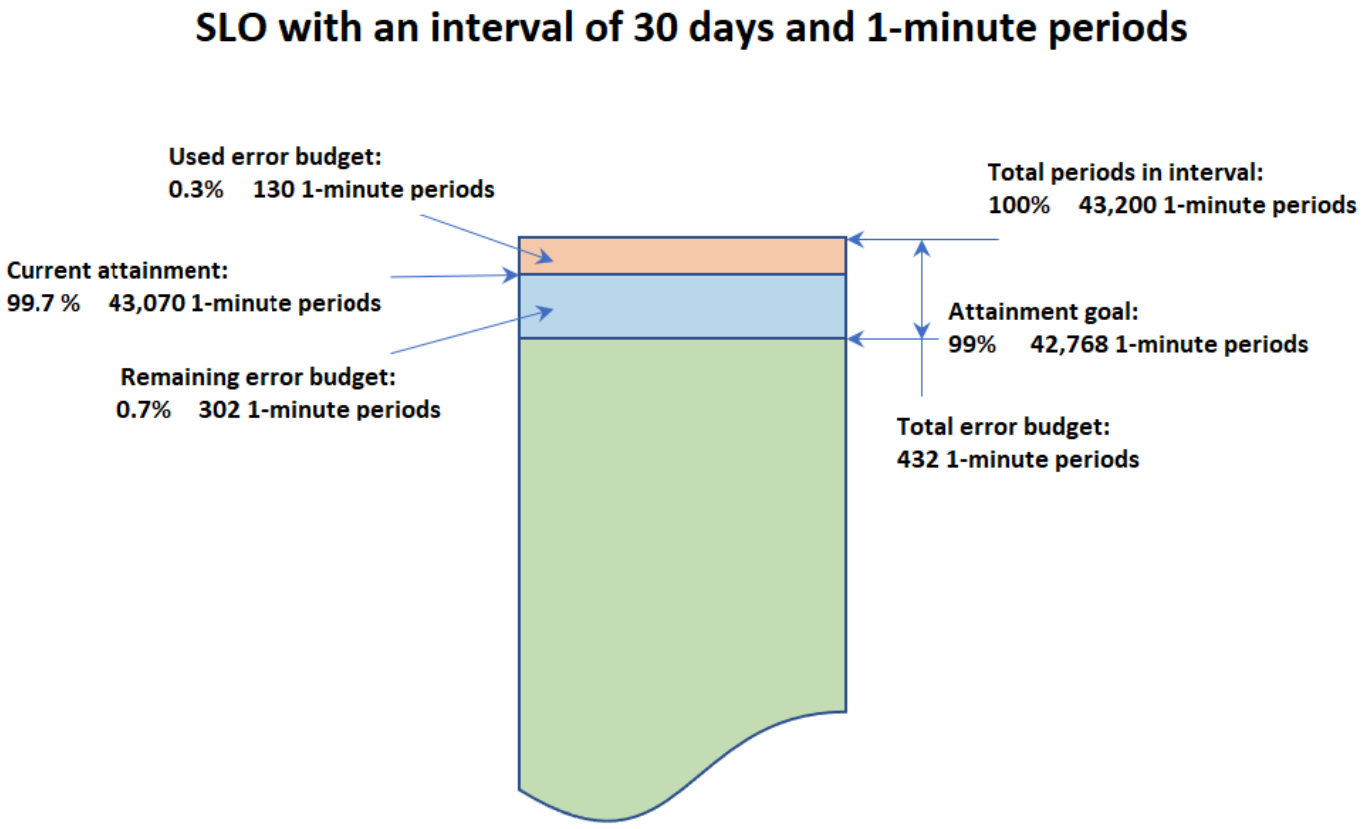
> **Note**
>
> Attualmente sono supportati solo i calcoli basati sul periodo. Il supporto per i calcoli basati su volumi o richieste è previsto per le versioni future.

Calcolo di budget di errore e raggiungimento

Quando si visualizzano le informazioni su uno SLO, vengono visualizzati lo stato di integrità corrente e il relativo budget di errore. Il budget di errore è la quantità di tempo all'interno dell'intervallo che

può superare la soglia ma consentire comunque di rispettare lo SLO. Il budget di errore totale è la quantità totale di tempo di superamento della soglia che può essere tollerato durante l'intero intervallo. Il budget di errore residuo è la quantità di tempo residuo di superamento della soglia che può essere tollerato durante l'intervallo corrente. Questo si calcola sottraendo dal budget di errore totale la quantità di tempo in cui la soglia è già stata superata.

L'immagine seguente illustra i concetti di budget di raggiungimento e di errore per un obiettivo con un intervallo di 30 giorni, periodi di 1 minuto e un obiettivo di raggiungimento del 99%. 30 giorni contengono 43.200 periodi da 1 minuto. Il 99% di 43.200 è 42.768, quindi per raggiungere lo SLO è necessario che 42.768 raggiungano l'obiettivo. Finora, nell'intervallo attuale, 130 periodi di 1 minuto non hanno raggiunto l'obiettivo.



Determinazione del successo in ogni periodo

All'interno di ogni periodo, i dati SLI vengono aggregati in un unico punto dati basato sulla statistica utilizzata per lo SLI. Questo punto dati rappresenta l'intera durata del periodo. Quel singolo punto dati viene confrontato con la soglia SLI per determinare se il periodo ha raggiunto l'obiettivo. La visualizzazione nel pannello di controllo dei periodi che non hanno raggiunto l'obiettivo durante l'intervallo di tempo corrente può avvisare gli operatori del servizio che è necessario controllarlo.

Se si ritiene che il periodo non abbia raggiunto l'obiettivo, l'intera durata del periodo viene conteggiata come non riuscito ai fini del calcolo del budget di errore. Il monitoraggio del budget di errore consente di sapere se il servizio sta ottenendo le prestazioni desiderate per un periodo di tempo più lungo.

Creazione di uno SLO.

Ti consigliamo di impostare SLO sia di latenza che di disponibilità sulle tue applicazioni critiche. Questi parametri raccolti da Application Signals sono in linea con gli obiettivi aziendali comuni.

Puoi anche impostare gli SLO su qualsiasi CloudWatch metrica o espressione matematica metrica che risulti in una singola serie temporale.

La prima volta che crei uno SLO nel tuo account, crea CloudWatch automaticamente il ruolo AWSServiceRoleForCloudWatchApplicationSignalscollegato al servizio nel tuo account, se non esiste già. Questo ruolo collegato al servizio consente di CloudWatch raccogliere dati di CloudWatch log, dati di tracciamento X-Ray, dati di CloudWatch metrica e dati di etichettatura dalle applicazioni del tuo account. Per ulteriori informazioni sui ruoli collegati ai servizi, vedere. CloudWatch Utilizzo di ruoli collegati ai servizi per CloudWatch

Creazione di uno SLO

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegli Obiettivi del livello di servizio (SLO).
3. Scegli Crea SLO.
4. Inserisci un nome per lo SLO. L'inclusione del nome di un servizio o di un'operazione, insieme a parole chiave appropriate come latenza o disponibilità, ti aiuterà a identificare rapidamente cosa indica lo stato SLO durante la valutazione.
5. In Imposta l'indicatore del livello di servizio (SLI), effettua una delle seguenti operazioni:
   - Per impostare lo SLO su uno dei parametri dell'applicazione standard `Latency` o `Availability`:

     a. Seleziona Operazione del servizio.

     b. Seleziona il servizio che lo SLO monitorerà.

     c. Seleziona l'operazione che lo SLO monitorerà.

     I menu a discesa Seleziona servizio e Seleziona operazione sono popolati da servizi e operazioni che sono stati attivi nelle ultime 24 ore.

d. Seleziona Disponibilità o Latenza, quindi imposta la soglia.

- Per impostare lo SLO su qualsiasi CloudWatch metrica o espressione matematica CloudWatch metrica:

  a. Scegliete Metric. CloudWatch

  b. Scegli Seleziona CloudWatch metrica.

     Viene visualizzata la schermata Seleziona parametro. Utilizza le schede Sfoglia o Query per trovare il parametro desiderato oppure crea un'espressione matematica del parametro.

     Dopo aver selezionato il parametro desiderato, scegli la scheda Parametri nel grafico e seleziona le statistiche e il periodo da utilizzare per lo SLO. Quindi, scegli Seleziona parametro.

     Per informazioni su queste schermate, consulta Rappresentazione grafica di un parametro e Aggiungere un'espressione matematica a un grafico CloudWatch .

  c. Per Imposta condizione, seleziona un operatore di confronto e una soglia per lo SLO da utilizzare come indicatore di successo.

6. Se hai selezionato Operazione del servizio nel passaggio 5, puoi facoltativamente scegliere Impostazioni aggiuntive e quindi regolare la durata del periodo per questo SLO.
7. Imposta l'intervallo e l'obiettivo di raggiungimento per lo SLO. Per ulteriori informazioni sugli intervalli e sugli obiettivi di raggiungimento e su come interagiscono tra loro, consulta Concetti di SLO.
8. (Facoltativo) Imposta uno o più CloudWatch allarmi o una soglia di avviso per lo SLO.

   a. CloudWatch gli allarmi possono utilizzare Amazon SNS per avvisarti in modo proattivo se un'applicazione non è integra in base alle sue prestazioni SLI.

      Per creare un allarme, seleziona una delle caselle di controllo relative agli allarmi e inserisci o crea l'argomento Amazon SNS da utilizzare per le notifiche quando l'allarme entra nello stato `ALARM`. Per ulteriori informazioni sugli allarmi, consulta. CloudWatch Utilizzo degli CloudWatch allarmi Amazon La creazione di allarmi comporta addebiti. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

   b. Se imposti una soglia di avviso, questa viene visualizzata nelle schermate di Application Signals per aiutarti a identificare gli SLO che rischiano di non essere raggiunti, anche se al momento sono integri.

Per impostare una soglia di avviso, inserisci il valore della soglia in Soglia di avviso. Quando il budget di errore dello SLO è inferiore alla soglia di avviso, lo SLO viene contrassegnato con un avviso in diverse schermate di Application Signals. Le soglie di avviso vengono visualizzate anche nei grafici del budget di errore. Puoi anche creare un allarme di avviso per lo SLO basato sulla soglia di avviso.

9. Per aggiungere tag a questo SLO, scegli la scheda Tag, quindi scegli Aggiungi nuovo tag. Con i tag è possibile a gestire, identificare, organizzare, cercare e filtrare le risorse. Per ulteriori informazioni sui tag, consulta Tagging delle risorse AWS.

> **Note**
>
> Se l'applicazione a cui è correlato questo SLO è registrata AWS Service Catalog AppRegistry, puoi utilizzare il `awsApplication` tag per associare questo SLO a quell'applicazione in cui si riferisce. AppRegistry Per ulteriori informazioni, consulta Cos'è? AppRegistry

10. Scegli Crea SLO. Se hai scelto anche di creare uno o più allarmi, il nome del pulsante cambia di conseguenza.

Visualizza e valuta lo stato SLO

Puoi visualizzare rapidamente lo stato dei tuoi SLO utilizzando gli obiettivi del livello di servizio o le opzioni Services nella CloudWatch console. La visualizzazione Servizi fornisce una at-a-glance panoramica del rapporto tra i servizi non integri, calcolato in base agli SLO che hai impostato. Per ulteriori informazioni sull'uso dell'opzione Servizi, consulta Monitoraggio dell'integrità operativa delle applicazioni con Application Signals.

La visualizzazione Obiettivi del livello di servizio offre una panoramica macro dell'organizzazione. È possibile visualizzare gli SLO soddisfatti e non soddisfatti nel loro complesso. In questo modo puoi avere un'idea di quanti dei tuoi servizi e delle tue operazioni rispondono alle tue aspettative per periodi di tempo più lunghi, in base agli SLI che hai scelto.

Per visualizzare tutti gli SLO utilizzando la visualizzazione Obiettivi del livello di servizio

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegli Obiettivi del livello di servizio (SLO).

Viene visualizzato l'elenco Obiettivi del livello di servizio (SLO).

Puoi visualizzare rapidamente lo stato attuale degli SLO nella colonna Stato SLI. Per ordinare gli SLO in modo che tutti gli SLO non integri siano in cima all'elenco, scegli la colonna dello stato SLI finché gli SLO non integri non saranno tutti in cima alla lista.

La tabella dello SLO contiene le colonne predefinite riportate di seguito. Puoi modificare le colonne da visualizzare selezionando l'icona a forma di ingranaggio sopra l'elenco. Per ulteriori informazioni su obiettivi, SLI, raggiungimento e intervalli, consulta Concetti di SLO.

- Il nome dello SLO.
- La colonna Obiettivo mostra la percentuale di periodi di ogni intervallo che devono soddisfare correttamente la soglia SLI affinché venga raggiunto l'obiettivo SLO. Mostra anche la durata dell'intervallo per lo SLO.
- Lo stato SLI indica l'integrità dello stato operativo corrente dell'applicazione. Se un periodo dell'intervallo di tempo attualmente selezionato non era integro per lo SLO, lo stato SLI mostra Non integro.
- Il raggiungimento finale è il livello di successo raggiunto alla fine dell'intervallo di tempo selezionato. Ordina in base a questa colonna per vedere gli SLO che rischiano maggiormente di non essere rispettati.
- Il delta di raggiungimento è la differenza nel livello di raggiungimento tra l'inizio e la fine dell'intervallo di tempo selezionato. Un delta negativo indica che il parametro tende verso il basso. Ordina in base a questa colonna per vedere le tendenze più recenti degli SLO.
- Il budget di errore finale (%) è la percentuale di tempo totale all'interno del periodo in cui è possibile che si verifichino periodi non integri senza impedire che lo SLO sia raggiunto con successo. Se lo si imposta al 5% e lo SLI non è integro nel 5% o meno dei periodi rimanenti dell'intervallo, lo SLO viene comunque raggiunto con successo.
- Il delta del budget di errore è la differenza nel budget di errore tra l'inizio e la fine dell'intervallo di tempo selezionato. Un delta negativo indica che il parametro tende verso la non riuscita.
- Il budget di errore finale (tempo) è la quantità di tempo effettivo nell'intervallo che può essere non integro senza impedire che lo SLO sia raggiunto con successo. Ad esempio, se si tratta di 14 minuti, se lo SLI non è integro per meno di 14 minuti durante l'intervallo rimanente, lo SLO verrà comunque raggiunto con successo.
- Le colonne Servizio, Operazione e Tipo mostrano informazioni sul servizio e sull'operazione per cui è impostato questo SLO.

3. Per visualizzare i grafici del raggiungimento e del budget di errore per uno SLO, seleziona il pulsante di opzione accanto al nome dello SLO.

   I grafici nella parte superiore della pagina mostrano il raggiungimento dello SLO e lo stato del budget di errore. Viene inoltre visualizzato un grafico sul parametro SLI associato a questo SLO.

4. Per valutare ulteriormente uno SLO che non soddisfa il suo obiettivo, scegli il nome del servizio o dell'operazione associato a tale SLO. Verrà visualizzata la pagina dei dettagli dove puoi effettuare ulteriori operazioni di valutazione. Per ulteriori informazioni, consulta Visualizza l'attività di servizio dettagliata e lo stato operativo con la pagina dei dettagli del servizio.

5. Per modificare l'intervallo di tempo dei grafici e delle tabelle sulla pagina, scegli un nuovo intervallo di tempo nella parte superiore dello schermo.

Modifica di uno SLO esistente

Segui questa procedura per modificare uno SLO esistente. Quando modifichi uno SLO, puoi cambiare solo la soglia, l'intervallo, l'obiettivo di raggiungimento e i tag. Per modificare altri aspetti come il servizio, l'operazione o il parametro, crea un nuovo SLO invece di modificarne uno esistente.

La modifica di parte di una configurazione principale dello SLO, come il periodo o la soglia, invalida tutti i dati e le valutazioni precedenti relativi al raggiungimento e all'integrità. Questa operazione elimina e ricrea efficacemente lo SLO.

> **Note**
>
> Quando modifichi uno SLO, gli allarmi associati non vengono aggiornati automaticamente. Potrebbe essere necessario aggiornare gli allarmi per mantenerli sincronizzati con lo SLO.

Per modificare uno SLO esistente

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegli Obiettivi del livello di servizio (SLO).
3. Scegli il pulsante di opzione accanto allo SLO che desideri modificare, quindi scegli Operazioni, Modifica SLO.
4. Apporta le modifiche desiderate e seleziona Salva modifiche.

Eliminazione di uno SLO

Segui questa procedura per eliminare uno SLO esistente.

> **Note**
>
> Quando elimini uno SLO, gli allarmi associati non vengono eliminati automaticamente. Dovrai eliminarli tu stesso. Per ulteriori informazioni, consulta Gestione degli allarmi.

Per eliminare uno SLO

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegli Obiettivi del livello di servizio (SLO).
3. Scegli il pulsante di opzione accanto allo SLO che desideri modificare, quindi scegli Operazioni, Elimina SLO.
4. Scegli Conferma.

Monitoraggio dell'integrità operativa delle applicazioni con Application Signals

> **Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.**

Usa Application Signals all'interno della CloudWatch console per monitorare e risolvere i problemi relativi allo stato operativo delle tue applicazioni:

- Monitora i servizi delle applicazioni: nell'ambito del monitoraggio operativo quotidiano, utilizza la pagina Servizi per visualizzare un riepilogo di tutti i tuoi servizi. Visualizza i servizi con la frequenza di errore o la latenza più elevati e scopri quali servizi hanno indicatori di livello di servizio (SLI) non funzionanti. Seleziona un servizio per aprire la pagina dei dettagli del servizio e visualizzare parametri dettagliati, operazioni di servizio, canary Synthetics e richieste client. Questo può aiutarti a identificare la causa principale dei problemi operativi e risolverli.

- Ispeziona la topologia dell'applicazione: utilizza la Mappa del servizio per comprendere e monitorare la topologia dell'applicazione nel tempo, comprese le relazioni tra client, canary Synthetics, servizi e dipendenze. Visualizza istantaneamente lo stato dell'indicatore del livello di servizio (SLI) e visualizza i parametri chiave come il volume delle chiamate, la frequenza di errore e la latenza. Esplora e ottieni informazioni più dettagliate nella pagina dei dettagli del servizio.

Osserva uno scenario di esempio che dimostra come queste pagine possono essere utilizzate per risolvere rapidamente un problema di integrità del servizio operativo, a partire dal rilevamento iniziale fino all'identificazione della causa principale.

In che modo Application Signals consente il monitoraggio dello stato operativo

Dopo aver abilitato l'applicazione per Application Signals, i servizi applicativi, le API e le relative dipendenze vengono automaticamente individuati e visualizzati nelle pagine Servizi, Dettagli del servizio e Mappa del servizio. Application Signals raccoglie informazioni da più fonti per consentire il rilevamento servizi e il monitoraggio dello stato operativo:

- AWS Distro for OpenTelemetry (ADOT): come parte dell'abilitazione dei segnali di applicazione, una libreria di strumentazione automatica OpenTelemetry Java è configurata per emettere metriche e tracce raccolte dall'agente. CloudWatch I parametri e le tracce vengono utilizzati per consentire l'individuazione di servizi, operazioni, dipendenze e altre informazioni sui servizi.
- Obiettivi del livello di servizio (SLO): dopo aver creato gli obiettivi del livello di servizio per i tuoi servizi, le pagine Servizi, Dettagli del servizio e Mappa dei servizi mostrano lo stato dell'indicatore del livello di servizio (SLI). Gli SLI possono monitorare latenza, disponibilità e altri parametri operativi.
- CloudWatch Canari Synthetics: quando configuri il tracciamento a raggi X sui tuoi canarini, le chiamate ai tuoi servizi dagli script Canary vengono associate al servizio e visualizzate nella pagina dei dettagli del servizio.
- CloudWatch Monitoraggio degli utenti reali (RUM): quando il tracciamento X-Ray è abilitato sul client web CloudWatch RUM, le richieste ai servizi vengono automaticamente associate e visualizzate nella pagina dei dettagli del servizio.
- AWS Service Catalog AppRegistry— Application Signals rileva automaticamente AWS le risorse all'interno del tuo account e ti consente di raggrupparle in applicazioni logiche create in. AppRegistry Il nome dell'applicazione visualizzato nella pagina Servizi si basa sulla risorsa di calcolo sottostante su cui sono in esecuzione i servizi.

> **Note**
>
> Application Signals visualizza i servizi e le operazioni in base ai parametri e alle tracce emesse all'interno del filtro temporale corrente scelto. (Per impostazione predefinita, si tratta delle ultime tre ore). Se non è presente alcuna attività all'interno del filtro temporale corrente per un servizio, un'operazione, una dipendenza, canary Synthetics o pagina client, non sarà visibile.
> Attualmente è possibile visualizzare fino a 1.000 servizi. L'individuazione dei servizi e della topologia dei servizi potrebbe impiegare fino a 10 minuti. La valutazione dello stato dell'indicatore del livello di servizio (SLI) potrebbe impiegare fino a 15 minuti.

Visualizzazione generale dell'attività e dello stato operativo del servizio tramite la pagina Servizi

> Application Signals è in versione di anteprima per Amazon CloudWatch ed è soggetta a modifiche.

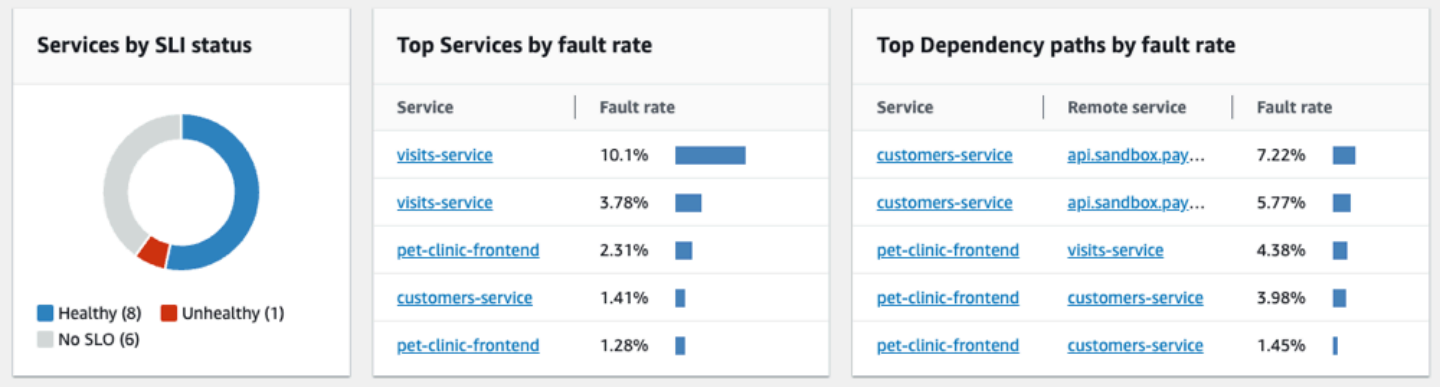
Utilizza la pagina Servizi per visualizzare un elenco dei tuoi servizi abilitati per Application Signals. Puoi anche vedere i parametri operativi e vedere rapidamente quali sono i servizi con indicatori del livello di servizio (SLI) non integri. Analizza le anomalie nelle prestazioni e identifica la causa principale dei problemi operativi. Per visualizzare questa pagina, apri la CloudWatch console e scegli Servizi nella sezione Application Signals nel riquadro di navigazione a sinistra.

Esplora i parametri di integrità operativa per i tuoi servizi

La parte superiore della pagina Servizi include un grafico dell'integrità operativa generale del servizio e diverse tabelle che mostrano i principali servizi e le dipendenze dei servizi in base alla percentuale di errore. Il grafico Servizi a sinistra mostra una suddivisione del numero di servizi con indicatori del livello di servizio (SLI) integri o non integri durante l'attuale filtro temporale a livello di pagina. Gli SLI possono monitorare latenza, disponibilità e altri parametri operativi.

Le due tabelle accanto al grafico mostrano un elenco dei principali servizi per frequenza di errore. Scegli un nome del servizio in entrambe le tabelle per aprire una pagina dei dettagli del servizio e visualizzare informazioni dettagliate sull'operazione del servizio. Scegli un percorso di dipendenza per aprire la pagina dei dettagli e visualizzare i dettagli sulla dipendenza del servizio. Entrambe le

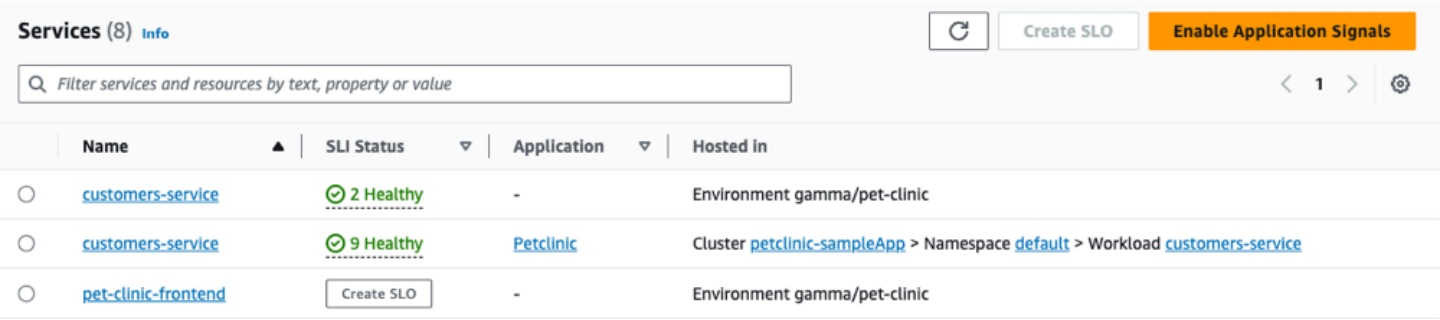
tabelle mostrano informazioni relative alle ultime tre ore, anche se viene scelto un filtro per periodi di tempo più lunghi in alto a destra della pagina.



Monitoraggio dell'integrità operativa con la tabella Servizi

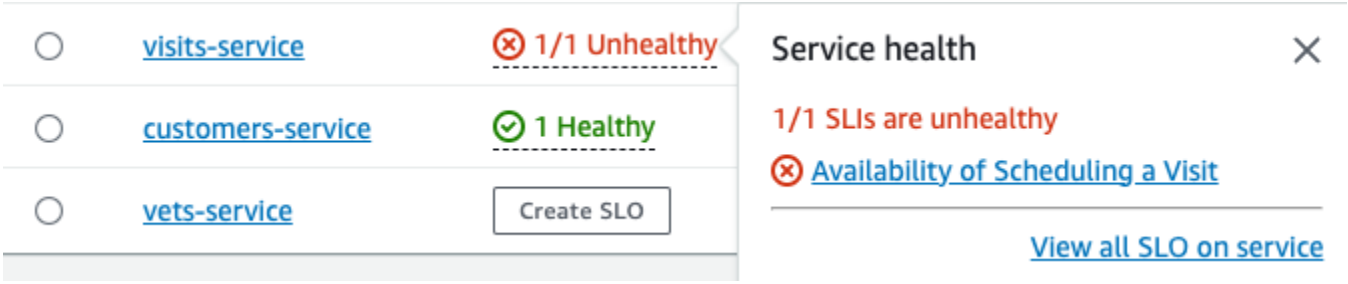
La pagina Servizi mostra un elenco dei tuoi servizi abilitati per Application Signals. Scegli Abilita Application Signals per aprire una pagina di configurazione e iniziare a configurare i servizi. Per ulteriori informazioni, consulta Abilitazione di Application Signals.

Filtra la tabella Servizi per trovare facilmente ciò che cerchi, scegliendo una o più proprietà dalla casella di testo del filtro. Quando scegli una proprietà, una procedura ti guida attraverso i criteri di filtro. Vedrai il filtro completo sotto la casella di testo del filtro. Seleziona Cancella filtri in qualsiasi momento per rimuovere il filtro della tabella.



Scegli il nome di qualsiasi servizio nella tabella per visualizzare una pagina dei dettagli del servizio contenente parametri a livello di servizio, operazioni e dettagli aggiuntivi. Se hai associato la risorsa di elaborazione sottostante del servizio a un'applicazione in AppRegistry o alla scheda Applicazioni nella AWS Management Console home page, scegli il nome dell'applicazione per visualizzare i dettagli dell'applicazione nella pagina della console MyApplications. Per i servizi ospitati in Amazon EKS, scegli qualsiasi link nella colonna Hosted in per visualizzare Cluster, Namespace o carico di lavoro all'interno CloudWatch di Container Insights. Per i servizi in esecuzione in Amazon ECS o Amazon EC2, viene mostrato il valore di ambiente.

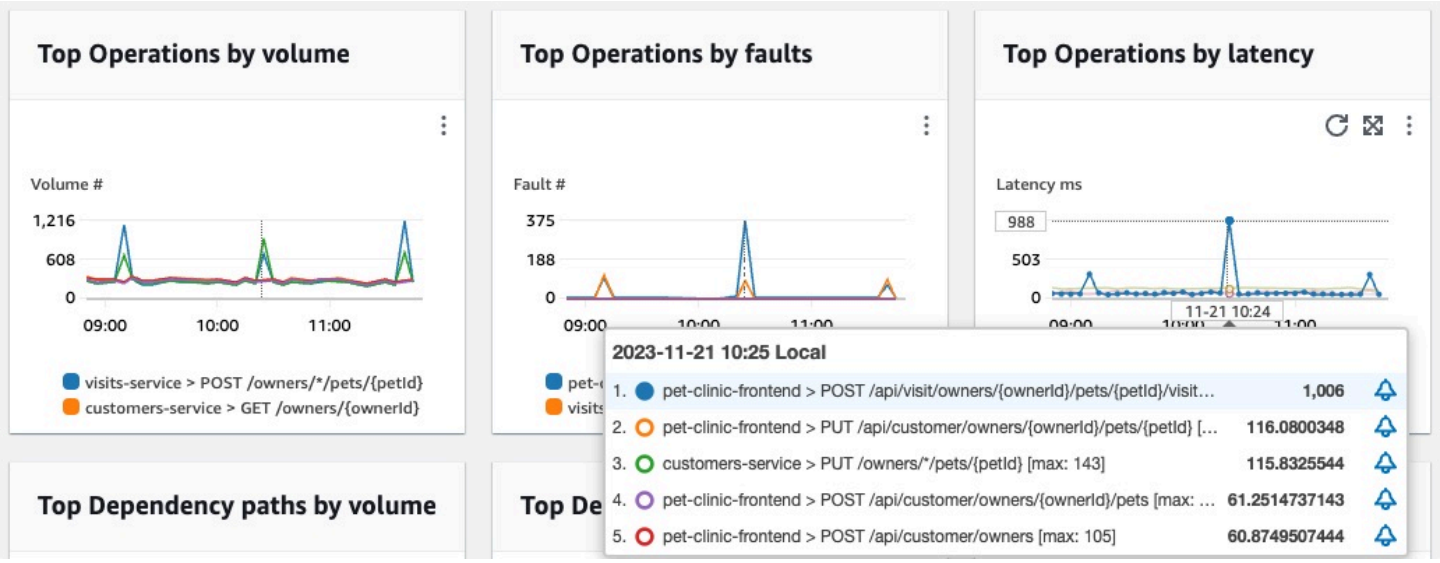
Lo stato dell'indicatore del livello di servizio (SLI) viene visualizzato per ogni servizio nella tabella. Scegli lo stato SLI di un servizio per visualizzare un pop-up contenente un collegamento a eventuali SLI non integri e un collegamento per visualizzare tutti gli SLO relativi al servizio.



Se non sono stati creati SLO per un servizio, seleziona il pulsante Crea SLO nella colonna Stato SLI. Per creare SLO aggiuntivi per qualsiasi servizio, seleziona il pulsante di opzione accanto al nome del servizio, quindi seleziona Crea SLO in alto a destra della tabella. Quando crei SLO, puoi vedere a colpo d'occhio quali dei tuoi servizi e operazioni funzionano correttamente e quali no. Consulta gli obiettivi del livello di servizio (SLO) per ulteriori informazioni.

Visualizza i parametri principali di operazione e dipendenza

Sotto la tabella Servizi, puoi visualizzare le operazioni e le dipendenze principali di tutti i servizi in base al volume delle chiamate, agli errori e alla latenza. Questo set di grafici fornisce informazioni fondamentali su quali operazioni o dipendenze potrebbero non essere integre in tutti i servizi. Scegli un punto qualsiasi in un grafico per visualizzare un pop-up contenente informazioni più dettagliate sulla serie. Passa il mouse sulle descrizioni delle serie nella parte inferiore di un grafico per visualizzare un pop-up contenente parametri dettagliati per un'operazione o un percorso di dipendenza specifico. Seleziona il pulsante del menu contestuale nell'angolo in alto a destra di un grafico per visualizzare opzioni aggiuntive, tra cui la visualizzazione di CloudWatch metriche o pagine di log.



Visualizza l'attività di servizio dettagliata e lo stato operativo con la pagina dei dettagli del servizio

> ⚠ Application Signals è in versione di anteprima per Amazon CloudWatch ed è soggetta a modifiche.

Quando strumentalizzi la tua CloudWatch applicazione, Amazon Application Signals mappa tutti i servizi rilevati dall'applicazione. Utilizza la pagina dei dettagli del servizio per visualizzare una panoramica dei tuoi servizi, operazioni, dipendenze, canarie e richieste dei clienti per un singolo servizio. Per visualizzare la pagina dei dettagli del servizio, procedi come segue:

- Apri la CloudWatch console.
- Scegli Servizi nella sezione Application Signals nel riquadro di navigazione a sinistra.
- Scegli il nome di qualsiasi servizio dalle tabelle Servizi, Servizi principali o dipendenze.

La pagina dei dettagli del servizio è organizzata nelle seguenti schede:

- Panoramica: utilizza questa scheda per visualizzare una panoramica di un singolo servizio, incluso il numero di operazioni, dipendenze, materiali sintetici e pagine client. La scheda mostra le metriche chiave per l'intero servizio, le operazioni principali e le dipendenze. Queste metriche includono dati di serie temporali su latenza, guasti ed errori in tutte le operazioni di servizio relative a quel servizio.

- Operazioni di servizio: utilizza questa scheda per visualizzare un elenco delle operazioni richiamate dal servizio e grafici interattivi con metriche chiave che misurano lo stato di ogni operazione. Puoi selezionare un punto dati in un grafico per ottenere informazioni su tracce, log o metriche associate a quel punto dati.
- Dipendenze: utilizza questa scheda per visualizzare un elenco di dipendenze richiamate dal servizio e un elenco di metriche per tali dipendenze.
- Canarini Synthetics: utilizza questa scheda per visualizzare un elenco di canarini sintetici che simulano le chiamate degli utenti al tuo servizio e le metriche chiave delle prestazioni relative a tali canarini.
- Pagine client: utilizza questa scheda per visualizzare un elenco di pagine client che richiamano il tuo servizio e metriche che misurano la qualità delle interazioni dei clienti con la tua applicazione.

Visualizza la panoramica del tuo servizio

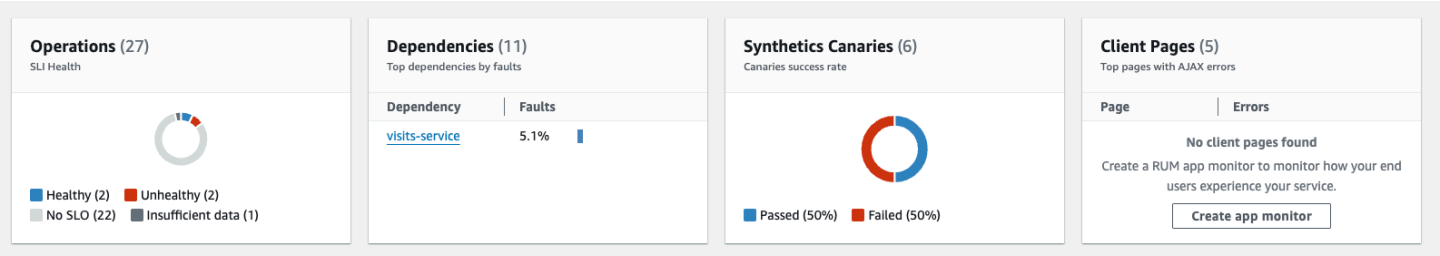
Utilizza la pagina di panoramica del servizio per visualizzare un riepilogo di alto livello delle metriche per tutte le operazioni di servizio in un'unica posizione. Controlla le prestazioni di tutte le operazioni, le dipendenze, le pagine client e i canali sintetici che interagiscono con la tua applicazione. Utilizzate queste informazioni per determinare dove concentrare gli sforzi per identificare i problemi, risolvere gli errori e trovare opportunità di ottimizzazione.

Scegliete un collegamento in Dettagli del servizio per visualizzare le informazioni relative a un servizio specifico. Ad esempio, per i servizi ospitati in Amazon EKS, la pagina dei dettagli del servizio mostra informazioni su cluster, namespace e carico di lavoro. Per i servizi ospitati in Amazon ECS o Amazon EC2, la pagina dei dettagli del servizio mostra il valore dell'ambiente.

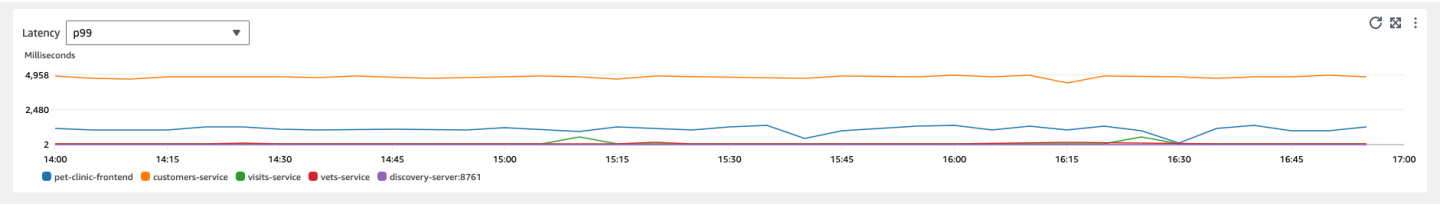
In Servizi, la scheda Panoramica mostra un riepilogo di quanto segue:

- Operazioni: utilizza questa scheda per visualizzare lo stato delle operazioni di servizio. Lo stato di salute è determinato dagli indicatori del livello di servizio (SLI) definiti come parte di un obiettivo del livello di servizio (SLO).
- Dipendenze: utilizza questa tabella per visualizzare le principali dipendenze dei servizi richiamati dall'applicazione, elencate in base alla percentuale di errore.
- Synthetics canaries: usa questa scheda per vedere il risultato delle chiamate simulate agli endpoint o alle API associate al tuo servizio e il numero di canarini falliti.
- Pagine client: utilizzate questa scheda per visualizzare le prime pagine richiamate dai client con errori JavaScript asincroni e XML (AJAX).

L'illustrazione seguente mostra una panoramica dei tuoi servizi:



La scheda Panoramica mostra anche un grafico delle dipendenze con la latenza più elevata tra tutti i servizi. Utilizza le metriche di latenza p99, p90 e p50 per valutare rapidamente quali dipendenze contribuiscono alla latenza totale del servizio, come segue:

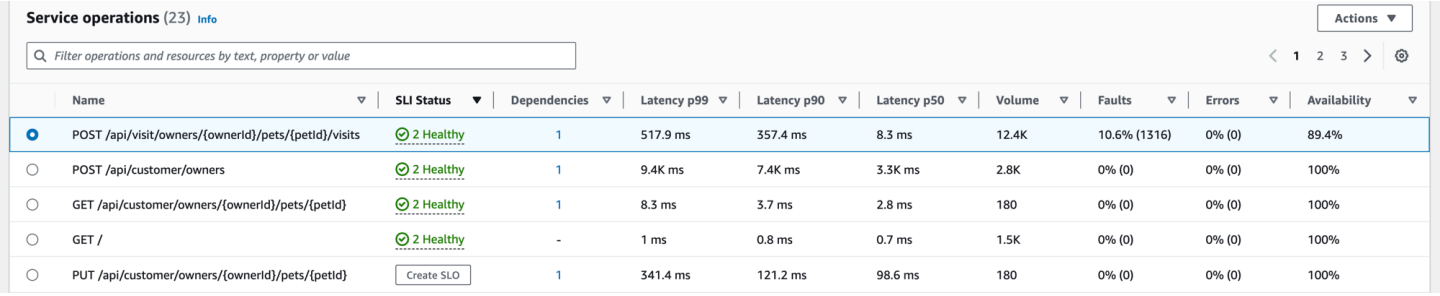


Ad esempio, il grafico precedente mostra che il 99% delle richieste fatte alla dipendenza dal servizio clienti sono state completate in circa 4.950 millisecondi. Le altre dipendenze hanno richiesto meno tempo.

I grafici che mostrano le prime quattro operazioni di servizio per latenza mostrano il volume di richieste, la disponibilità, la frequenza di errore e il tasso di errore per tali servizi, come mostrato nell'immagine seguente:

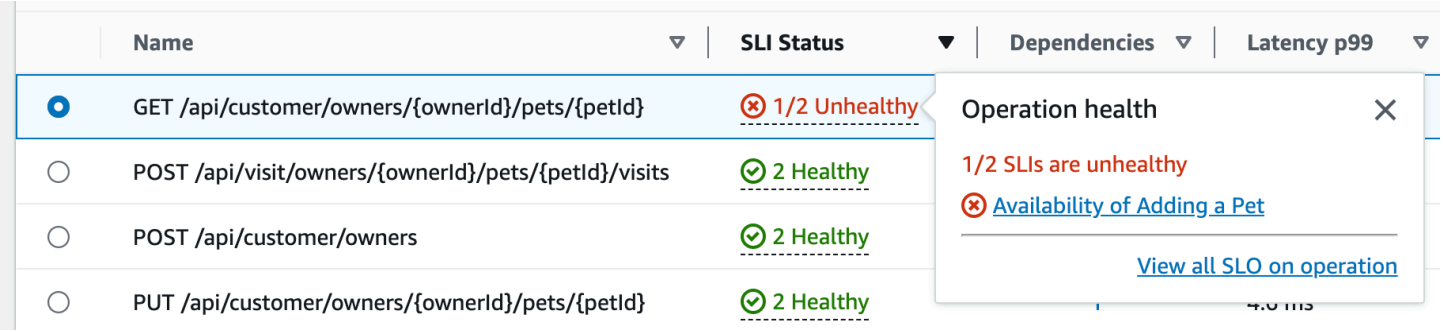
Visualizzazione delle operazioni del servizio

Quando si strumenta l'applicazione, Application Signals rileva tutte le operazioni di servizio richiamate dall'applicazione. Utilizza la scheda Operazioni di servizio per visualizzare una tabella che contiene le operazioni di servizio e un set di metriche che misurano le prestazioni di un'operazione selezionata. Queste metriche includono lo stato SLI, il numero di dipendenze, la latenza, il volume, gli errori, gli errori e la disponibilità, come mostrato nell'immagine seguente:



Filtra la tabella per facilitare la ricerca di un'operazione di servizio scegliendo una o più proprietà dalla casella di testo del filtro. Quando scegli una proprietà, una procedura ti guida attraverso i criteri di filtro e vedrai il filtro completo sotto la casella di testo del filtro. Seleziona Cancella filtri in qualsiasi momento per rimuovere il filtro della tabella.

Scegliete lo stato SLI di un'operazione per visualizzare un popup contenente un collegamento a qualsiasi SLI non funzionante e un collegamento per visualizzare tutti gli SLO relativi all'operazione, come mostrato nella tabella seguente:



La tabella delle operazioni di servizio elenca lo stato SLI, il numero di SLI integri o non integri e il numero totale di SLO per ogni operazione.

Utilizza gli SLI per monitorare la latenza, la disponibilità e altre metriche operative che misurano lo stato operativo di un servizio. Utilizza un SLO per verificare le prestazioni e lo stato di integrità dei tuoi servizi e delle tue operazioni.

Per creare uno SLO, procedi come segue:

- Se un'operazione non ha un SLO, scegliete il pulsante Crea SLO nella colonna Stato SLI.
- Se un'operazione ha già un SLO, procedi come segue:
  - Seleziona il pulsante radio accanto al nome dell'operazione.
  - Scegli Crea SLO dalla freccia rivolta verso il basso delle azioni in alto a destra della tabella.

Per ulteriori informazioni, consulta gli obiettivi del livello di servizio (SLO).

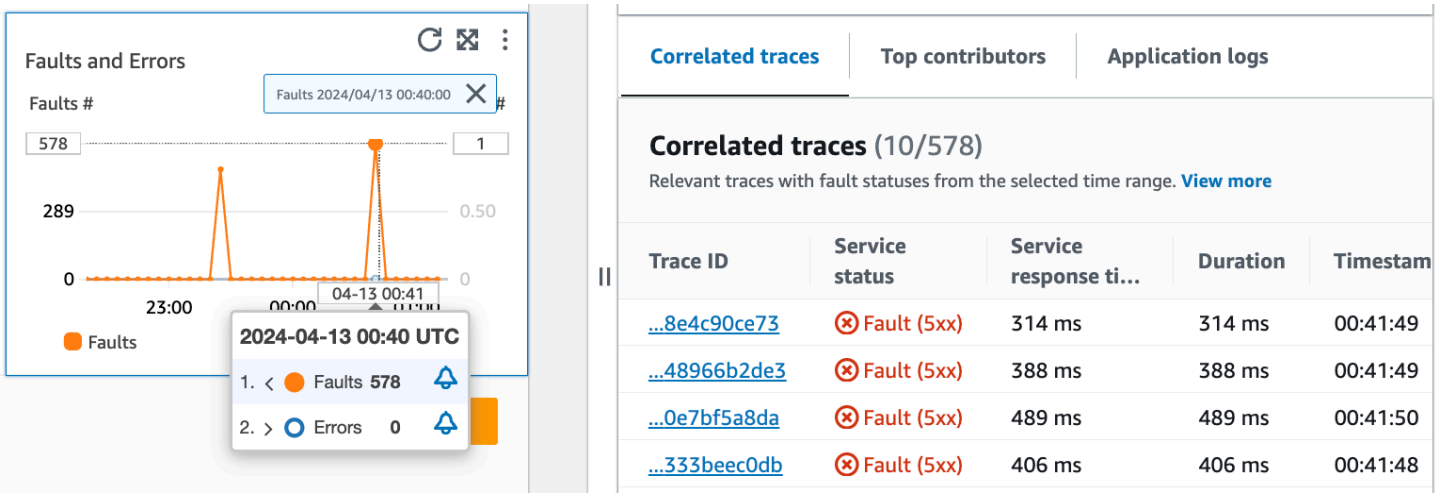
La colonna Dipendenze mostra il numero di dipendenze chiamate da questa operazione. Seleziona questo numero per aprire la scheda Dipendenze filtrata in base all'operazione selezionata.

Visualizza le metriche delle operazioni di servizio, le tracce correlate e i registri delle applicazioni

Application Signals mette in correlazione le metriche relative alle operazioni di servizio con AWS X-Ray trace, CloudWatch Container Insights e registri delle applicazioni. Utilizza queste metriche per risolvere i problemi di integrità operativa. Per visualizzare le metriche come informazioni grafiche, procedi come segue:

1. Seleziona un'operazione di servizio nella tabella Operazioni di servizio per visualizzare un set di grafici per l'operazione selezionata sopra la tabella con le metriche per Volume e disponibilità, Latenza e Guasti ed errori.
2. Passa il mouse su un punto in un grafico per visualizzare ulteriori informazioni.
3. Seleziona un punto per aprire un riquadro di diagnostica che mostra tracce, metriche e registri delle applicazioni correlati per il punto selezionato nel grafico.

L'immagine seguente mostra il tooltip che appare dopo aver passato il mouse su un punto del grafico e il riquadro di diagnostica che appare dopo aver fatto clic su un punto. Il tooltip contiene informazioni sul punto dati associato nel grafico Guasti ed errori. Il riquadro contiene le tracce correlate, i collaboratori principali e i registri delle applicazioni associati al punto selezionato.



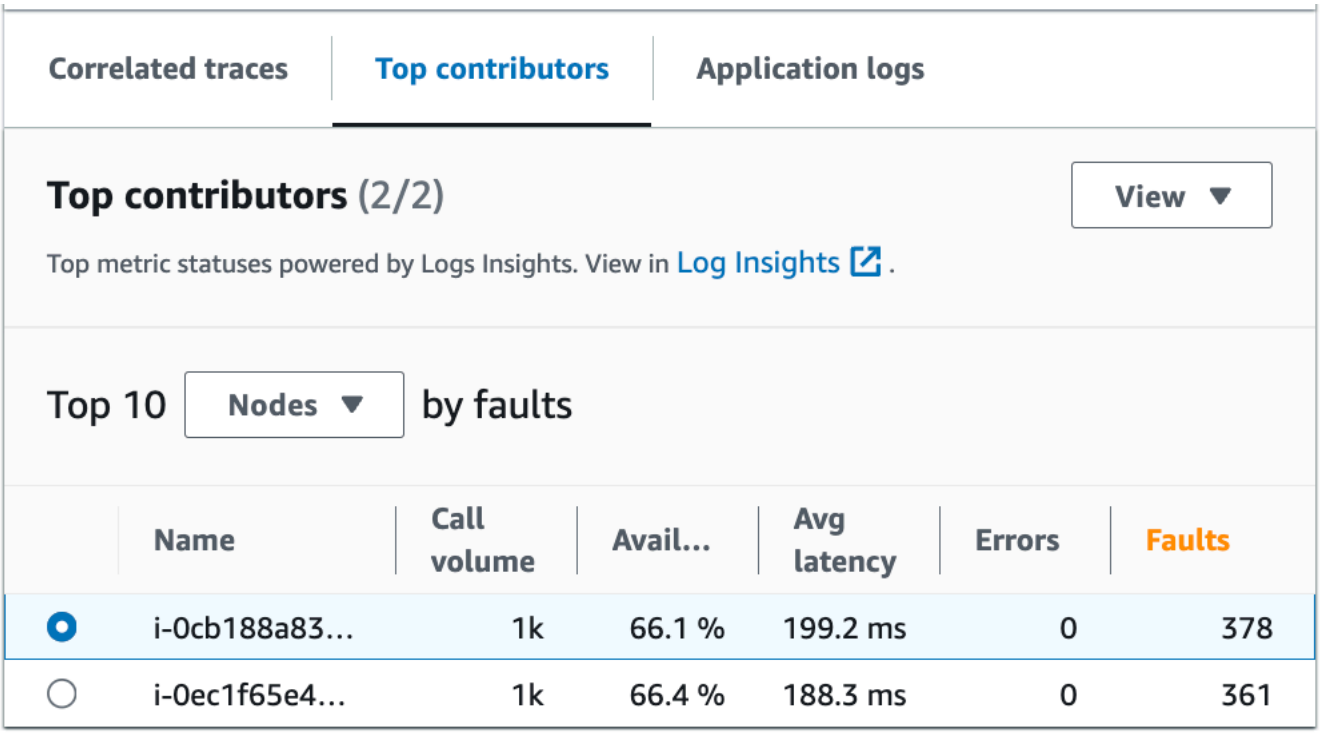
Tracce correlate

Guarda le tracce correlate per comprendere un problema di fondo con una traccia. Puoi verificare se le tracce correlate o gli eventuali nodi di servizio ad esse associati si comportano in modo simile. Per esaminare le tracce correlate, scegliete un ID di traccia dalla tabella Tracce correlate per aprire la pagina dei dettagli della traccia a raggi X per la traccia scelta. La pagina dei dettagli di traccia contiene una mappa dei nodi di servizio associati alla traccia selezionata e una sequenza temporale dei segmenti di traccia.

Collaboratori principali

Visualizza i principali collaboratori per trovare le principali fonti di input per una metrica. Raggruppa i contributori in base ai diversi componenti per cercare le somiglianze all'interno del gruppo e capire in che modo il comportamento di traccia differisce tra loro.

La scheda Collaboratori principali fornisce le metriche relative al volume delle chiamate, alla disponibilità, alla latenza media, agli errori e agli errori per ciascun gruppo. L'immagine di esempio seguente mostra i principali contributori a una suite di parametri per un'applicazione distribuita su una piattaforma Amazon EKS:



I contributori principali contengono le seguenti metriche:

- Volume delle chiamate: utilizza il volume delle chiamate per comprendere il numero di richieste per intervallo di tempo per un gruppo.
- Disponibilità: utilizza la disponibilità per vedere in quale percentuale di tempo non sono stati rilevati errori per un gruppo.
- Latenza media: utilizza la latenza per verificare il tempo medio di esecuzione delle richieste per un gruppo in un intervallo di tempo che dipende da quanto tempo sono state effettuate le richieste oggetto di indagine. Le richieste effettuate meno di 15 giorni prima vengono valutate a intervalli di 1 minuto. Le richieste effettuate tra i 15 e i 30 giorni precedenti, inclusi, vengono valutate a intervalli di 5 minuti. Ad esempio, se state esaminando le richieste che hanno causato un errore 15 giorni fa, la metrica del volume delle chiamate è uguale al numero di richieste per intervallo di 5 minuti.
- Errori: il numero di errori per gruppo misurato in un intervallo di tempo.
- Guasti: il numero di errori per gruppo in un intervallo di tempo.

Collaboratori principali che utilizzano Amazon EKS o Kubernetes

Utilizza le informazioni sui principali contributori per le applicazioni distribuite su Amazon EKS o Kubernetes per visualizzare i parametri di salute operativa raggruppati per Node, Pod e. PodTemplateHash Si applicano le seguenti definizioni:

- Un pod è un gruppo di uno o più Docker contenitori che condividono spazio di archiviazione e risorse. Un pod è l'unità più piccola che può essere implementata su una Kubernetes piattaforma. Raggruppa per pod per verificare se gli errori sono correlati a limitazioni specifiche dei pod.
- Un nodo è un server che esegue i pod. Raggruppa per nodi per verificare se gli errori sono correlati a limitazioni specifiche del nodo.
- Un hash del modello pod viene utilizzato per trovare una versione particolare di una distribuzione. Raggruppa per hash del modello pod per verificare se gli errori sono correlati a una particolare distribuzione.

Collaboratori principali che utilizzano Amazon EC2

Utilizza le informazioni sui principali contributori per le applicazioni distribuite su Amazon EKS per visualizzare i parametri di integrità operativa raggruppati per ID di istanza e gruppo di scalabilità automatica. Si applicano le seguenti definizioni:

- Un Instance ID è un identificatore univoco per l'istanza Amazon EC2 eseguita dal servizio. Raggruppa per ID istanza per verificare se gli errori sono correlati a un'istanza Amazon EC2 specifica.
- Un gruppo di auto scaling è una raccolta di istanze Amazon EC2 che consentono di aumentare o ridurre le risorse necessarie per soddisfare le richieste delle applicazioni. Raggruppa per gruppo con ridimensionamento automatico se desideri verificare se gli errori sono limitati alle istanze all'interno del gruppo.

Collaboratori principali che utilizzano una piattaforma personalizzata

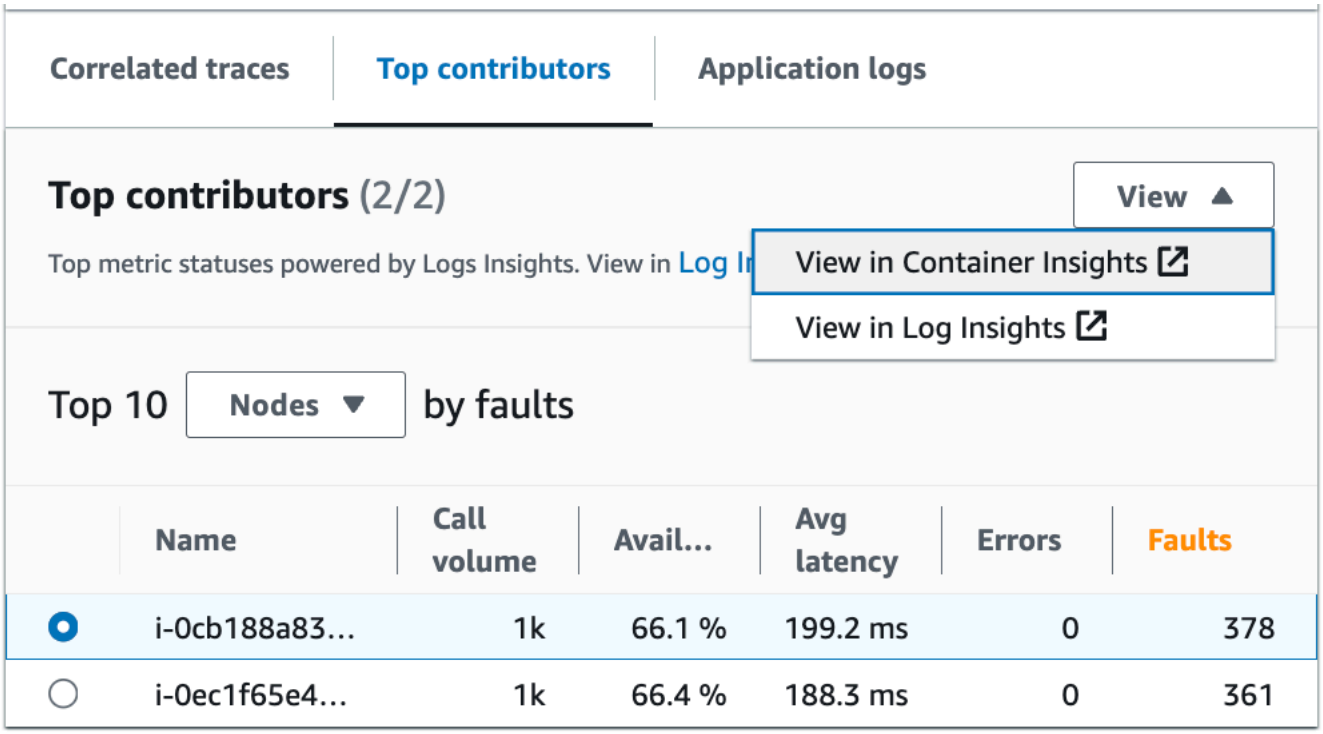
Utilizza le informazioni sui principali contributori per le applicazioni distribuite utilizzando strumentazione personalizzata per visualizzare le metriche sullo stato operativo raggruppate per nome host. Si applicano le seguenti definizioni:

- Un nome host identifica un dispositivo come un endpoint o un'istanza Amazon EC2 connesso a una rete. Raggruppa per nome host per verificare se gli errori sono correlati a uno specifico dispositivo fisico o virtuale.

Visualizza i principali collaboratori in e Log Insights Container Insights

Visualizza e modifica la query automatica che ha generato le metriche per i tuoi collaboratori principali in Log Insights. Visualizza le metriche delle prestazioni dell'infrastruttura per gruppi specifici

come pod o nodi in Container Insights. Puoi ordinare cluster, nodi o carichi di lavoro in base al consumo di risorse e identificare rapidamente le anomalie o mitigare i rischi in modo proattivo prima che l'esperienza dell'utente finale ne risenta. Segue un'immagine che mostra come selezionare queste opzioni:



In Container Insights, puoi visualizzare i parametri per il tuo contenitore Amazon EKS o Amazon ECS specifici per il raggruppamento dei tuoi collaboratori principali. Ad esempio, se raggruppi per pod per un contenitore EKS per generare i principali contributori, Container Insights mostrerà le metriche e le statistiche filtrate per il tuo pod.

In Log Insights, puoi modificare la query che ha generato le metriche in Top contributors utilizzando i seguenti passaggi:

1. Seleziona Visualizza in Log Insights. La pagina Logs Insights che si apre contiene una query generata automaticamente e contiene le seguenti informazioni:

   - Il nome del gruppo di cluster di log.
   - L'operazione su cui stavi indagando CloudWatch.
   - L'aggregato della metrica di salute operativa con cui si è interagito nel grafico.

   I risultati del registro vengono filtrati automaticamente per mostrare i dati degli ultimi cinque minuti prima della selezione del punto dati sul grafico del servizio.

2. Per modificare la query, sostituisci il testo generato con le tue modifiche. È inoltre possibile utilizzare il generatore di query per generare una nuova query o aggiornare l'interrogazione esistente.

Log di applicazioni

Utilizzate la query nella scheda Application logs per generare informazioni registrate per il gruppo di log corrente, il servizio e inserire un timestamp. Un gruppo di log è un gruppo di flussi di log che è possibile definire quando si configura l'applicazione.

Utilizzate un gruppo di log per organizzare i log con caratteristiche simili, tra cui:

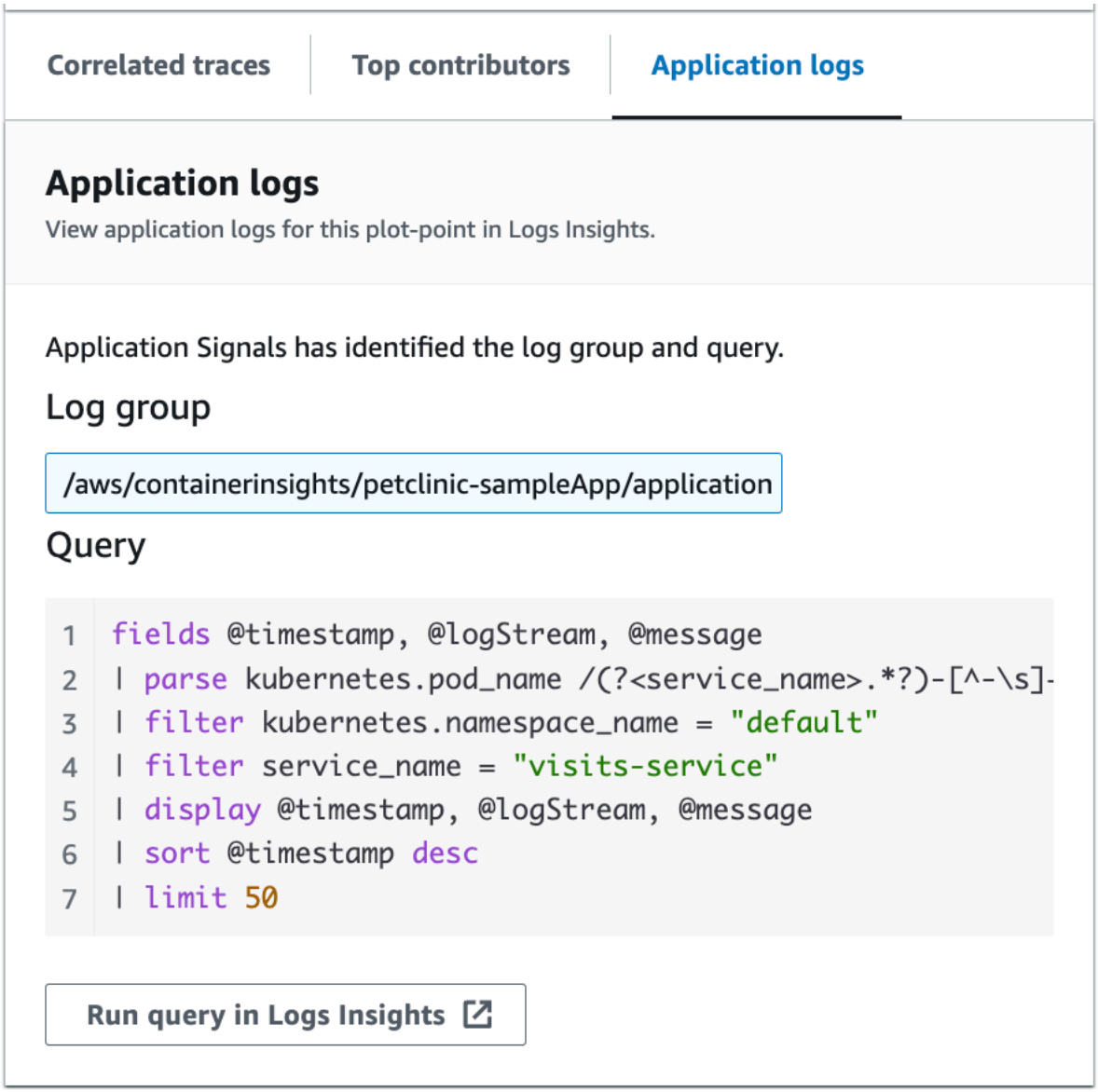
- Acquisisci i log da un'organizzazione, una fonte o una funzione specifica.
- Acquisisci i log a cui accede un determinato utente.
- Acquisisci i registri per un periodo di tempo specifico.

Usa questi flussi di log per tenere traccia di gruppi o intervalli di tempo specifici. Puoi anche impostare regole di monitoraggio, allarmi e notifiche per questi gruppi di log. Per ulteriori informazioni sui gruppi di log, consulta Lavorare con gruppi di log e flussi di log.

La query dei log dell'applicazione restituisce i log, i modelli di testo ricorrenti e le visualizzazioni grafiche per i gruppi di log.

Per eseguire la query, selezionare Esegui query in Logs Insights per eseguire la query generata automaticamente o modificare la query. Per modificare la query, sostituisci il testo generato automaticamente con le tue modifiche. È inoltre possibile utilizzare il generatore di query per generare una nuova query o aggiornare l'interrogazione esistente.

L'immagine seguente mostra l'interrogazione di esempio che viene generata automaticamente in base al punto selezionato nel grafico delle operazioni di servizio:



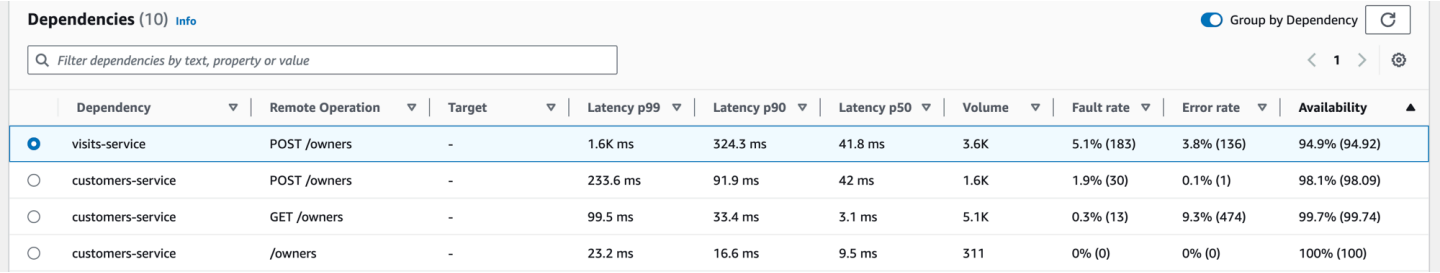
Nell'immagine precedente, CloudWatch ha rilevato automaticamente il gruppo di log associato al punto selezionato e lo ha incluso in una query generata.

Visualizzazione delle dipendenze del servizio

Scegli la scheda Dipendenze per visualizzare la tabella Dipendenze e un insieme di parametri per le dipendenze di tutte le operazioni del servizio o di una singola operazione. La tabella contiene un elenco di dipendenze individuate da Application Signals, tra cui parametri relativi a latenza, volume delle chiamate, frequenza di guasto, frequenza di errore e disponibilità.

Nella parte superiore della pagina, scegliete un'operazione dall'elenco delle frecce in basso per visualizzarne le dipendenze oppure scegliete Tutto per visualizzare le dipendenze per tutte le operazioni.

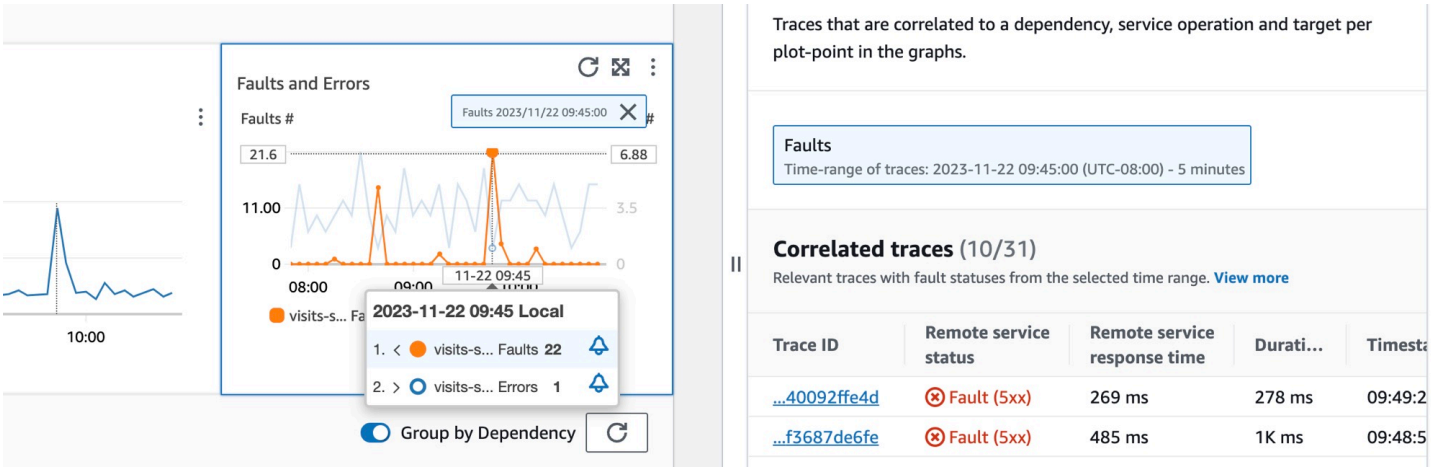
Filtra la tabella per trovare facilmente ciò che cerchi, scegliendo una o più proprietà dalla casella di testo del filtro. Quando scegli una proprietà, una procedura ti guida attraverso i criteri di filtro e vedrai il filtro completo sotto la casella di testo del filtro. Seleziona Cancella filtri in qualsiasi momento per rimuovere il filtro della tabella. Seleziona Raggruppa per dipendenza in alto a destra della tabella per raggruppare le dipendenze in base al nome del servizio e dell'operazione. Quando il raggruppamento è attivo, puoi espandere o comprimere un gruppo di dipendenze con l'icona + accanto al nome della dipendenza.



Dependency	Remote Operation	Target	Latency p99	Latency p90	Latency p50	Volume	Fault rate	Error rate	Availability
visits-service	POST /owners	-	1.6K ms	324.3 ms	41.8 ms	3.6K	5.1% (183)	3.8% (136)	94.9% (94.92)
customers-service	POST /owners	-	233.6 ms	91.9 ms	42 ms	1.6K	1.9% (30)	0.1% (1)	98.1% (98.09)
customers-service	GET /owners	-	99.5 ms	33.4 ms	3.1 ms	5.1K	0.3% (13)	9.3% (474)	99.7% (99.74)
customers-service	/owners	-	23.2 ms	16.6 ms	9.5 ms	311	0% (0)	0% (0)	100% (100)

La colonna Dipendenza mostra il nome del servizio di dipendenza, mentre la colonna Operazione remota mostra il nome dell'operazione del servizio. Quando si chiamano AWS i servizi, la colonna Target mostra la AWS risorsa, ad esempio la tabella DynamoDB o la coda Amazon SNS.

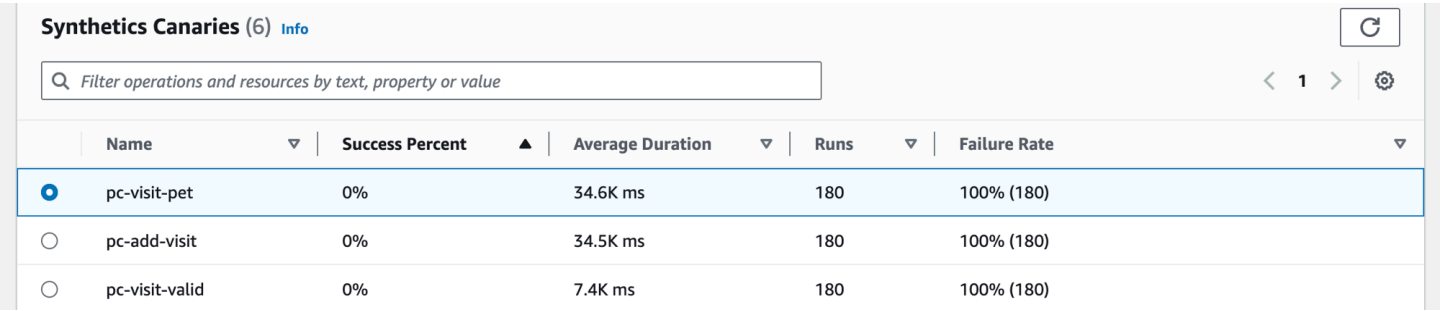
Per selezionare una dipendenza, seleziona l'opzione accanto a una dipendenza nella tabella Dipendenze. Questo mostra una serie di grafici che mostrano metriche dettagliate per il volume delle chiamate, la disponibilità, i guasti e gli errori. Passa il mouse su un punto di un grafico per visualizzare un popup contenente ulteriori informazioni. Seleziona un punto in un grafico per aprire un riquadro di diagnostica che mostra le tracce correlate per il punto selezionato nel grafico. Scegliete un ID di traccia dalla tabella Tracce correlate per aprire la pagina dei dettagli di X-Ray Trace per la traccia selezionata.



Visualizzazione dei canary Synthetics

Scegli la scheda Canary Synthetics per visualizzare la tabella Canary Synthetics e un set di tabelle per ogni canary presente nella tabella. La tabella include i parametri relativi alla percentuale di successo, alla durata media, alle esecuzioni e alla frequenza di errore. Vengono visualizzati solo i canarini abilitati per il AWS X-Ray tracciamento.

Usa la casella di testo del filtro nella tabella dei canarini sintetici per trovare il canarino che ti interessa. Ogni filtro creato viene visualizzato sotto la casella di testo del filtro. Seleziona Cancella filtri in qualsiasi momento per rimuovere il filtro della tabella.



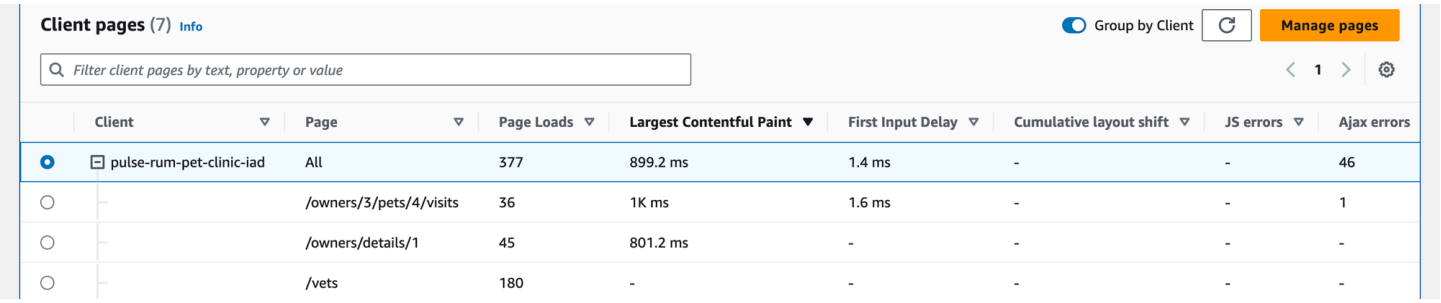
Seleziona il pulsante di opzione accanto al nome del canarino per visualizzare una serie di schede contenenti grafici, metriche dettagliate tra cui percentuale di successo, errori e durata. Passa il mouse su un punto di un grafico per visualizzare un popup contenente ulteriori informazioni. Seleziona un punto in un grafico per aprire un riquadro di diagnostica che mostra le corse dei canarini correlate al punto selezionato. Seleziona una corsa canaria e scegli la durata per visualizzare gli elementi relativi alla corsa canaria selezionata, tra cui registri, file di HTTP archivio (HAR), schermate e passaggi suggeriti per aiutarti a risolvere i problemi. Scegli Ulteriori informazioni per aprire la pagina CloudWatch Synthetics Canaries accanto a Canary run.

Visualizzazione delle pagine client

Scegli la scheda Pagine client per visualizzare un elenco di pagine Web client che richiamano il tuo servizio. Utilizza il set di metriche per la pagina client selezionata per misurare la qualità dell'esperienza del cliente quando interagisce con un servizio o un'applicazione. Queste metriche includono caricamenti di pagine, dati vitali web ed errori.

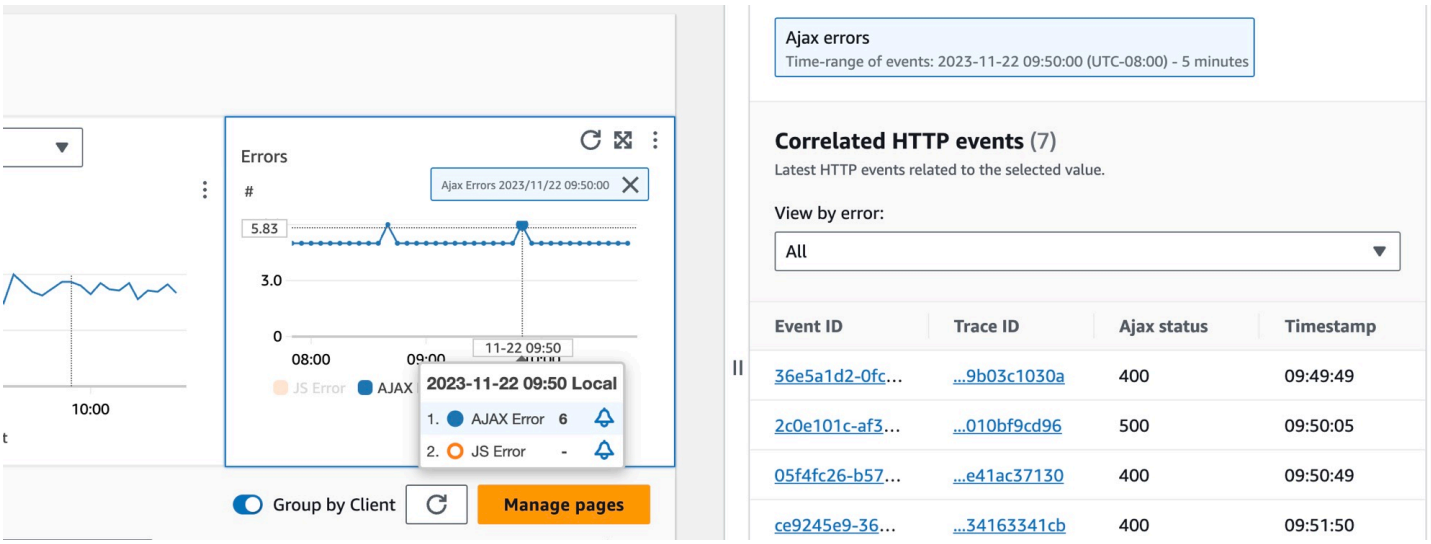
Per visualizzare le pagine client nella tabella, è necessario configurare il client Web CloudWatch RUM per il tracciamento a raggi X e attivare le metriche di Application Signals per le pagine client. Scegli Gestisci pagine per selezionare quali pagine sono abilitate per le metriche di Application Signals.

Utilizza la casella di testo del filtro per trovare la pagina client o il monitor dell'applicazione che ti interessa sotto la casella di testo del filtro. Scegli Cancella filtri per rimuovere il filtro della tabella. Seleziona Raggruppa per client per raggruppare le pagine client per client. Una volta raggruppate, seleziona l'icona + accanto al nome di un client per espandere la riga e visualizzare tutte le pagine relative a quel client.



Per selezionare una pagina client, selezionare l'opzione accanto a una pagina client nella tabella Pagine client. Verrà visualizzata una serie di grafici che mostrano parametri dettagliati. Passa il mouse su un punto di un grafico per visualizzare un popup contenente ulteriori informazioni. Seleziona un punto in un grafico per aprire un riquadro di diagnostica che mostra gli eventi di navigazione relativi alle prestazioni correlati per il punto selezionato nel grafico. Scegliete un ID

evento dall'elenco degli eventi di navigazione per aprire la visualizzazione della pagina CloudWatch RUM per l'evento scelto.



> **Note**
>
> Per visualizzare gli errori AJAX nelle pagine client, utilizzate la versione 1.15 o successiva del client web CloudWatch RUM.
> Attualmente è possibile visualizzare fino a 100 operazioni, canary e pagine client e fino a 250 dipendenze per servizio.

Visualizza la topologia dell'applicazione e monitora lo stato operativo con la mappa dei CloudWatch servizi

> **Application Signals è in versione di anteprima per Amazon CloudWatch ed è soggetta a modifiche.**

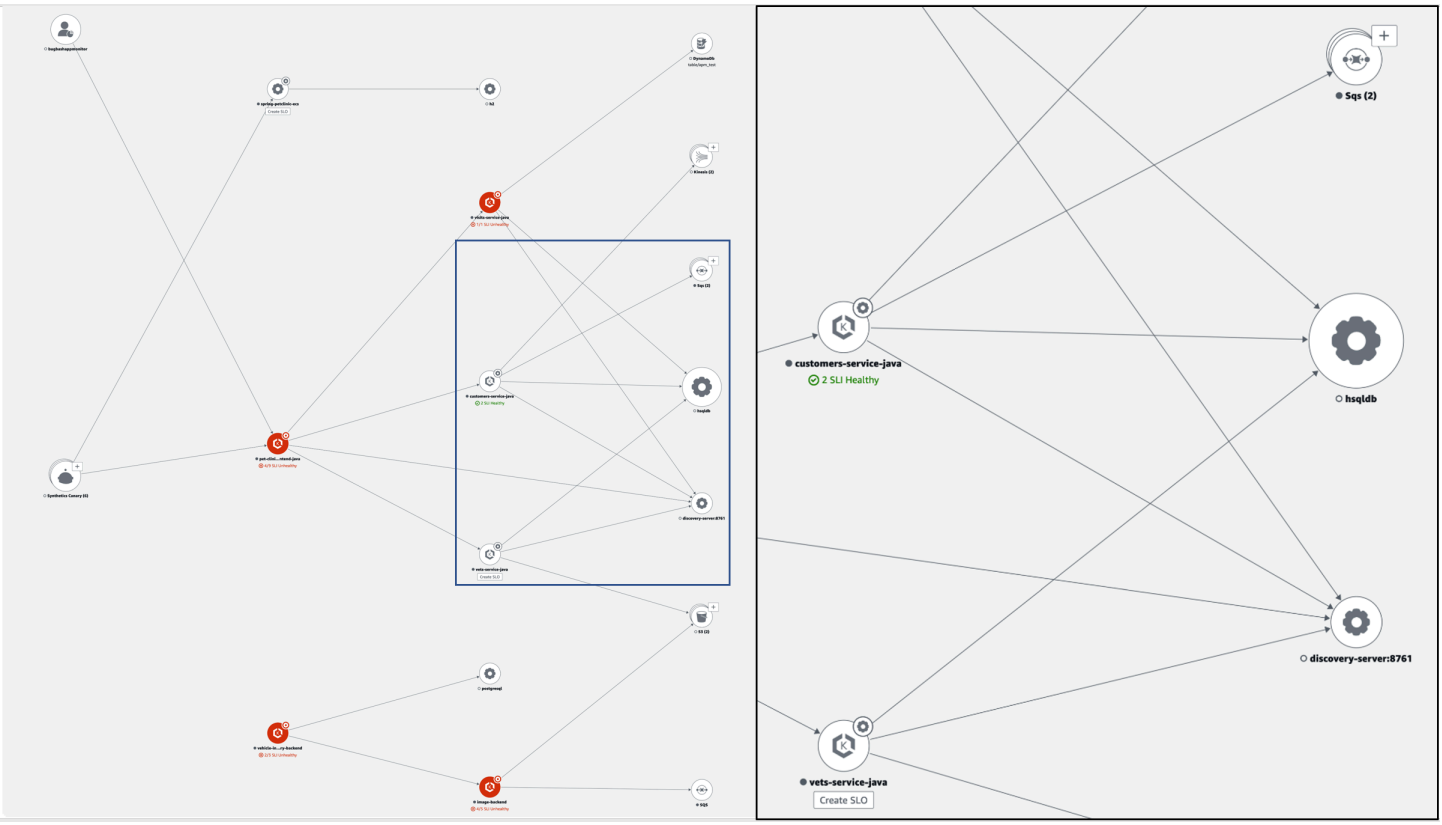
> **Note**
>
> La mappa dei CloudWatch servizi sostituisce la ServiceLens mappa. Per visualizzare una mappa dell'applicazione basata sulle AWS X-Ray tracce, apri la X-Ray Trace Map. Scegli Trace Map nella sezione X-Ray nel riquadro di navigazione a sinistra della CloudWatch console.

Utilizza la mappa dei servizi per visualizzare la topologia dei client delle applicazioni, dei canali sintetici, dei servizi e delle dipendenze e monitorare lo stato operativo. Per visualizzare la mappa dei servizi, apri la CloudWatch console e scegli Service Map nella sezione Application Signals nel riquadro di navigazione a sinistra.

Dopo aver abilitato l'applicazione per Application Signals, utilizza la mappa dei servizi per semplificare il monitoraggio dello stato operativo dell'applicazione:

- Visualizza le connessioni tra client, canary, servizio e nodi di dipendenza per comprendere la topologia e il flusso di esecuzione dell'applicazione. Ciò è particolarmente utile se gli operatori del servizio non fanno parte del team di sviluppo.
- Scopri quali servizi soddisfano o meno i tuoi obiettivi del livello di servizio (SLO). Quando un servizio non soddisfa i tuoi SLO, puoi capire rapidamente se un servizio a valle o una dipendenza potrebbero contribuire al problema o influire su più servizi upstream.
- Seleziona un singolo client, synthetics canary, servizio o nodo di dipendenza per visualizzare le metriche correlate. La pagina dei dettagli del servizio mostra informazioni più dettagliate su operazioni, dipendenze, synthetics canaries e pagine client.
- Filtra e ingrandisci la mappa dei servizi per concentrarti più facilmente su una parte della topologia dell'applicazione o visualizza l'intera mappa. Crea un filtro scegliendo una o più proprietà dalla casella di testo del filtro. Quando scegli una proprietà, una procedura ti guida attraverso i criteri di filtro. Vedrai il filtro completo sotto la casella di testo del filtro. Seleziona Cancella filtri in qualsiasi momento per rimuovere il filtro.

L'esempio seguente di mappa dei servizi mostra i servizi con bordi che li collegano ai componenti con cui interagiscono. Se viene definito un SLO, la mappa dei servizi mostra anche lo stato di salute.

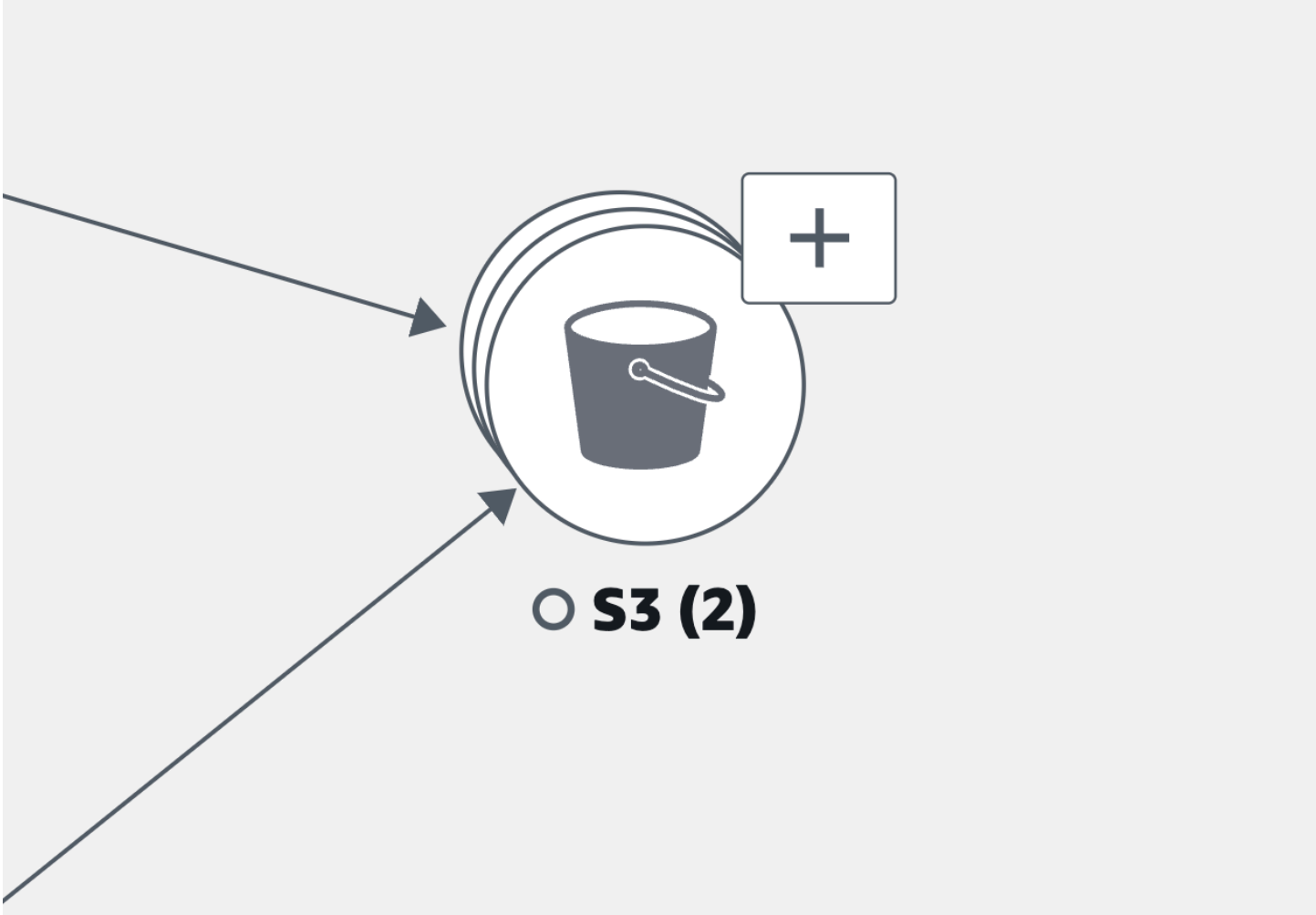


Esplora la mappa dei servizi

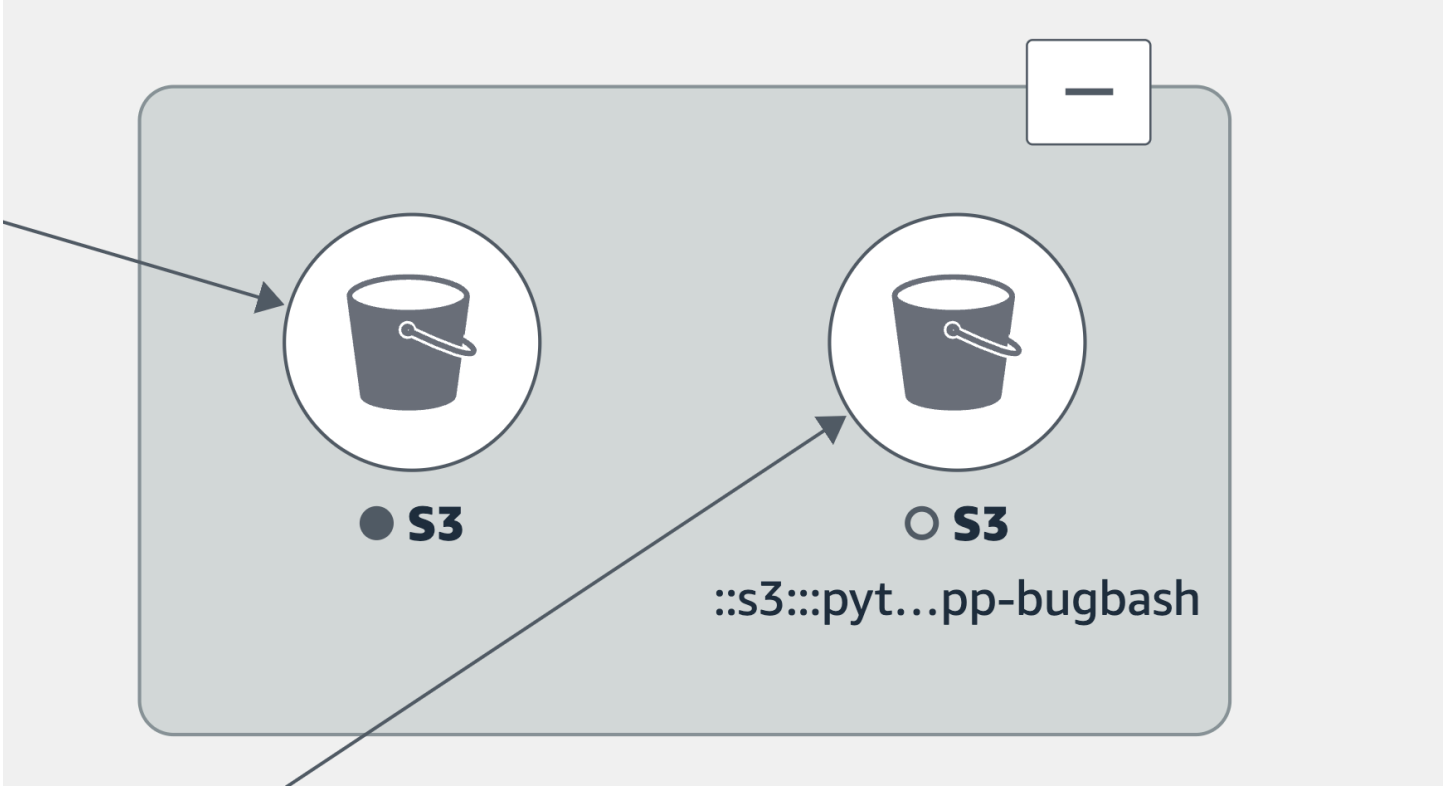
Dopo aver abilitato l'applicazione per Application Signals, la mappa dei servizi mostra i nodi che rappresentano i tuoi servizi e le loro dipendenze.

Attiva il tracciamento attivo per i tuoi client CloudWatch RUM e synthetics canaries per vedere i nodi client e canary sulla mappa.

Per impostazione predefinita, i canaries, i client RUM e le dipendenze di AWS servizio dello stesso tipo sono raggruppati in un'unica icona espandibile nella mappa dei servizi. Per impostazione predefinita, le dipendenze dei servizi esterne a non AWS sono raggruppate insieme. Ad esempio, nell'immagine seguente, tutti i bucket Amazon S3 sono raggruppati sotto un'unica icona espandibile:



Nell'immagine precedente, l'etichetta tra il servizio di raggruppamento Amazon S3 e quello di origine mostra il numero di bordi del gruppo tra parentesi sotto l'icona della dipendenza. Seleziona l'icona (+) per espandere il gruppo e visualizzarne i singoli elementi, come mostrato nell'immagine seguente:

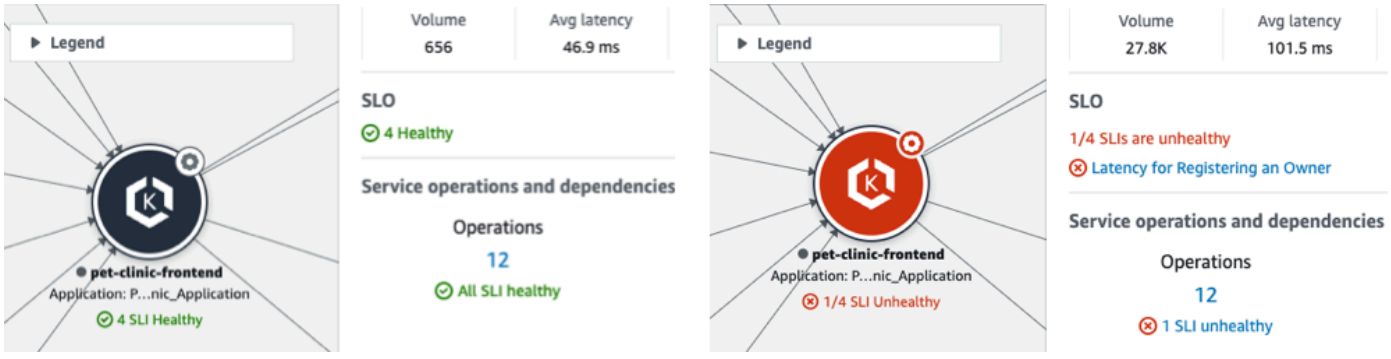


Scegli una scheda per informazioni sull'esplorazione di ogni tipo di nodo e dei bordi (connessioni) tra di essi.

View your application services

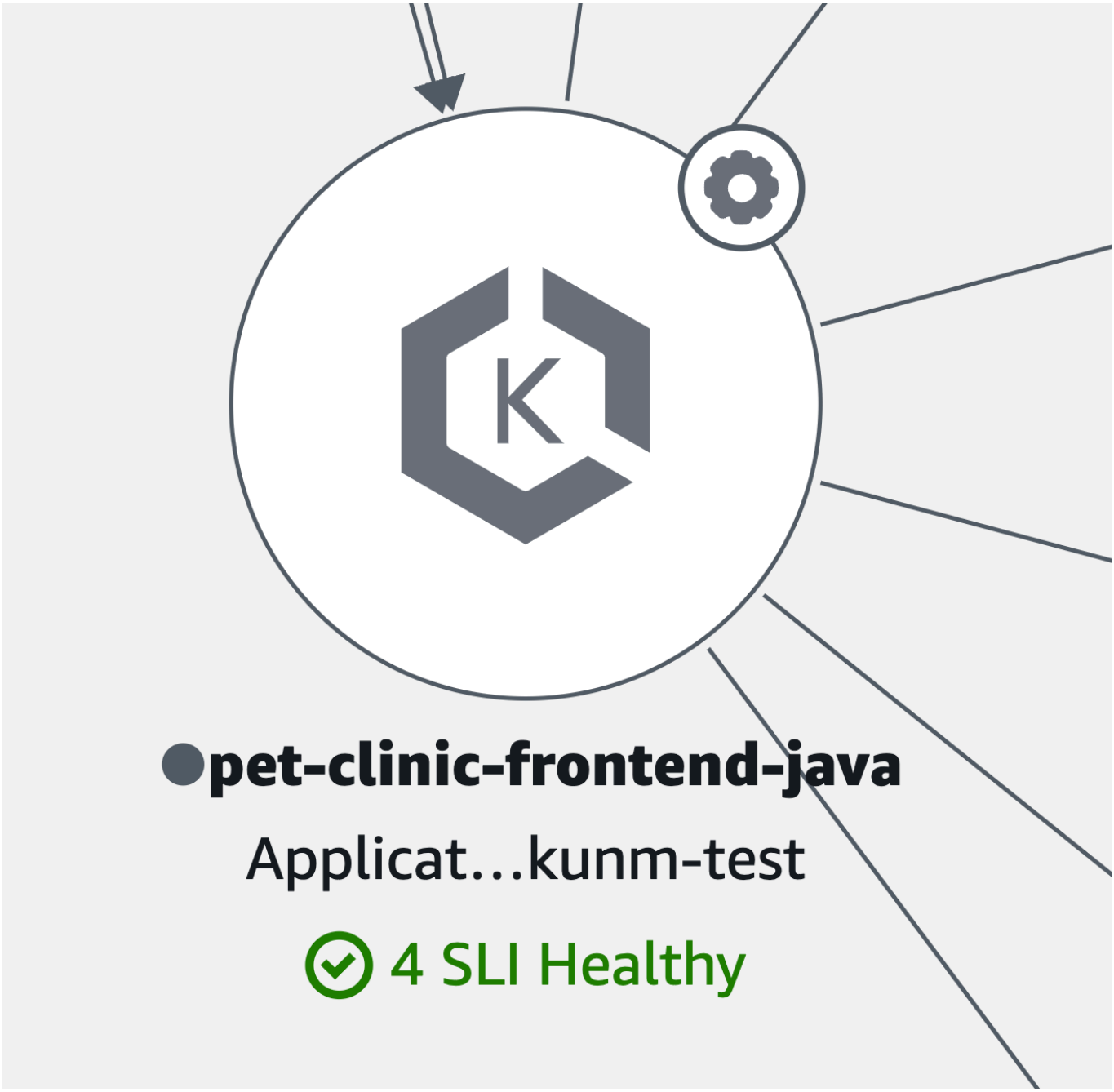
È possibile visualizzare i servizi applicativi e lo stato dei relativi SLO e degli indicatori di livello di servizio (SLI) nella Service Map. Se non hai creato SLO per un servizio, scegli il pulsante Crea SLO sotto il nodo del servizio.

La mappa dei servizi mostra tutti i tuoi servizi. Mostra anche i clienti e i canarini che utilizzano il servizio e le dipendenze richiamate dai servizi, come mostrato nell'immagine seguente:



Le icone seguenti rappresentano esempi di servizi applicativi nella mappa dei servizi:

- Servizio Amazon Elastic Kubernetes:



- Un contenitore Kubernetes:

- Amazon Elastic Compute Cloud (Amazon EC2):

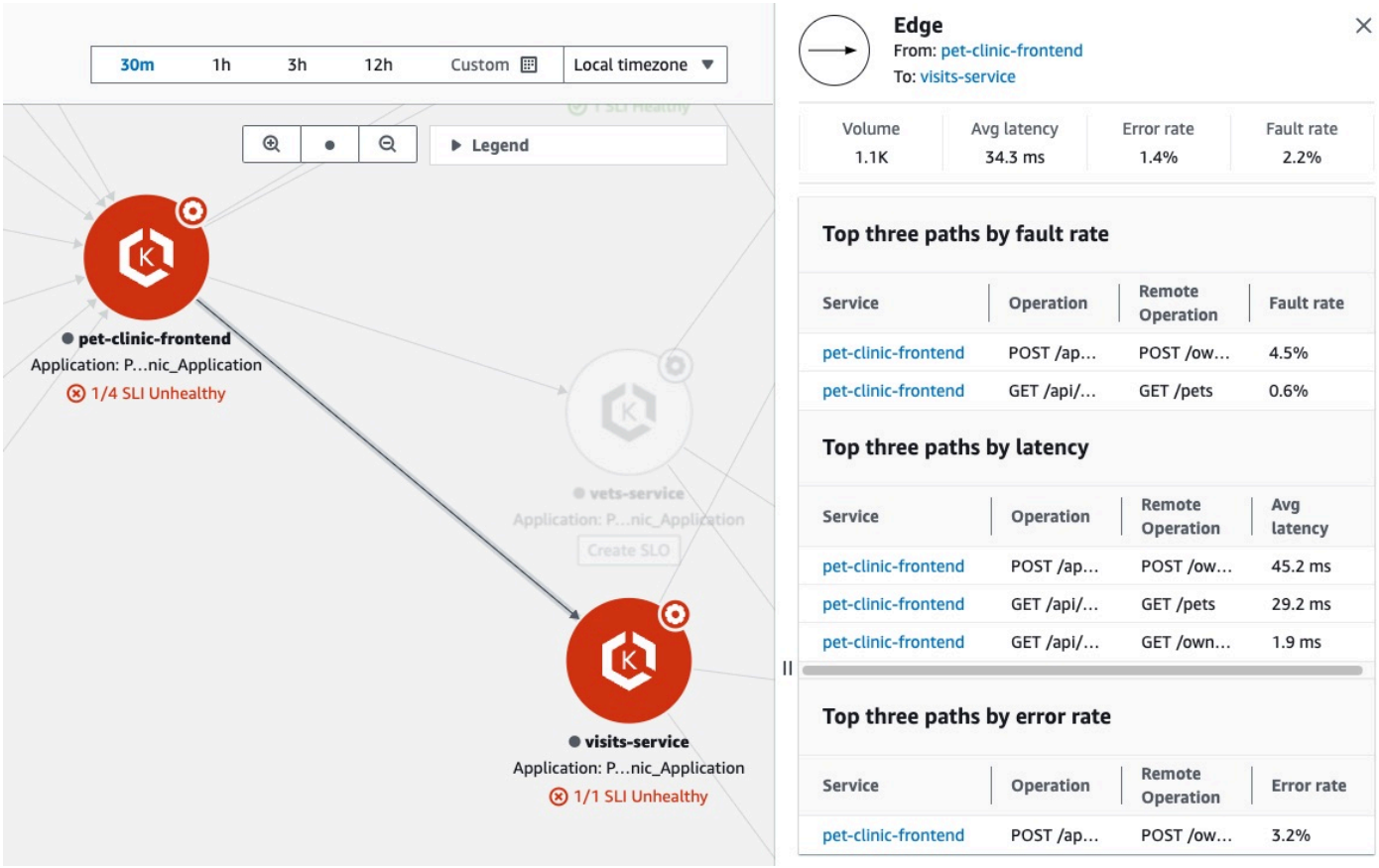
- Altri tipi di servizi applicativi non elencati in precedenza:

Quando si seleziona un nodo di servizio, si apre un riquadro con informazioni dettagliate sul servizio:

- Parametri relativi al volume delle chiamate, alla latenza, agli errori e alla frequenza di errore.
- Il numero di SLI e SLO che sono `healthy` o. `unhealthy`
- L'opzione per visualizzare ulteriori informazioni su uno SLO.
- Il numero di operazioni di servizio, dipendenze, canali sintetici e pagine client.
- L'opzione per selezionare ogni numero per aprire la relativa pagina dei dettagli del servizio.
- Il nome dell'applicazione, se la risorsa di elaborazione sottostante è stata associata a un'applicazione utilizzando AppRegistry o la scheda Applicazioni nella AWS Management Console home page.

- Scegli il nome dell'applicazione per visualizzare i dettagli dell'applicazione nella pagina della console myApplications.
- Il `ClusterNamespace`, e `Workload` per i servizi ospitati in Amazon EKS o `Environment` per i servizi ospitati in Amazon ECS o Amazon EC2. Per i servizi ospitati su Amazon EKS, scegli un link qualsiasi per aprire CloudWatch Container Insights.

Seleziona un edge o una connessione tra un nodo di servizio e un nodo di servizio o di dipendenza a valle. Si apre un riquadro contenente i percorsi principali per frequenza di errore, latenza e frequenza di errore, come mostrato nella seguente immagine di esempio. Scegli un link qualsiasi nel riquadro per aprire la pagina dei dettagli del servizio e visualizzare informazioni dettagliate sul servizio o sulla dipendenza scelti.

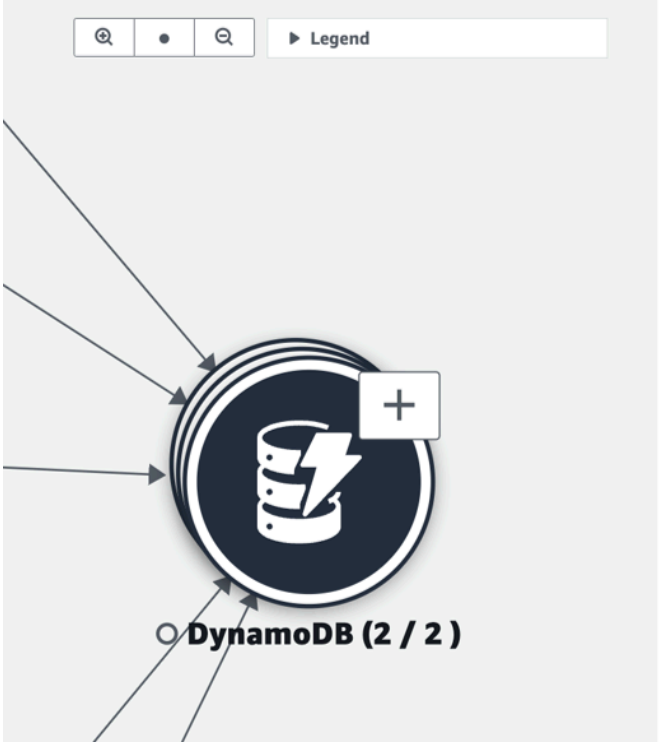


View dependencies

Le dipendenze delle applicazioni vengono visualizzate sulla mappa dei servizi, connesse ai servizi che le chiamano.

Scegliete un nodo di dipendenza per aprire un riquadro contenente i percorsi principali per frequenza di errore, latenza e frequenza di errore. Scegliete un collegamento al servizio o alla

destinazione per aprire la pagina Dettagli del servizio e visualizzare informazioni dettagliate sul servizio o sull'obiettivo di dipendenza scelto, come mostrato nell'immagine di esempio seguente:



Volume	Avg latency	Error rate	Fault rate
-	-	-	-

Top three paths by fault rate

Service	Remote operation	Fault rate
No paths with faults		

Top three paths by latency

Service	Remote operation	Avg latency
billing-service-ec2-python	PutItem	282.8 ms
billing-service-python	PutItem	75.6 ms
visits-service-java	PutItem	64.9 ms

Top three paths by error rate

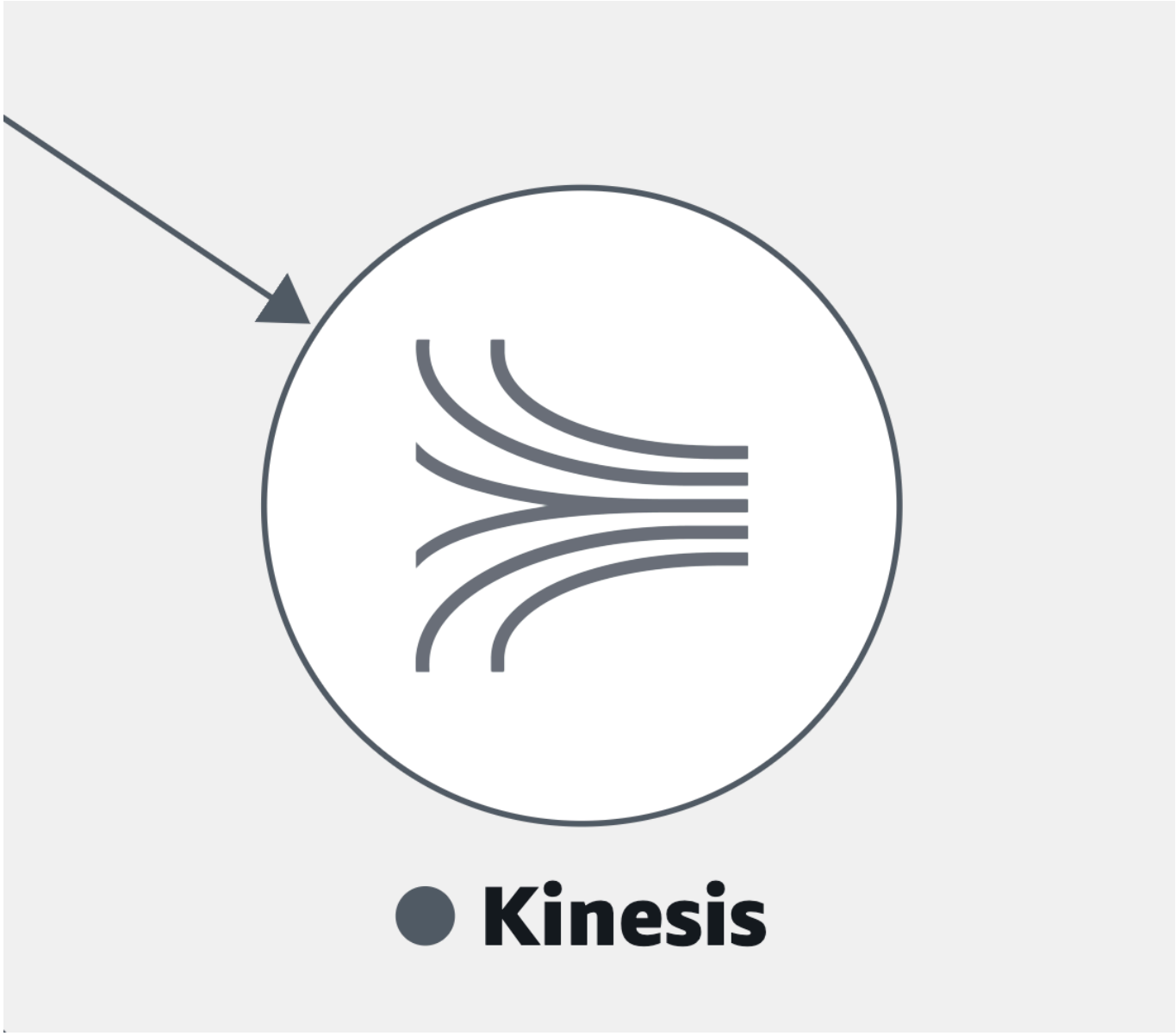
Service	Remote operation	Error rate
visits-service-java	PutItem	9.6%

Le dipendenze dei servizi sono raggruppate per impostazione predefinita in un'unica icona espandibile. Selezionate l'icona (+), come mostrato nell'immagine precedente, per espandere il gruppo e visualizzarne i singoli elementi.

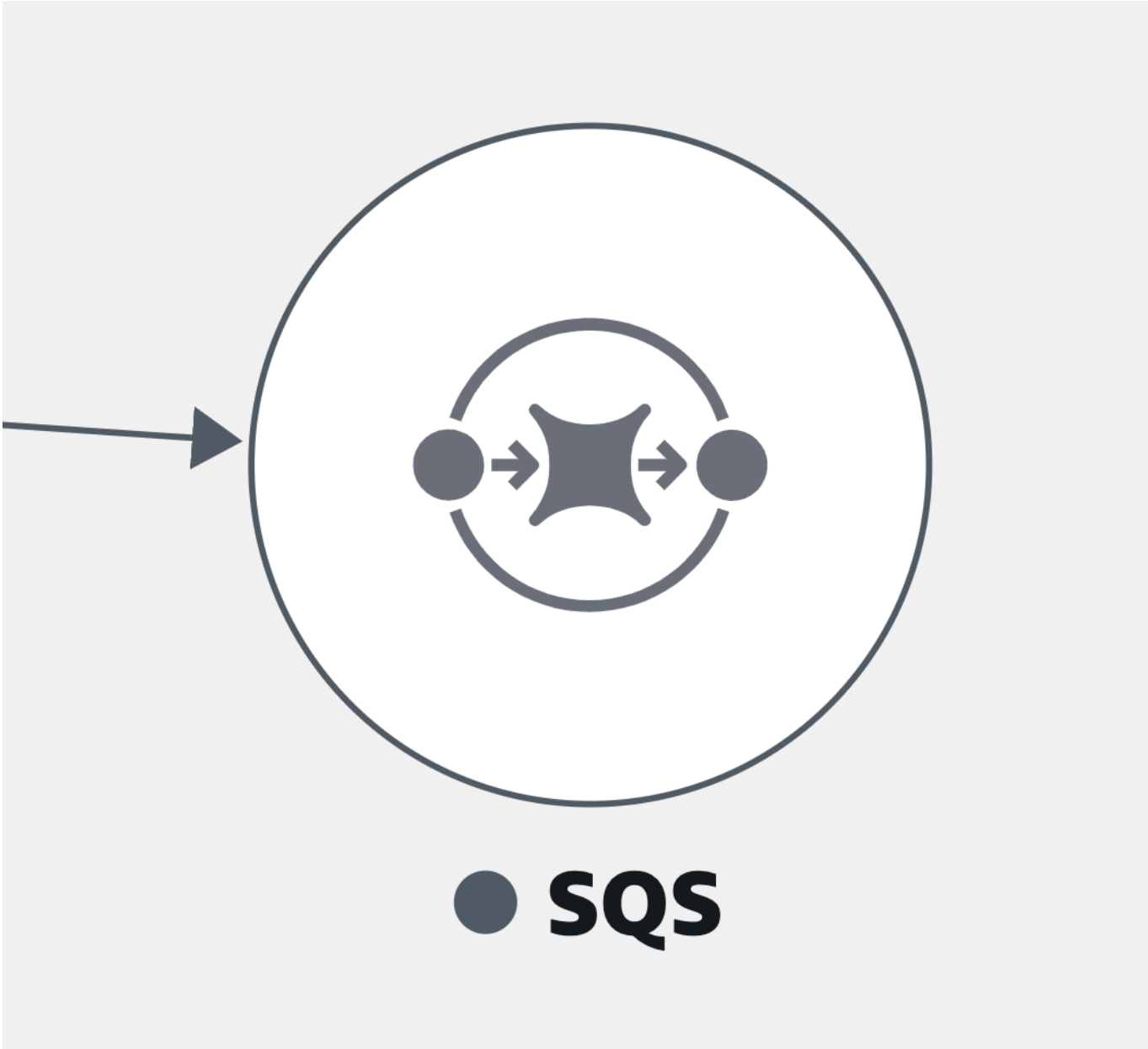
Le icone seguenti rappresentano esempi di nodi di dipendenza nella mappa dei servizi:

- Un bucket Amazon S3:

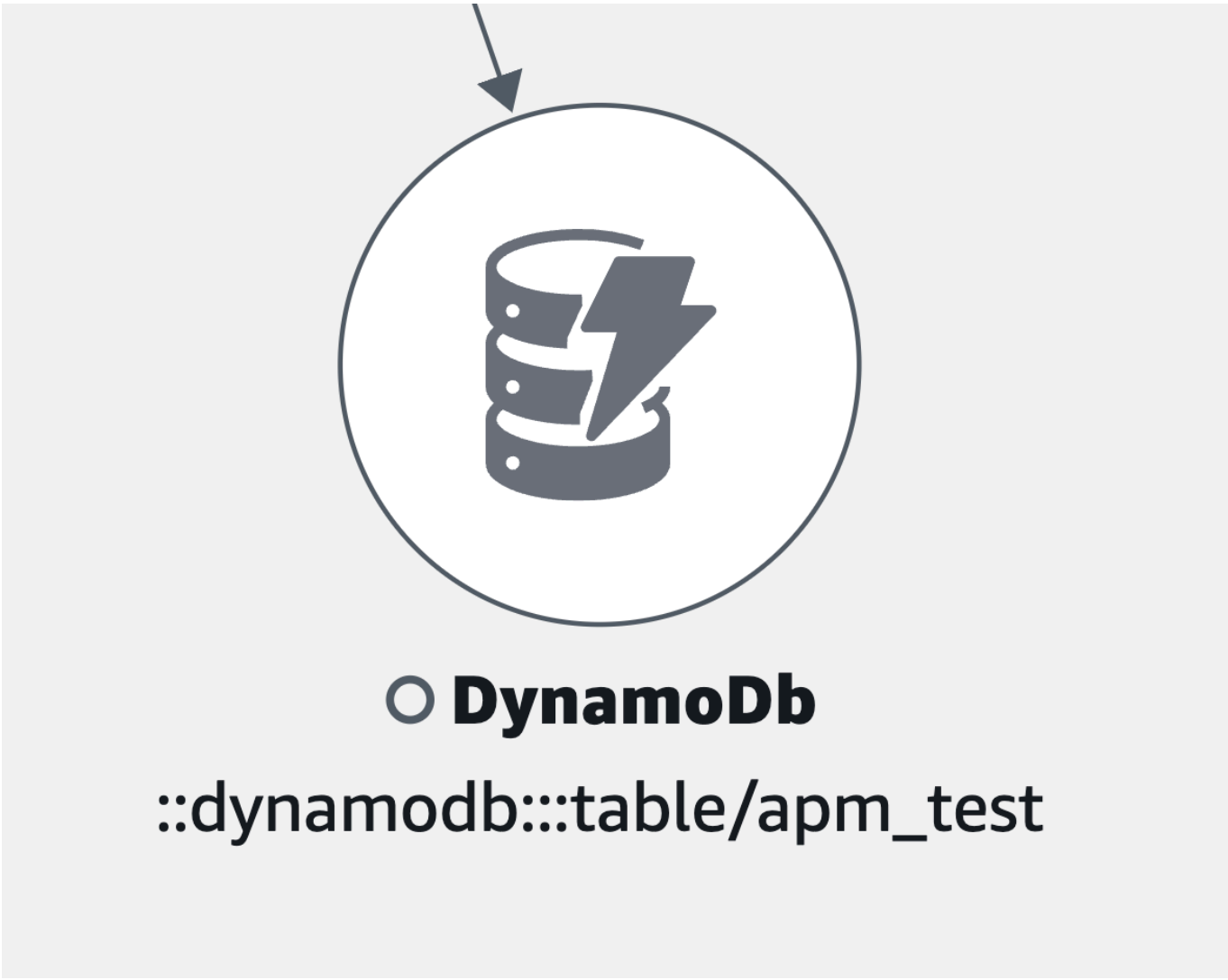
- Uno stream Amazon Kinesis:



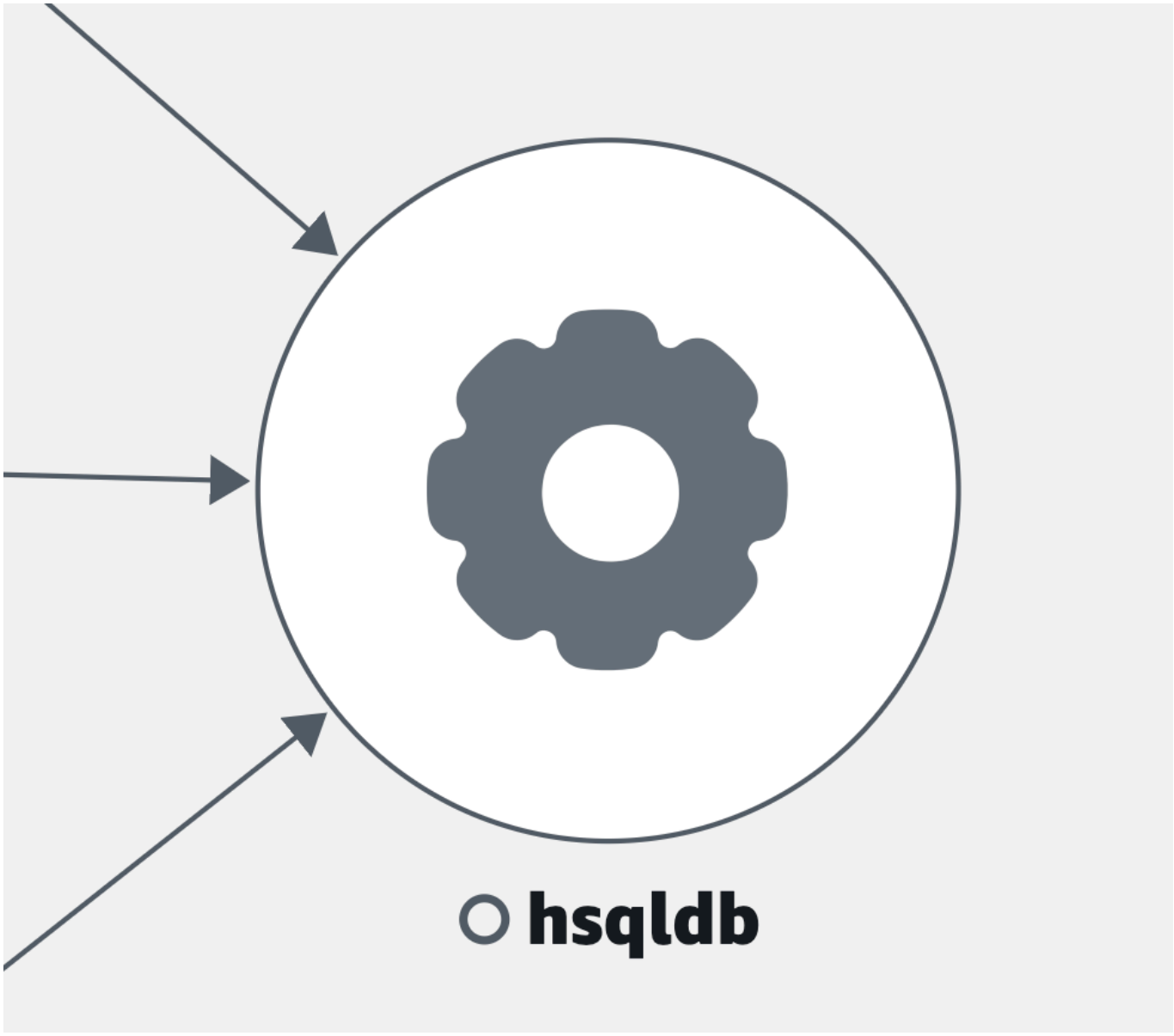
- Amazon Simple Queue Service (Amazon SQS):



- Una tabella Amazon DynamoDB:



- Altri tipi di dipendenza non elencati in precedenza:



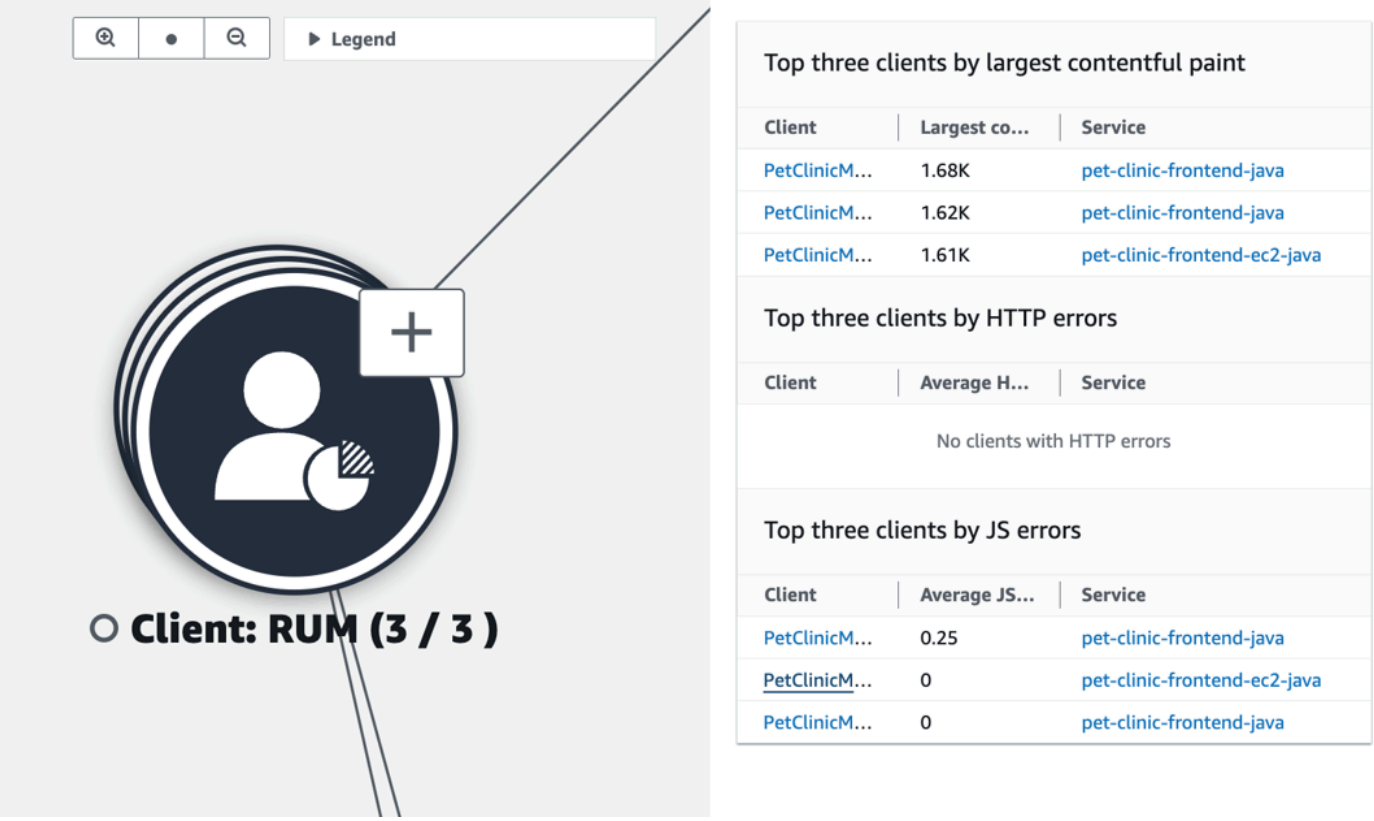
View clients

Dopo aver attivato il tracciamento X-Ray per i client Web CloudWatch RUM, questi vengono visualizzati sulla mappa dei servizi connessi ai servizi che chiamano.

Scegliete un nodo client per aprire un riquadro che mostra informazioni dettagliate sul client:

- Parametri relativi al caricamento delle pagine, al tempo medio di caricamento, agli errori e ai parametri vitali web medi.
- Un grafico che mostra una suddivisione degli errori.
- Un link per visualizzare i dettagli del client in CloudWatch RUM.

Per impostazione predefinita, i client RUM sono raggruppati in un'unica icona espandibile. Selezionate l'icona (+), come mostrato nell'immagine seguente, per espandere il gruppo e visualizzarne i singoli elementi.



L'icona seguente rappresenta un esempio di client RUM nella mappa dei servizi:

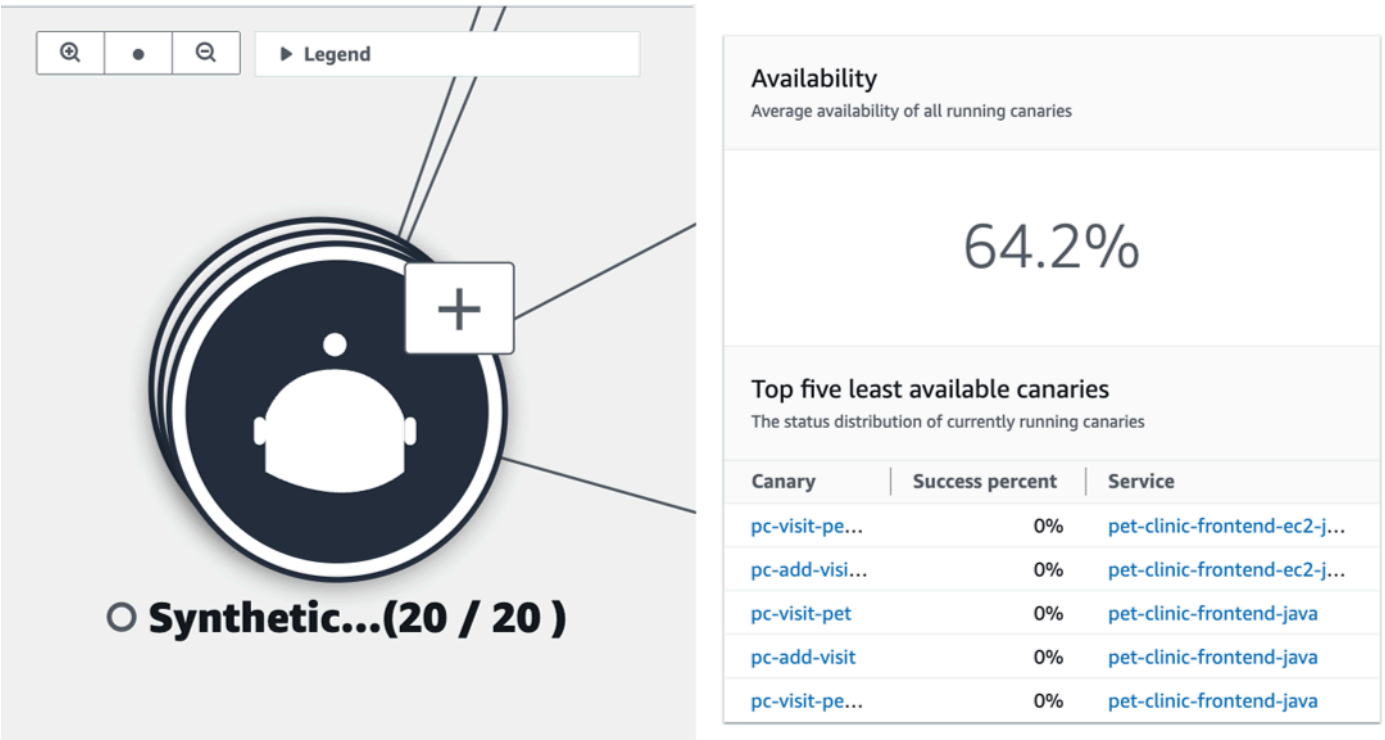
- Un client RUM:

> **Note**
>
> Per visualizzare gli errori AJAX nelle pagine dei client, utilizzate la versione 1.15 o successiva del client web CloudWatch RUM.

View synthetics canaries

Dopo aver attivato il AWS X-Ray tracciamento per i CloudWatch canarini Synthetics, questi vengono visualizzati sulla mappa dei servizi collegati ai servizi che chiamano, come mostrato nella seguente immagine di esempio:

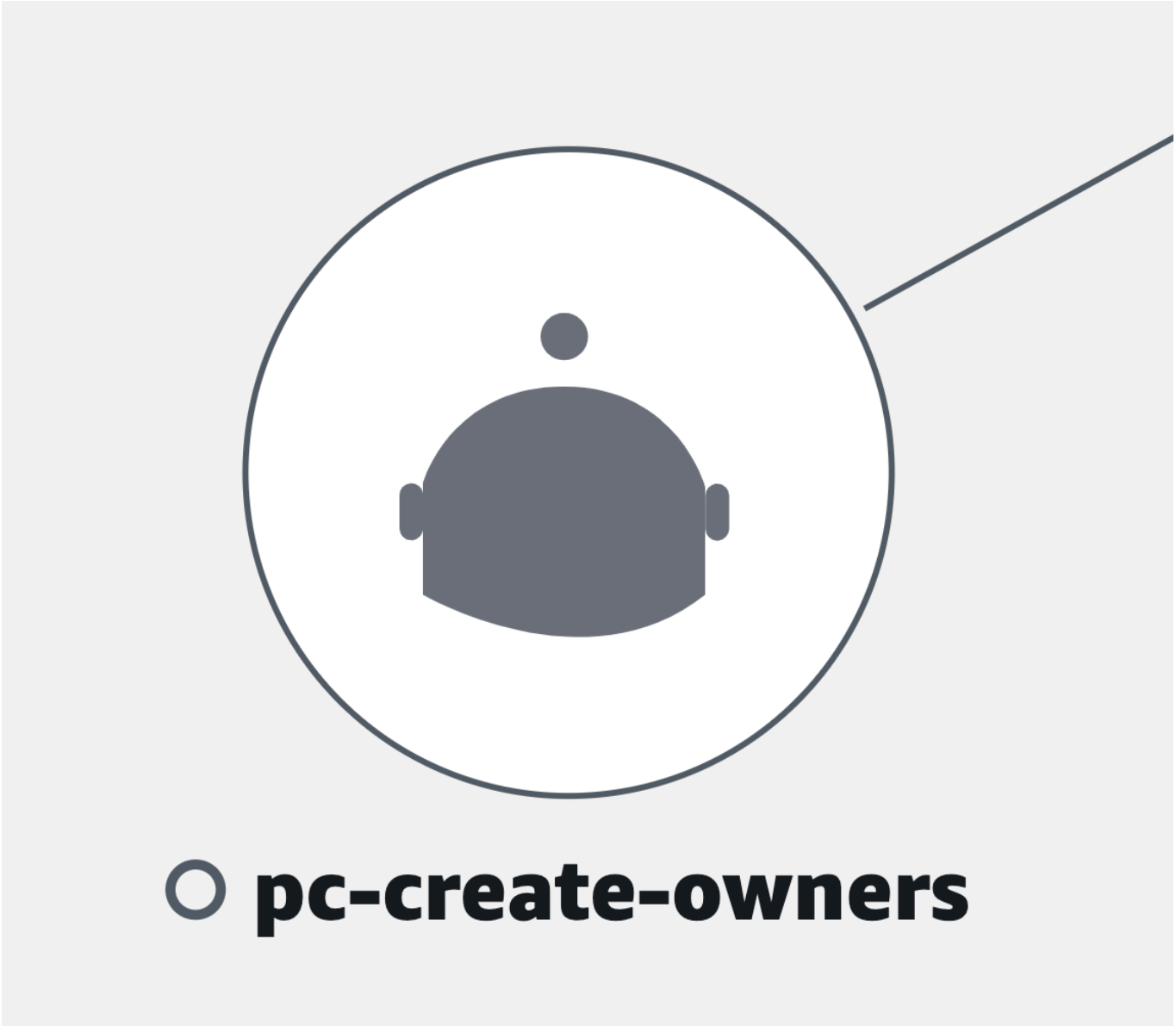
Scegliete un nodo canarino per aprire un riquadro che mostra informazioni dettagliate sui canarini, come mostrato nell'immagine seguente:

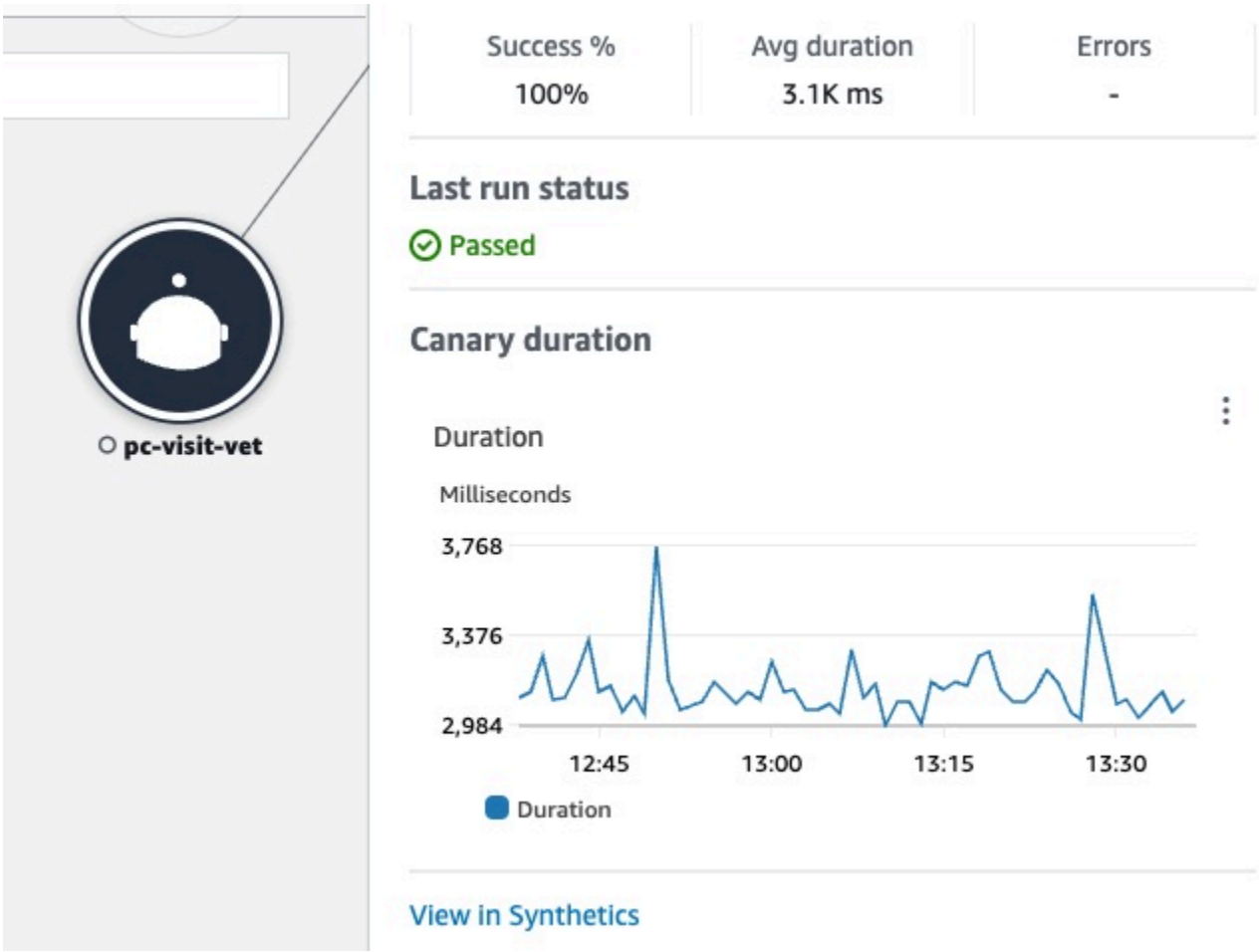


I canarini sono raggruppati per impostazione predefinita in un'unica icona espandibile. Seleziona l'icona (+), come mostrato nell'immagine precedente, per espandere il gruppo e visualizzarne i singoli elementi.

Le icone seguenti rappresentano esempi di client nella mappa dei servizi:

- Un canarino sintetico —


pc-create-owners



Nel riquadro relativo ai nodi canarini, puoi vedere quanto segue:

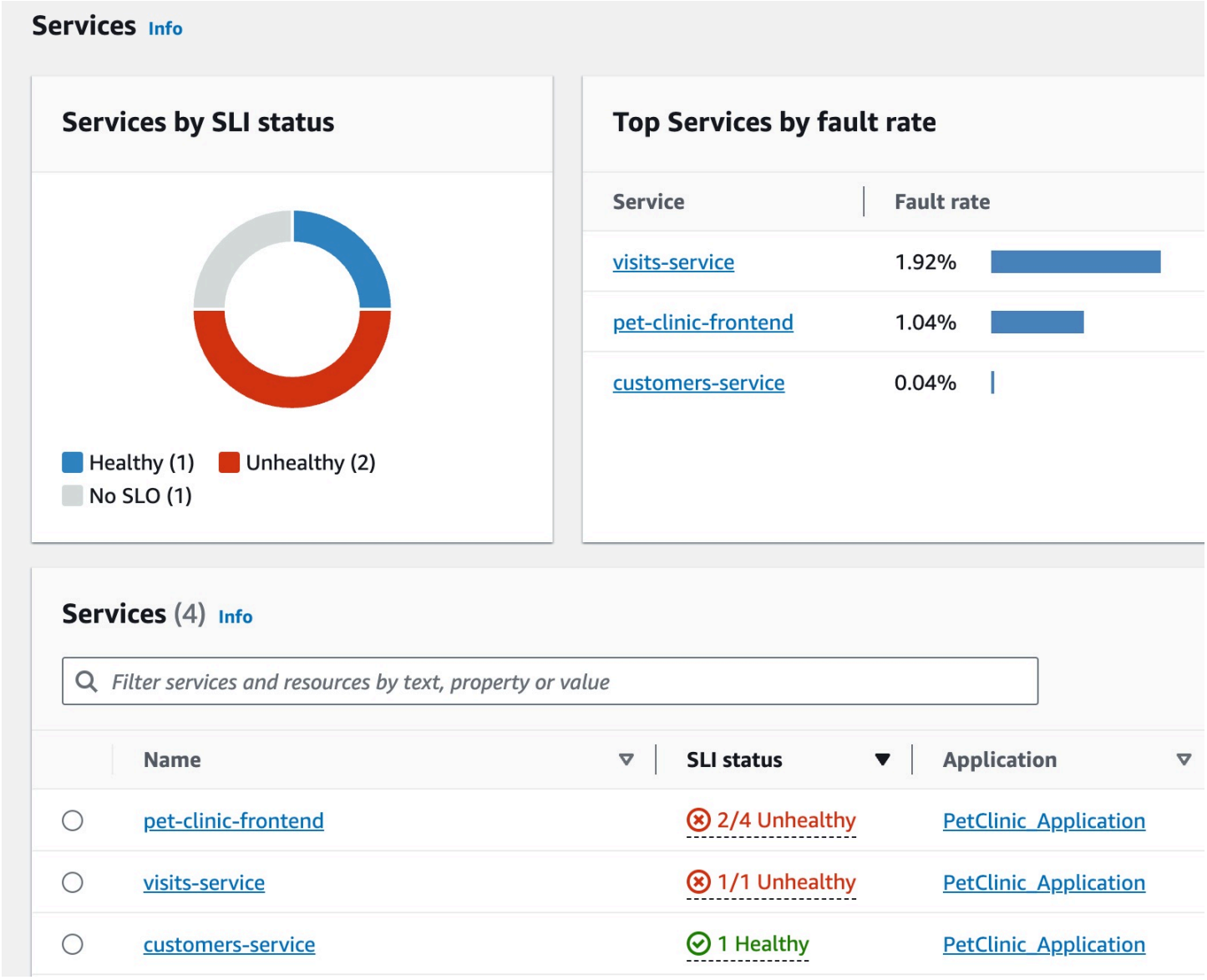
- Parametri relativi alla percentuale di successo, alla durata media e agli errori.
- Lo stato dell'ultima esecuzione canary.
- Un grafico che mostra la durata dell'esecuzione canary. Passa il mouse su una serie di grafici per visualizzare un pop-up contenente ulteriori informazioni.
- Un collegamento per visualizzare i dettagli dei canarini in CloudWatch Synthetics.

Esempio: utilizzo di Application Signals per risolvere un problema di integrità operativa

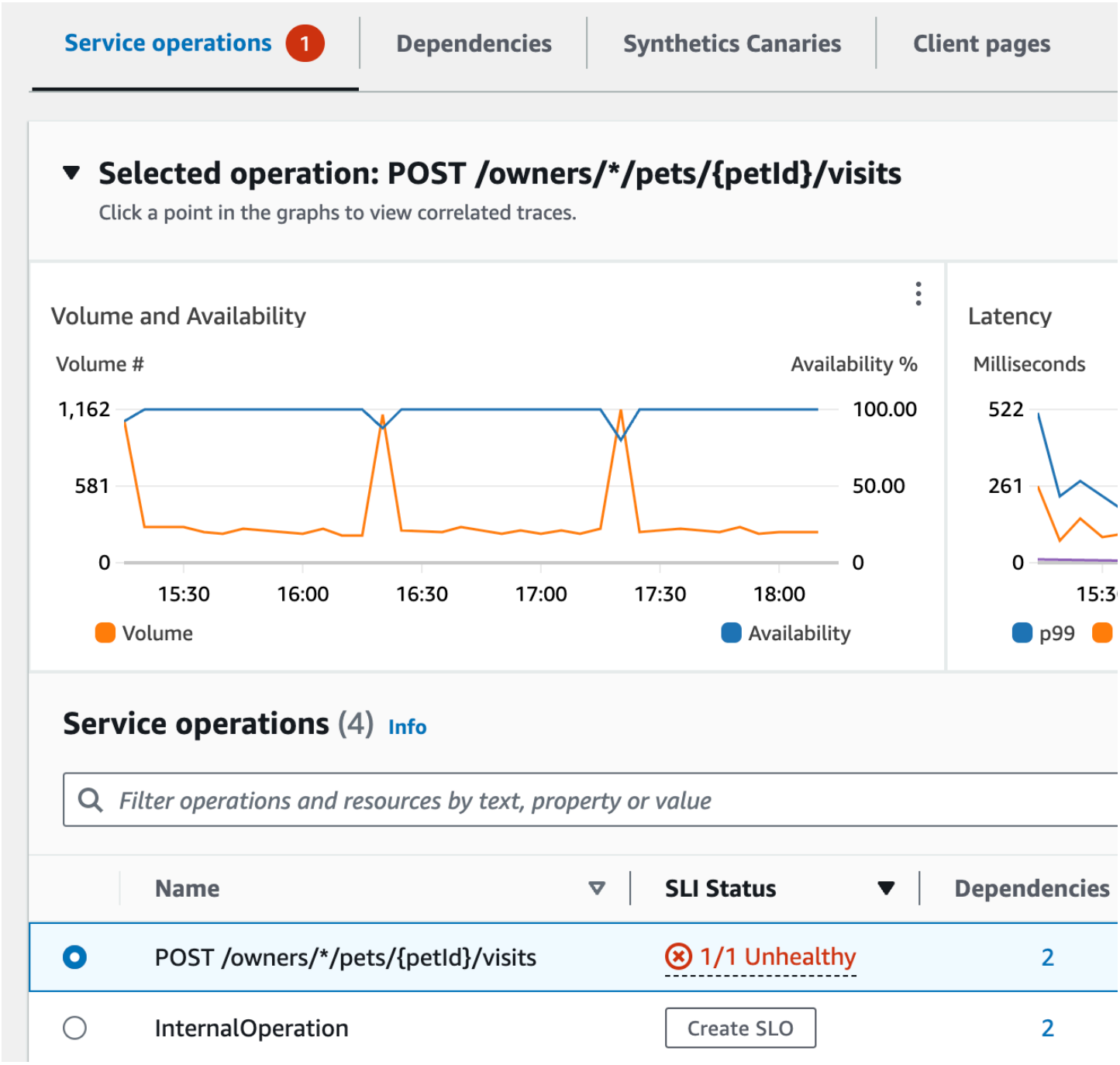
⚠ Application Signals è in versione di anteprima per Amazon CloudWatch ed è soggetta a modifiche.

Lo scenario seguente fornisce un esempio di come Application Signals può essere utilizzato per monitorare i servizi e identificare problemi di qualità. Approfondisci per identificare le cause principali potenziali e intervenire per risolvere il problema. Questo esempio è incentrato su un'applicazione per una clinica per animali composta da diversi microservizi che chiamano Servizi AWS DynamoDB.

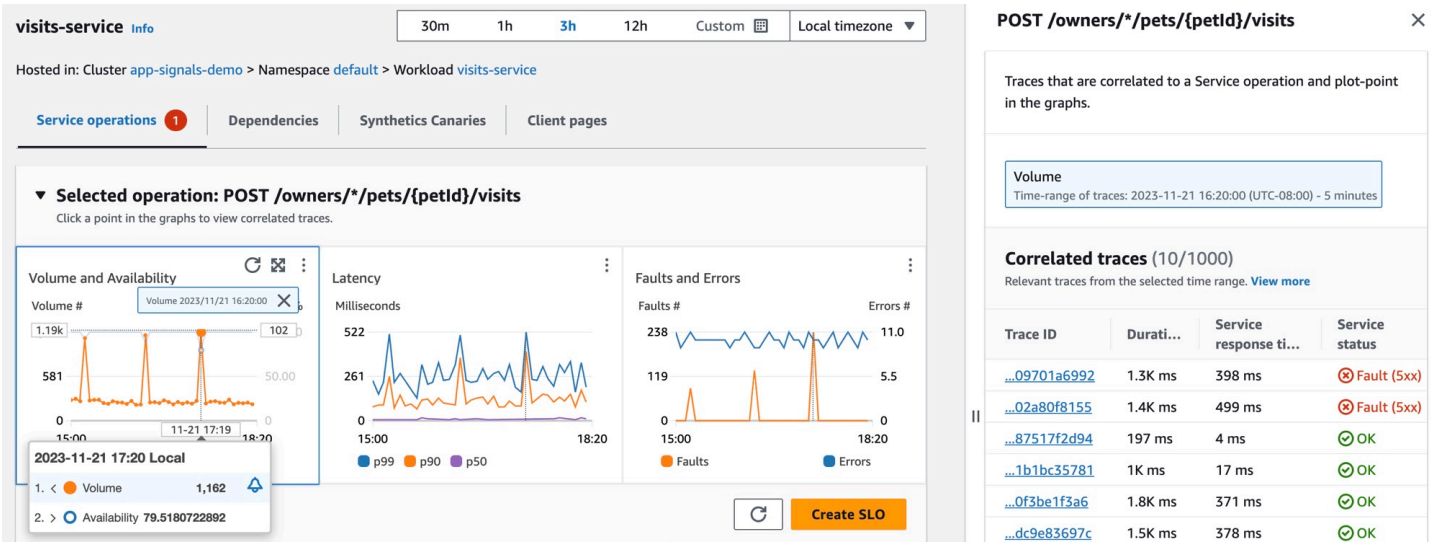
Jane fa parte di un DevOps team che supervisiona lo stato operativo di un'applicazione di clinica per animali domestici. Il team di Jane si impegna a garantire che l'applicazione sia altamente disponibile e reattiva. Utilizza gli obiettivi del livello di servizio (SLO) per misurare le prestazioni delle applicazioni rispetto a questi impegni aziendali. Riceve un avviso relativo a diversi indicatori del livello di servizio (SLI) non integri. Apre la CloudWatch console e accede alla pagina Servizi, dove vede diversi servizi in uno stato non integro.



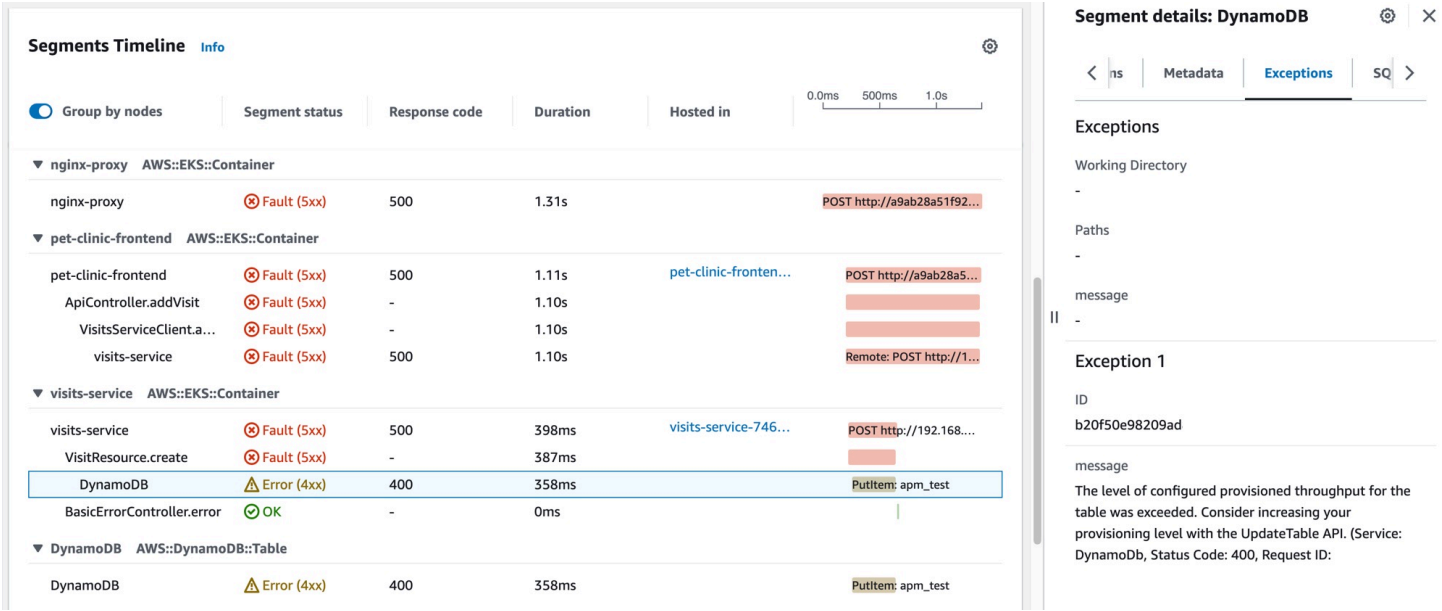
Nella parte superiore della pagina, Jane vede che `visits-service` è il servizio migliore per quanto riguarda la percentuale di guasti. Seleziona il collegamento nel grafico, che apre la pagina dei dettagli del servizio. Vede che c'è un'operazione non integra nella tabella delle operazioni del servizio. Seleziona questa operazione e vede nel grafico Volume e Disponibilità che ci sono picchi periodici di volume delle chiamate che sembrano essere correlati ai cali di disponibilità.



Per esaminare ulteriormente i cali di disponibilità del servizio, Jane seleziona uno dei punti dati della disponibilità nel grafico. Si apre un cassetto che mostra le tracce X-Ray correlate al punto dati selezionato. Vede che ci sono più tracce contenenti errori.



Jane seleziona una delle tracce correlate con uno stato di errore, aprendo la relativa pagina dei dettagli del tracciamento X-Ray. Jane scorre verso il basso fino alla sezione relativa alla sequenza temporale dei segmenti e segue il percorso della chiamata finché non vede che le chiamate a una tabella DynamoDB restituiscono errori. Seleziona il segmento DynamoDB e accede alla scheda Eccezioni del cassetto a destra.



Jane rileva che una risorsa DynamoDB non è configurata correttamente, con conseguenti errori durante i picchi di richieste dei clienti. Il livello di velocità di trasmissione effettiva assegnato dalla tabella DynamoDB viene periodicamente superato, con conseguenti problemi di disponibilità

del servizio e SLI non integri. Sulla base di queste informazioni, il suo team può configurare un livello più alto di velocità di trasmissione effettiva assegnata e garantire un'elevata disponibilità dell'applicazione.

Parametri dell'applicazione standard raccolti

⚠ Application Signals è in versione di anteprima. Se hai commenti su questa funzionalità, puoi contattarci all'indirizzo app-signals-feedback@amazon .com.

Application Signals raccoglie i parametri dell'applicazione standard dai servizi che rileva. Questi parametri si riferiscono agli aspetti più critici delle prestazioni di un servizio: latenza, guasti ed errori. Possono aiutarti a identificare i problemi, monitorare le tendenze delle prestazioni e ottimizzare le risorse per migliorare complessivamente l'esperienza utente.

La tabella seguente elenca i parametri raccolti da Application Signals. Queste metriche vengono inviate CloudWatch nel `AppSignals` namespace.

Parametro	Descrizione
`Latency`	Il ritardo prima dell'inizio del trasferimento dei dati dopo la richiesta. Unità: millisecondi
`Faults`	Un conteggio degli errori sul lato server HTTP 5XX e degli errori di stato span. OpenTelemetry Unità: nessuna
`Errors`	Un numero di errori HTTP 4XX lato client. Questi sono considerati errori di richiesta non causati da problemi del servizio. Pertanto, il parametro `Availability` visualizzato nei pannelli di controllo di Application Signals non considera questi errori guasti del servizio. Unità: nessuna

La **`Availability`** metrica visualizzata nei dashboard di Application Signals viene calcolata come (1 - /totale) *100. **`Faults`** Le risposte con esito positivo sono tutte risposte senza errori `5XX`. Le risposte `4XX` vengono considerate corrette durante il calcolo di `Availability` di Application Signals.

Dimensioni raccolte e combinazioni di dimensioni

Le seguenti dimensioni sono definite per ciascuno dei parametri standard dell'applicazione. Per ulteriori informazioni sulle dimensioni, consulta Dimensioni.

Vengono raccolte diverse dimensioni per i parametri del servizio e i parametri di dipendenza. All'interno dei servizi scoperti da Application Signals, quando il microservizio A chiama il microservizio B, il microservizio B elabora la richiesta. In questo caso, il microservizio A emette parametri di dipendenza e il microservizio B emette parametri del servizio. Quando un client chiama il microservizio A, il microservizio A elabora la richiesta ed emette i parametri del servizio.

Dimensioni per i parametri del servizio

Le seguenti dimensioni vengono utilizzate per i parametri del servizio.

Dimensione	Descrizione
`Service`	Nome del servizio.
`Operation`	Nome dell'operazione API o altra attività.
`HostedIn.EKS.Cluster`	Nome del cluster Amazon EKS su cui sono eseguiti i servizi. Questa dimensione viene raccolta solo se i servizi sono in esecuzione su Amazon EKS.
`HostedIn.K8s.Namespace`	Nome dello spazio dei nomi Kubernetes in cui sono eseguiti i servizi. Questa dimensione viene raccolta solo se i servizi sono in esecuzione su Amazon EKS.
`HostedIn.Environment`	Nome definito dall'utente dell'ambiente in cui i servizi sono in esecuzione. Questa dimensione viene raccolta solo se i servizi sono in esecuzione in un ambiente diverso da Amazon EKS.

Quando visualizzi queste metriche nella CloudWatch console, puoi scegliere di visualizzarle con le seguenti combinazioni di dimensioni.

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`

Per le piattaforme diverse da Amazon EKS, puoi anche visualizzare i parametri del servizio con le seguenti combinazioni di dimensioni.

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`

Dimensioni per i parametri di dipendenza

Le seguenti dimensioni vengono utilizzate per i parametri di dipendenza.

Dimensione	Descrizione
`Service`	Nome del servizio.
`Operation`	Nome dell'operazione API o altra attività.
`RemoteService`	Nome del servizio remoto richiamato.
`RemoteOperation`	Nome dell'operazione API richiamata.
`HostedIn.EKS.Cluster`	Nome del cluster Amazon EKS su cui sono eseguiti i servizi. Questa dimensione viene raccolta solo se i servizi sono in esecuzione su Amazon EKS.
`HostedIn.K8s.Namespace`	Nome dello spazio dei nomi Kubernetes in cui sono eseguiti i servizi. Questa dimensione viene raccolta solo se i servizi sono in esecuzione su Amazon EKS.
`K8s.RemoteNamespace`	Nome dello spazio dei nomi Kubernetes in cui sono eseguiti i servizi di dipendenza.

Dimensione	Descrizione
	Questa dimensione viene raccolta solo se i servizi sono in esecuzione su Amazon EKS.
`RemoteTarget`	Nome della risorsa richiamata dalle chiamate remote. Questa dimension e non ha un valore se le chiamate remote non sono dirette verso risorse specifiche. Questa dimensione viene raccolta solo se i servizi sono in esecuzione su Amazon EKS.
`HostedIn. Environment`	Nome definito dall'utente dell'ambiente in cui i servizi sono in esecuzione. Questa dimensione viene raccolta solo se i servizi sono in esecuzione in un ambiente diverso da Amazon EKS.

Quando visualizzi queste metriche nella CloudWatch console, puoi scegliere di visualizzarle con le seguenti combinazioni di dimensioni.

Esecuzione su qualsiasi piattaforma

- `RemoteService`

Esecuzione su cluster Amazon EKS

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `RemoteService`, `K8s.RemoteNamespace`

- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`

Esecuzione su piattaforme diverse dai cluster Amazon EKS

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.Environment`, `RemoteService`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,

Usa il monitoraggio sintetico

Puoi usare Amazon CloudWatch Synthetics per creare canaries, script configurabili eseguiti secondo una pianificazione, per monitorare endpoint e API. I canary seguono gli stessi percorsi ed eseguono le stesse azioni di un cliente, il che ti consente di verificare continuamente la tua esperienza cliente anche quando non hai alcun traffico clienti sulle tue applicazioni. Con i canary puoi scoprire i problemi prima che vengano rilevati dai clienti.

I canary sono script scritti in Node.js o Python. Creano funzioni Lambda nel tuo account che utilizzano Node.js o Python come framework. I canary funzionano su protocolli HTTP e HTTPS. Le Canarie utilizzano livelli Lambda che contengono la libreria CloudWatch Synthetics. La libreria contiene la versione NodeJS di CloudWatch Synthetics for NodeJS canaries e la versione Python di Synthetics for Python canaries. CloudWatch I livelli appartengono all'account del servizio CloudWatch Synthetics. Le biblioteche non trasmettono o archiviano mai le informazioni sui clienti. Tutti i dati dei clienti vengono memorizzati solo nell'account del cliente.

I canary offrono accesso programmatico a un browser Google Chrome headless tramite Puppeteer o Selenium Webdriver. Per ulteriori informazioni su Puppeteer, consulta Puppeteer. Per ulteriori informazioni su Selenium, consulta www.selenium.dev/.

I canary controllano la disponibilità e la latenza degli endpoint e possono archiviare i dati relativi al tempo di caricamento e le schermate dell'interfaccia utente. Monitorano le REST API, gli URL e il contenuto del sito Web e possono verificare le modifiche non autorizzate da phishing, code injection e cross-site scripting.

CloudWatch Synthetics è integrato con Application Signals, che può scoprire e monitorare i servizi applicativi, i client, i canari Synthetics e le dipendenze dei servizi. Utilizza Application Signals per visualizzare un elenco o una mappa visiva dei tuoi servizi, visualizzare i parametri di integrità in base agli obiettivi del livello di servizio (SLO) e approfondire le tracce X-Ray correlate per una risoluzione dei problemi più dettagliata. Per vedere i canary in Application Signals, attiva il tracciamento attivo X-Ray. I tuoi canary vengono visualizzati nella mappa del servizio connessa ai tuoi servizi e nella pagina dei dettagli dei servizi che chiamano.

Per una dimostrazione video dei canary, consulta quanto segue:

- Introduzione ad Amazon CloudWatch Synthetics
- Dimostrazione di Amazon CloudWatch Synthetics
- Crea canarie usando Amazon CloudWatch Synthetics
- Monitoraggio visivo con Amazon CloudWatch Synthetics

Puoi eseguire un canary una volta o a intervalli regolari. I canary possono funzionare con una frequenza di una volta al minuto. Puoi utilizzare espressioni Cron e Rate per programmare i canary.

Per informazioni sui problemi di sicurezza da considerare prima di creare ed eseguire i canary, consulta Considerazioni sulla sicurezza per Canary Synthetics.

Per impostazione predefinita, i canari creano diverse CloudWatch metriche nel namespace. `CloudWatchSynthetics` Questi parametri hanno `CanaryName` come dimensione. Anche i canary che utilizzano la funzione `executeStep()` o `executeHttpStep()` dalla libreria funzioni hanno `StepName` come dimensione. Per ulteriori informazioni sulla libreria di funzioni canary, consulta Funzioni di libreria disponibili per gli script canary.

CloudWatch Synthetics si integra bene con la X-Ray Trace Map, che CloudWatch utilizza AWS X-Ray with per fornire end-to-end una panoramica dei vostri servizi per aiutarvi a individuare in modo

più efficiente i punti deboli delle prestazioni e identificare gli utenti interessati. I canarini creati con CloudWatch Synthetics vengono visualizzati sulla mappa di tracciamento. Per ulteriori informazioni sul tracciamento con X-Ray, consulta Mappa di tracciamento X-Ray.

CloudWatch Synthetics è attualmente disponibile in tutte le regioni AWS commerciali e le regioni. GovCloud

> **Note**
>
> In Asia Pacifico (Osaka), non AWS PrivateLink è supportato. In Asia Pacifico (Jakarta), AWS PrivateLink e X-Ray non sono supportati.

Argomenti

Ruoli e autorizzazioni richiesti per i canarini CloudWatch

Sia gli utenti che creano e gestiscono i canary che i canary stessi devono disporre di determinate autorizzazioni.

Ruoli e autorizzazioni richiesti per gli utenti che gestiscono i canarini CloudWatch

Per visualizzare i dettagli dei canary e i risultati delle esecuzioni dei canary, è necessario accedere come utente con le policy `CloudWatchSyntheticsFullAccess` o `CloudWatchSyntheticsReadOnlyAccess` collegate. Per leggere tutti i dati Synthetics nella console, sono necessarie anche le policy IAM `AmazonS3ReadOnlyAccesse` `CloudWatchReadOnlyAccess`. Per visualizzare il codice sorgente utilizzato dai canary, è necessaria anche la policy `AWSLambda_ReadOnlyAccess`.

Per creare i canary, è necessario eseguire l'accesso come utente che dispone della policy `CloudWatchSyntheticsFullAccess` o di un insieme di autorizzazioni simile. Per creare ruoli IAM per i canary, è necessaria anche la seguente istruzione sulle policy inline:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

> **Important**
>
> La concessione a un utente delle `iam:CreateRole` `iam:AttachRolePolicy` autorizzazioni e e conferisce a tale utente l'accesso amministrativo completo al tuo account. `iam:CreatePolicy` AWS Ad esempio, un utente con queste autorizzazioni può creare una policy che dispone di autorizzazioni complete per tutte le risorse e collegare tale policy a qualsiasi ruolo. Presta molta attenzione a chi concedi queste autorizzazioni.

Per informazioni su come collegare policy e concedere autorizzazioni a utenti, consulta Modifica delle autorizzazioni per un utente IAM e Per incorporare una policy inline per un utente o un ruolo.

Ruoli e autorizzazioni richiesti per i canary

Ogni canary deve essere associato a un ruolo IAM con determinate autorizzazioni collegate. Quando crei un canarino utilizzando la CloudWatch console, puoi scegliere che CloudWatch Synthetics crei un ruolo IAM per il canarino. In questo modo, il ruolo avrà le autorizzazioni necessarie.

Se si desidera creare personalmente il ruolo IAM o creare un ruolo IAM da utilizzare quando si usa la AWS CLI o le API per creare un canary, il ruolo deve contenere le autorizzazioni elencate in questa sezione.

Tutti i ruoli IAM per i canary devono includere la seguente istruzione sulle policy di attendibilità.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Inoltre, per il ruolo IAM del canary è necessaria una delle seguenti istruzioni.

Canary di base che non utilizza AWS KMS o non richiede l'accesso ad Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/s3/bucket/canary/results/folder"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/s3/bucket/that/contains/canary/results"

            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
               "StringEquals": {
                "cloudwatch:namespace": "CloudWatchSynthetics"
               }
            }
        }
```

```
    ]
}
```

Canary che utilizza AWS KMS per crittografare gli artefatti dei canarini ma non necessita dell'accesso ad Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
```

```
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Canary che non utilizza AWS KMS ma necessita dell'accesso ad Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
```

```
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
```

```
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       }
    ]
}
```

Canary che utilizza AWS KMS per crittografare gli artefatti dei canarini e necessita anche dell'accesso ad Amazon VPC

Se aggiorni un canary non VPC per iniziare a utilizzare un VPC, devi aggiornare il ruolo del canary per includere le autorizzazioni dell'interfaccia di rete elencate nella seguente policy.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       },
```

```
        {
             "Effect": "Allow",
             "Action": [
                 "kms:Decrypt",
                 "kms:GenerateDataKey"
             ],
             "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
             "Condition": {
                 "StringEquals": {
                     "kms:ViaService": [
                         "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                     ]
                 }
             }
         }
    ]
}
```

AWS politiche gestite per CloudWatch Synthetics

Per aggiungere le autorizzazioni a utenti, gruppi e ruoli, è più semplice utilizzare policy gestite da AWS piuttosto che scrivere autonomamente le policy. La creazione di policy gestite dai clienti IAM che forniscono al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre policy AWS gestite. Queste politiche coprono casi d'uso comuni e sono disponibili nel tuo AWS account. Per ulteriori informazioni sulle policy gestite da AWS , consulta Policy gestite da AWS policy gestite da AWS nella Guida per l'utente di IAM.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi modificano occasionalmente le autorizzazioni in una policy gestita da AWS . Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy.

CloudWatch Synthetics: aggiornamenti alle politiche gestite AWS

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per CloudWatch Synthetics da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei CloudWatch documenti.

Modifica	Descrizione	Data
Azioni ridondanti rimosse da CloudWatchSyntheticsFullAccess	CloudWatch Synthetics ha rimosso le azioni `lambda:GetLayerVersionByArn` and `s3:PutBucketEncryption` CloudWatchSyntheticsFullAccessdalla policy perché tali azioni erano ridondanti rispetto ad altre autorizzazioni della policy. Le operazioni rimosse non hanno fornito alcuna autorizzazione e non vi è alcuna modifica netta alle autorizzazioni concesse dalla policy.	12 marzo 2021
CloudWatch Synthetics ha iniziato a tracciare le modifiche	CloudWatch Synthetics ha iniziato a tenere traccia delle modifiche per AWS le sue politiche gestite.	10 marzo 2021

CloudWatchSyntheticsFullAccess

Ecco i contenuti della policy `CloudWatchSyntheticsFullAccess`:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "synthetics:*"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
```

```
         "Action":[
            "s3:CreateBucket",
            "s3:PutEncryptionConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::cw-syn-results-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles",
            "s3:ListAllMyBuckets",
            "s3:GetBucketLocation",
            "xray:GetTraceSummaries",
            "xray:BatchGetTraces",
            "apigateway:GET"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource":"arn:aws:s3:::cw-syn-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::aws-synthetics-library-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ],
         "Resource":[
            "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
         ],
         "Condition":{
```

```
            "StringEquals":{
               "iam:PassedToService":[
                  "lambda.amazonaws.com",
                  "synthetics.amazonaws.com"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:GetRole"
         ],
         "Resource":[
            "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "cloudwatch:GetMetricData",
            "cloudwatch:GetMetricStatistics"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "cloudwatch:PutMetricAlarm",
            "cloudwatch:DeleteAlarms"
         ],
         "Resource":[
            "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "cloudwatch:DescribeAlarms"
         ],
         "Resource":[
            "arn:aws:cloudwatch:*:*:alarm:*"
         ]
      },
```

```
      {
         "Effect":"Allow",
         "Action":[
            "lambda:CreateFunction",
            "lambda:AddPermission",
            "lambda:PublishVersion",
            "lambda:UpdateFunctionConfiguration",
            "lambda:GetFunctionConfiguration"
         ],
         "Resource":[
            "arn:aws:lambda:*:*:function:cwsyn-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "lambda:GetLayerVersion",
            "lambda:PublishLayerVersion"
         ],
         "Resource":[
            "arn:aws:lambda:*:*:layer:cwsyn-*",
            "arn:aws:lambda:*:*:layer:Synthetics:*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:ListTopics"
         ],
         "Resource":[
            "*"
         ]
      },
```

```
      {
         "Effect":"Allow",
         "Action":[
            "sns:CreateTopic",
            "sns:Subscribe",
            "sns:ListSubscriptionsByTopic"
         ],
         "Resource":[
            "arn:*:sns:*:*:Synthetics-*"
         ]
      }
   ]
}
```

CloudWatchSyntheticsReadOnlyAccess

Ecco i contenuti della policy `CloudWatchSyntheticsReadOnlyAccess`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Limitazione di un utente alla visualizzazione dei canary specifici

È possibile limitare la capacità di un utente di visualizzare le informazioni sui canary, in modo che possano visualizzare solo le informazioni sui canary specificati. Per fare ciò, utilizza una policy IAM con un istruzione `Condition` simile alla seguente e collega questa policy a un utente IAM o ruolo IAM.

L'esempio seguente limita l'utente a visualizzare solo le informazioni su `name-of-allowed-canary-1` e `name-of-allowed-canary-2`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "synthetics:Names": [
                        "name-of-allowed-canary-1",
                        "name-of-allowed-canary-2"
                    ]
                }
            }
        }
    ]
}
```

CloudWatch Synthetics supporta l'elenco di fino a cinque elementi nell'array. `synthetics:Names`

Puoi anche creare una policy che utilizza un * come carattere jolly nei nomi canary che è necessario consentire, come nell'esempio seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "synthetics:Names": [
                        "my-team-canary-*"
                    ]
                }
            }
        }
    ]
```

```
}
```

Qualsiasi utente che ha effettuato l'accesso con una di queste politiche allegate non può utilizzare la CloudWatch console per visualizzare le informazioni sulle isole Canarie. Possono visualizzare le informazioni sui canari solo per i canarini autorizzati dalla politica e solo utilizzando l'DescribeCanariesAPI o il comando describe-canaries. AWS CLI

Creazione di un Canary

> **Important**
>
> Assicurati di utilizzare Canary Synthetics per monitorare solo gli endpoint e le API in cui disponi di autorizzazioni o di cui sei proprietario. In base alle impostazioni di frequenza del canary, per questi endpoint può verificarsi un aumento del traffico.

Quando usi la CloudWatch console per creare un canarino, puoi usare un progetto fornito da CloudWatch per creare il tuo canarino oppure puoi scrivere il tuo script. Per ulteriori informazioni, consulta Utilizzo dei blueprint dei canary.

Puoi anche creare un canarino usando AWS CloudFormation uno script personalizzato per il canarino. Per ulteriori informazioni, consulta AWS::Synthetics::Canaryla Guida per l'AWS CloudFormation utente.

Se stai scrivendo il tuo script, puoi usare diverse funzioni che CloudWatch Synthetics ha integrato in una libreria. Per ulteriori informazioni, consulta Versioni di runtime Synthetics.

> **Note**
>
> Quando create un canarino, uno dei livelli creati è un livello Synthetics a cui fa preposto. `Synthetics` Questo livello è di proprietà dell'account del servizio Synthetics e contiene il codice di runtime.

Per creare un Canary

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Synthetics Canaries.

3. Scegli Create Canary (Crea Canary).
4. Scegli una delle seguenti opzioni:

   - Per basare il canary su uno script di blueprint, scegli Use a blueprint (Usa un blueprint), quindi scegli il tipo di canary che desideri creare. Per ulteriori informazioni sulle attività di ciascun tipo di blueprint, consulta Utilizzo dei blueprint dei canary.
   - Per caricare il tuo script Node.js per creare un Canary personalizzato, scegli Upload a script (Carica uno script).

     Puoi quindi trascinare lo script in Script o scegliere Browse files (Sfoglia file) per passare allo script nel file system.
   - Per importare lo script da un bucket S3, scegli Import from S3 (Importa da S3). Quindi, in Source location (Posizione origine), digita il percorso completo al canary o scegli Browse S3 (Sfoglia S3).

     È necessario disporre delle autorizzazioni `s3:GetObject` e `s3:GetObjectVersion` per il bucket S3 in uso. Il bucket deve trovarsi nella stessa AWS regione in cui state creando il canarino.
5. In Name (Nome), immetti un nome per il Canary. Il nome viene utilizzato in molte pagine, pertanto è consigliabile assegnare un nome descrittivo che lo distingua da altri canary creati.
6. In Application or endpoint URL (URL applicazione o endpoint), immetti l'URL che il Canary deve testare. Questo URL deve includere il protocollo (ad esempio https://).

   Se si desidera che il canary esegua il test di un endpoint su un VPC, è inoltre necessario immettere le informazioni sul VPC più avanti in questa procedura.
7. Se utilizzi il tuo script per il canary, in Lambda handler (Gestore Lambda), immetti il punto d'ingresso da cui avviare il canary. Se utilizzi un runtime precedente a `syn-nodejs-puppeteer-3.4` o `syn-python-selenium-1.1`, la stringa immessa deve terminare con `.handler`. Se utilizzi `syn-nodejs-puppeteer-3.4` o `syn-python-selenium-1.1` oppure un runtime successivo, questa restrizione non viene applicata.
8. Se utilizzi variabili di ambiente nello script, scegli Environment variables (Variabili di ambiente) e specifica un valore per ciascuna variabile di ambiente definita nello script. Per ulteriori informazioni, consulta Variabili di ambiente.
9. In Schedule (Pianifica), scegli se eseguire questo canary una volta, continuamente utilizzando un'espressione Rate o utilizzando un'espressione Cron.

- Quando usi la CloudWatch console per creare un canarino che funziona continuamente, puoi scegliere una frequenza compresa tra una volta al minuto e una volta all'ora.

- Per ulteriori informazioni sulla scrittura di un'espressione Cron per la pianificazione del canary, consulta Pianificazione delle esecuzioni di canary usando cron.

10. (Facoltativo) Per impostare un valore di timeout per il canary, scegli Additional configuration (Configurazione aggiuntiva), quindi specificare il valore di timeout. Non più breve di 15 secondi per consentire l'avvio a freddo Lambda e il tempo necessario per avviare la strumentazione canary.
11. In Data retention (Conservazione dei dati), specifica per quanto tempo conservare le informazioni relative a esecuzioni Canary non riuscite e riuscite. L'intervallo è 1-455 giorni.

    Questa impostazione influisce solo sui dati che CloudWatch Synthetics archivia e visualizza nella console. Non influisce sui dati memorizzati nei bucket Amazon S3 o su registri o parametri pubblicati dal Canary.
12. In Data Storage (Storage dati), seleziona il bucket S3 da utilizzare per memorizzare i dati delle esecuzioni del canary. Il nome del bucket non può contenere un punto (.). Se lo lasci vuoto, verrà utilizzato o creato un bucket S3 predefinito.

    Se utilizzi `syn-nodejs-puppeteer-3.0` o runtime successivo, quando immetti l'URL per il bucket nella casella di testo, puoi specificare un bucket nell'area corrente o in un'area diversa. Se utilizzi una versione runtime precedente, il bucket deve trovarsi nell'area corrente.
13. (Facoltativo) Per impostazione predefinita, i canarini archiviano i propri artefatti su Amazon S3 e gli artefatti vengono crittografati quando sono inattivi utilizzando una chiave gestita. AWS AWS KMS È possibile utilizzare un'opzione di crittografia diversa scegliendo Configurazione aggiuntiva nella sezione Archiviazione dati. È quindi possibile scegliere il tipo di chiave da utilizzare per la crittografia. Per ulteriori informazioni, consulta Crittografia di artefatti canary.
14. In Access permissions (Autorizzazioni di accesso), scegli se creare un nuovo ruolo IAM per eseguire il canary o usarne uno esistente.

    Se il ruolo è stato CloudWatch creato da Synthetics, include automaticamente tutte le autorizzazioni necessarie. Se si desidera creare personalmente il ruolo, consulta Ruoli e autorizzazioni richiesti per i canary per ottenere informazioni sulle autorizzazioni necessarie.

    Se usi la CloudWatch console per creare un ruolo per un canarino quando crei il canarino, non puoi riutilizzare il ruolo per altri canarini, perché questi ruoli sono specifici per un solo canarino. È

possibile utilizzare un ruolo esistente solo se è stato creato manualmente un ruolo che funziona per più canary.

Per utilizzare un ruolo esistente, è necessario disporre dell' autorizzazione `iam:PassRole` per passare tale ruolo a Synthetics e Lambda. Devi anche avere l'autorizzazione `iam:GetRole`.

15. (Facoltativo) In Allarmi, scegli se desideri creare CloudWatch allarmi predefiniti per questo canarino. Se scegli di creare allarmi, questi vengono creati con la seguente convenzione di denominazione: `Synthetics-Alarm-`*`canaryName`*`-`*`index`*

    `index` è un numero che rappresenta ogni allarme creato per questo canary. Il primo allarme ha un indice di 1, il secondo allarme ha un indice di 2 e così via.

16. Per fare in modo che questo canary verifichi un endpoint che si trova su un VPC, scegli VPC settings (Impostazioni VPC) e procedi come segue:

    a. Seleziona il VPC che ospita l'endpoint.

    b. Seleziona una o più sottoreti nel VPC. È necessario selezionare una sottorete privata perché l'istanza Lambda non può essere configurata per l'esecuzione in una sottorete pubblica quando non è possibile assegnare un indirizzo IP all'istanza Lambda durante l'esecuzione. Per ulteriori informazioni, consulta Configurazione di una funzione Lambda per accedere alle risorse in un VPC.

    c. Seleziona uno o più gruppi di sicurezza nel VPC.

    Se l'endpoint si trova su un VPC, devi abilitare il tuo canary a inviare informazioni ad CloudWatch Amazon S3. Per ulteriori informazioni, consulta Esecuzione di un Canary su un VPC.

17. (Facoltativo) In Tags (Tag), aggiungere facoltativamente una o più coppie chiave/valore come tag per questo canary. I tag possono aiutarti a identificare e organizzare le tue AWS risorse e tenere traccia dei costi. AWS Per ulteriori informazioni, consulta Taggare le tue risorse Amazon CloudWatch .

18. (Facoltativo) In Active tracing (Tracciamento attivo) scegli se abilitare il tracciamento attivo con X-Ray per questo canary. Questa opzione è disponibile solo se il canary utilizza la versione runtime syn-nodejs-2.0 o versione successiva. Per ulteriori informazioni, consulta Canary e tracciamento X-Ray.

Risorse create per Canary

Quando crei un canary, vengono create le seguenti risorse:

- Un ruolo IAM con il nome `CloudWatchSyntheticsRole-`*`canary-name`*`-`*`uuid`* (se usi la CloudWatch console per creare il canarino e specifichi che deve essere creato un nuovo ruolo per il canarino)
- Una policy IAM con il nome `CloudWatchSyntheticsPolicy-`*`canary-name`*`-`*`uuid`*.
- Un bucket S3 con il nome `cw-syn-results-`*`accountID`*`-`*`region`*.
- Allarmi con il nome `Synthetics-Alarm-`*`MyCanaryName`* (se specifichi gli allarmi da creare per il canary)
- Funzioni Lambda e livelli, se utilizzi un piano per creare il Canary. Queste risorse hanno il prefisso `cwsyn-`*`MyCanaryName`*.
- CloudWatch Registra i gruppi di log con il nome. `/aws/lambda/cwsyn-`*`MyCanaryName`*`-`*`randomId`*

Utilizzo dei blueprint dei canary

Questa sezione fornisce dettagli su ciascuno dei blueprint dei canary e sulle attività per cui il blueprint è più adatto. I piani sono forniti per i seguenti tipi di Canary:

- Monitoraggio dell'heartbeat
- Canary API
- Controllo collegamento interrotto
- Monitoraggio visivo
- Registratore di Canary
- Flusso di lavoro dell'interfaccia utente grafica

Quando usi un blueprint per creare un canarino, mentre compili i campi nella CloudWatch console, l'area Script editor della pagina mostra il canarino che stai creando come script Node.js. Puoi anche modificare il canary in quest'area per personalizzarlo ulteriormente.

Monitoraggio dell'heartbeat

Gli script Hearbeat caricano l'URL specificato e memorizzano uno screenshot della pagina e un file di archivio HTTP (file HAR). Memorizzano anche i registri degli URL a cui si accede.

È possibile utilizzare i file HAR per visualizzare dati dettagliati sulle prestazioni delle pagine Web. Puoi analizzare l'elenco delle richieste Web e rilevare problemi di prestazioni, ad esempio il tempo di caricamento di un elemento.

Se il tuo canary usa la versione runtime `syn-nodejs-puppeteer-3.1` o versioni successive, puoi utilizzare il blueprint di monitoraggio dell'heartbeat per monitorare più URL e visualizzare lo stato, la durata, le schermate associate e il motivo dell'errore per ogni URL nel riepilogo dei passaggi del report di esecuzione del canary.

Canary API

I canary API possono testare le funzioni di lettura e scrittura di base di una REST API. REST sta per trasferimento di stato rappresentativo ed è un insieme di regole che gli sviluppatori seguono durante la creazione di un'API. Una di queste regole afferma che un collegamento a un URL specifico deve restituire una parte di dati.

I canary possono funzionare con qualsiasi API e testare tutti i tipi di funzionalità. Ogni canary può effettuare più chiamate API.

Nei canary che utilizzano la versione runtime `syn-nodejs-2.2` o versioni successive, il blueprint del canary API supporta i canary in più passaggi che monitorano le API come passaggi HTTP. È possibile testare più API in un singolo canary. Ogni passaggio è una richiesta separata che può accedere a un URL diverso, utilizzare intestazioni diverse e utilizzare regole diverse per l'acquisizione di intestazioni e corpi di risposta. Non catturando intestazioni e corpo della risposta, è possibile impedire la registrazione di dati sensibili.

Ogni richiesta in un canary dell'API comprende le seguenti informazioni:

- L'endpoint, ovvero l'URL richiesto.
- Il metodo, che è il tipo di richiesta che viene inviata al server. Le REST API supportano le operazioni GET (lettura), POST (scrittura), PUT (aggiornamento), PATCH (aggiornamento) e DELETE (eliminazione).
- Le intestazioni, che forniscono informazioni sia al client che al server. Vengono utilizzate per l'autenticazione e per fornire informazioni sul contenuto del corpo. Per un elenco di intestazioni valide, consulta Intestazioni HTTP.
- I dati (o corpo), che contengono informazioni da inviare al server. Questo viene utilizzato solo per le richieste POST, PUT, PATCH o DELETE.

Il blueprint del canary API supporta i metodi GET e POST. Quando utilizzi questo blueprint, devi specificare le intestazioni. Ad esempio, puoi specificare **`Authorization`** come chiave e specificare i dati di autorizzazione necessari come valore per tale chiave.

Se si sta testando una richiesta POST, si specifica anche il contenuto da pubblicare nel campo Dati .

Integrazione con API Gateway

Il blueprint API è integrato con Amazon API Gateway. Ciò ti consente di selezionare un'API Gateway API e lo stage dallo stesso AWS account e dalla stessa regione di Canary, oppure di caricare un modello Swagger da API Gateway per il monitoraggio delle API tra account e regioni diverse. Puoi quindi scegliere il resto dei dettagli nella console per creare il canary, invece di inserirli da zero. Per ulteriori informazioni su API Gateway, consulta Che cos'è Amazon API Gateway?

Utilizzo di un'API privata

È possibile creare un canary che utilizzi un'API privata in Amazon API Gateway. Per ulteriori informazioni, consulta la pagina Creazione di un'API privata in Amazon API Gateway?

Controllo collegamento interrotto

Il controllo dei collegamenti interrotti raccoglie tutti i collegamenti all'interno dell'URL che si sta testando tramite `document.getElementsByTagName('a')`. Verifica solo fino al numero di collegamenti specificato e l'URL stesso viene conteggiato come primo collegamento. Ad esempio, se si desidera controllare tutti i collegamenti in una pagina contenente cinque collegamenti, è necessario specificare che il canary segua sei collegamenti.

I canary di controllo del collegamento interrotto creati utilizzando la versione runtime `syn-nodejs-2.0-beta` o versioni successive supportano le caratteristiche aggiuntive elencate di seguito:

- Fornisce un report che include i collegamenti controllati, il codice di stato, il motivo dell'errore (se presente) e le schermate della pagina di origine e di destinazione.
- Quando si visualizzano i risultati del canary, è possibile filtrarli per visualizzare solo i collegamenti interrotti e quindi correggere il collegamento in base al motivo dell'errore.
- Questa versione cattura screenshot della pagina sorgente annotata per ogni collegamento ed evidenzia l'ancoraggio in cui è stato trovato il collegamento. I componenti nascosti non vengono annotati.
- Puoi configurare questa versione per acquisire screenshot di entrambe le pagine di origine e di destinazione, solo pagine di origine o solo pagine di destinazione.
- Questa versione risolve un problema nella versione precedente in cui lo script canary si interrompe dopo il primo collegamento interrotto anche quando più collegamenti vengono estratti dalla prima pagina.

Se si desidera aggiornare un canary esistente utilizzando `syn-1.0` per utilizzare il nuovo runtime, devi eliminare e ricreare il canary. L'aggiornamento di un canary esistente al nuovo runtime non rende disponibili queste funzionalità.

Un canary di controllo del collegamento interrotto rileva i seguenti tipi di errori di collegamento:

- 404 Pagina non trovata
- Nome host non valido
- URL errato. Ad esempio, nell'URL manca una parentesi, ha barre aggiuntive o è il protocollo sbagliato.
- Codice di risposta HTTP non valido.
- Il server host restituisce risposte vuote senza contenuto e senza codice di risposta.
- Le richieste HTTP vanno costantemente in timeout durante l'esecuzione del canary.
- L'host elimina costantemente le connessioni perché non è configurato correttamente o è troppo occupato.

Blueprint di monitoraggio visivo

Il blueprint di monitoraggio visivo include il codice per confrontare gli screenshot acquisiti durante un'esecuzione del canary con gli screenshot acquisiti durante un'esecuzione del canary di riferimento. Se la discrepanza tra i due screenshot supera una percentuale di soglia, il canary fallisce. Il monitoraggio visivo è supportato nei canari con versione syn-puppeteer-node-3.2 e versioni successive. Attualmente non è supportato nei canary che eseguono Python e Selenium.

Il blueprint di monitoraggio visivo include la seguente riga di codice nello script canary del blueprint predefinito, che consente il monitoraggio visivo.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

La prima volta che il canary viene eseguito correttamente dopo che questa riga è stata aggiunta allo script, utilizza gli screenshot acquisiti durante l'esecuzione come riferimento per il confronto. Dopo la prima esecuzione del canarino, puoi usare la CloudWatch console per modificare il canarino per eseguire una delle seguenti operazioni:

- Imposta l'esecuzione successiva del canary come nuovo riferimento.
- Disegna dei limiti sullo screenshot di riferimento corrente per designare le aree dello screenshot da ignorare durante i confronti visivi.

- Rimuovi uno screenshot dall'utilizzo per il monitoraggio visivo.

Per ulteriori informazioni sull'utilizzo della CloudWatch console per modificare un canarino, consulta. Modifica o eliminazione di un canary

Puoi anche modificare la canary run utilizzata come linea di base utilizzando i `lastrun` parametri `nextrun` o o specificando un ID Canary Run nell'API. UpdateCanary

Quando utilizzi il blueprint di monitoraggio visivo, immetti l'URL in cui desideri acquisire lo screenshot e specifica una soglia di differenza come percentuale. Dopo l'esecuzione di riferimento, le esecuzioni future del canary che rilevano una differenza visiva maggiore di quella soglia attivano un errore del canary. Dopo l'esecuzione di riferimento, puoi anche modificare il canary per “disegnare” i limiti sullo screenshot della linea di base che desideri ignorare durante il monitoraggio visivo.

La funzionalità di monitoraggio visivo è alimentata dal toolkit software ImageMagick open source. Per ulteriori informazioni, vedere ImageMagick.

Registratore di Canary

Con il progetto Canary Recorder, è possibile utilizzare CloudWatch Synthetics Recorder per registrare le azioni di clic e digitazione su un sito Web e generare automaticamente uno script Node.js che può essere utilizzato per creare un canarino che segue gli stessi passaggi. CloudWatchSynthetics Recorder è un'estensione di Google Chrome fornita da Amazon.

Crediti: Il CloudWatch Synthetics Recorder è basato sul registratore Headless.

Per ulteriori informazioni, consulta Utilizzo del CloudWatch Synthetics Recorder per Google Chrome.

Generatore di flussi di lavoro GUI

Il blueprint di GUI Workflow Builder verifica che le azioni possono essere eseguite nella pagina Web. Ad esempio, se si dispone di una pagina Web con un modulo di accesso, il canary può compilare i campi utente e password e inviare il modulo per verificare che la pagina Web funzioni correttamente.

Quando utilizzi un blueprint per creare questo tipo di canary, si specificano le azioni che si desidera che il canary esegua nella pagina Web. Le azioni che è possibile utilizzare sono le seguenti:

- Click (Fai clic): seleziona l'elemento specificato e simula un utente che fa clic o sceglie l'elemento.

  Per specificare l'elemento in uno script Node.js, utilizza `[id=]` o `a[class=]`.

Per specificare l'elemento in uno script Python, utilizza `xpath //*[@id=]` o `//*[@class=]`.

- Verify selector (Verifica selettore): verifica che l'elemento specificato esista nella pagina Web. Questo test è utile per verificare che un'azione precedente ha permesso agli elementi corretti di popolare la pagina.

  Per specificare l'elemento per verificare uno script Node.js, utilizza `[id=]` o `a[class=]`.

  Per specificare l'elemento per verificare uno script Python, utilizza `xpath //*[@id=]` o `//*[class=]`.

- Verify text (Verifica testo): verifica che la stringa specificata sia contenuta all'interno dell'elemento di destinazione. Questo test è utile per verificare che un'azione precedente abbia portato alla visualizzazione del testo corretto.

  Per specificare l'elemento un uno script Node.js, utilizza un formato ad esempio `div[@id=]//h1`, perché questa azione utilizza la funzione `waitForXPath` in Puppeteer.

  Per specificare l'elemento in uno script Python, utilizza il formato xpath ad esempio `//*[@id=]` o //*[@class=] perché questa azione utilizza la funzione `implicitly_wait` in Selenium.

- Input text (Testo di input): scrive il testo specificato nell'elemento di destinazione.

  Per specificare l'elemento per verificare uno script Node.js, utilizza `[id=]` o `a[class=]`.

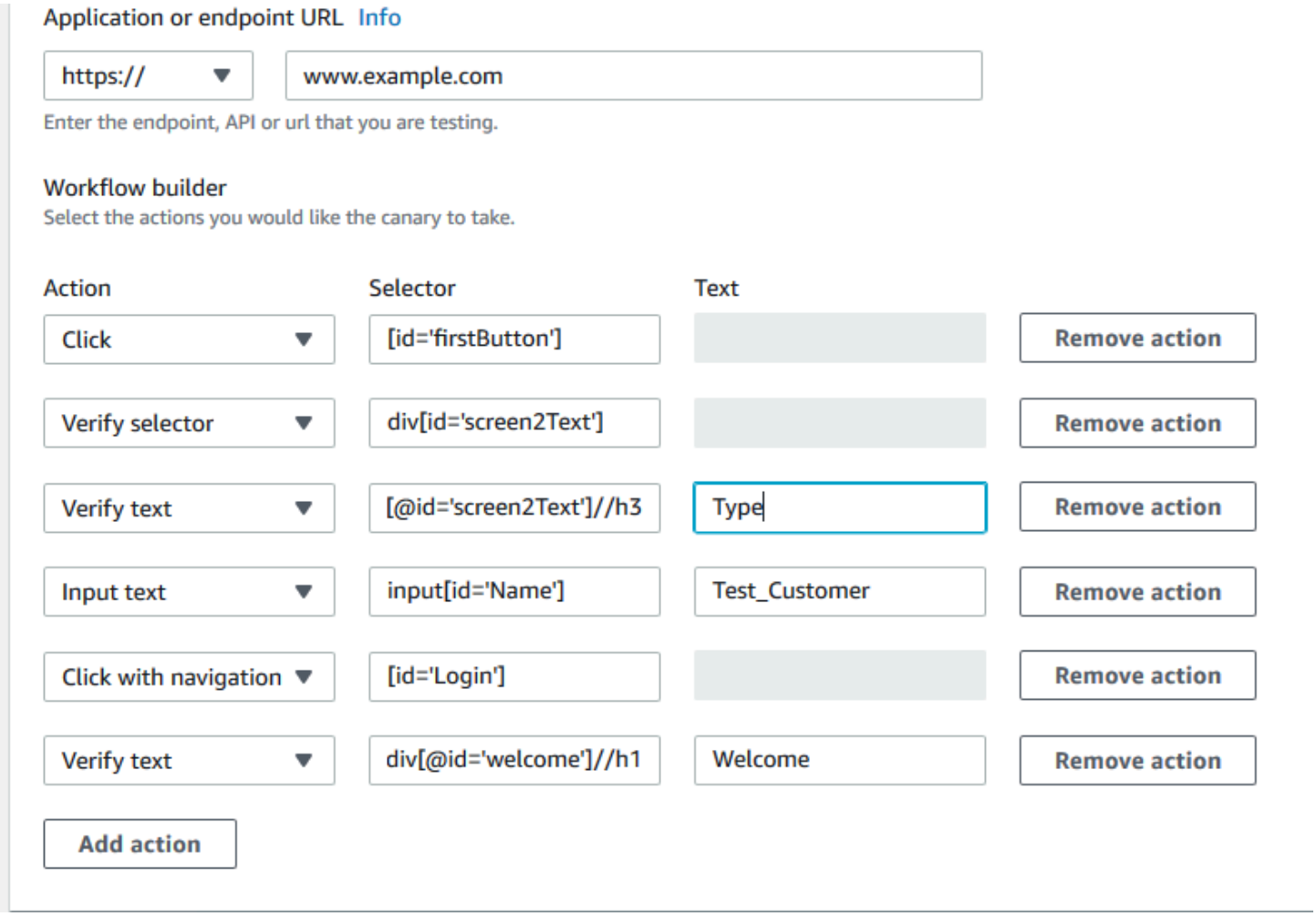
  Per specificare l'elemento per verificare uno script Python, utilizza `xpath //*[@id=]` o `//*[@class=]`.

- Click with navigation (Fai clic con la navigazione): attende il caricamento dell'intera pagina dopo aver scelto l'elemento specificato. Questo è molto utile quando è necessario ricaricare la pagina.

  Per specificare l'elemento in uno script Node.js, utilizza `[id=]` o `a[class=]`.

  Per specificare l'elemento in uno script Python, utilizza `xpath //*[@id=]` o `//*[@class=]`.

Ad esempio, il blueprint riportato di seguito utilizza Node.js. Fa clic sul pulsante firstButton nell'URL specificato, verifica che venga visualizzato il selettore previsto con il testo previsto, immette il nome `Test_Customer` nel campo Name (Nome), fa clic sul pulsante Login (Accedi) e verifica quindi che l'accesso abbia esito positivo controllando il testo di benvenuto nella pagina successiva.



I canary del flusso di lavoro GUI che utilizzano i seguenti runtime forniscono anche un riepilogo dei passaggi eseguiti per ogni esecuzione del canary. Puoi utilizzare le schermate e il messaggio di errore associati a ogni passaggio per trovare la causa principale dell'errore.

- `syn-nodejs-2.0` o versione successiva
- `syn-python-selenium-1.0` o versione successiva

Utilizzo del CloudWatch Synthetics Recorder per Google Chrome

Amazon fornisce un CloudWatch Synthetics Recorder per aiutarti a creare canarini più facilmente. Il registratore è un'estensione di Google Chrome.

Il registratore registra le operazioni di clic e digitazione su un sito Web e genera automaticamente uno script Node.js che è possibile utilizzare per creare un canary che segue gli stessi passaggi.

Dopo aver avviato la registrazione, CloudWatch Synthetics Recorder rileva le tue azioni nel browser e le converte in uno script. È possibile sospendere e riprendere la registrazione in base alle esigenze. Quando si interrompe la registrazione, il registratore produce uno script Node.js delle operazioni, che è possibile copiare facilmente con il comando Copy to Clipboard (Copia negli appunti). È quindi possibile utilizzare questo script per creare un canarino in CloudWatch Synthetics.

Crediti: Il CloudWatch Synthetics Recorder è basato sul registratore Headless.

Installazione dell'estensione CloudWatch Synthetics Recorder per Google Chrome

Per utilizzare il CloudWatch Synthetics Recorder, puoi iniziare a creare un canarino e scegliere il progetto Canary Recorder. Se lo fai quando non hai ancora scaricato il registratore, la console CloudWatch Synthetics fornisce un link per scaricarlo.

In alternativa, puoi eseguire questa procedura per scaricare e installare direttamente il registratore.

Per installare il CloudWatch Synthetics Recorder

1. Utilizzando Google Chrome, visitate questo sito Web: https://chrome.google.com/webstore/detail/ /bhdnlmmgiplmbcdmkkdfplenecpegfno cloudwatch-synthetics-rec
2. Scegli Add to Chrome (Aggiungi a Chrome), quindi scegli Add extension (Aggiungi estensione).

Utilizzo del CloudWatch Synthetics Recorder per Google Chrome

Per utilizzare CloudWatch Synthetics Recorder per aiutarti a creare un canarino, puoi scegliere Crea canarino nella console, quindi scegliere Usa un CloudWatch progetto, Canary Recorder. Per ulteriori informazioni, consulta Creazione di un Canary.

In alternativa, puoi utilizzare il registratore per registrare i passaggi senza utilizzarli immediatamente per creare un canary.

Per utilizzare il CloudWatch Synthetics Recorder per registrare le azioni dell'utente su un sito Web

1. Passa alla pagina che desideri monitorare.
2. Scegli l'icona delle estensioni di Chrome, quindi scegli CloudWatchSynthetics Recorder.
3. Scegli Start Recording (Avvia registrazione).
4. Esegui i passaggi che vuoi+ registrare. Per sospendere la registrazione, scegli Pause (Pausa).
5. Al termine della registrazione del flusso di lavoro, scegli Stop recording (Interrompi la registrazione).

6. Scegli Copy to clipboard (Copia negli appunti) per copiare lo script generato negli appunti. Oppure, se si desidera ricominciare da capo, scegli New recording (Nuova registrazione).
7. Per creare un canary con lo script copiato, puoi incollare lo script copiato nell'editor inline del blueprint del registratore o salvarlo in un bucket Amazon S3 e importarlo da lì.
8. Se non crei immediatamente un canary, puoi salvare lo script registrato in un file.

Limitazioni note del CloudWatch Synthetics Recorder

Il CloudWatch Synthetics Recorder per Google Chrome presenta attualmente le seguenti limitazioni.

- Gli elementi HTML che non hanno ID utilizzeranno selettori CSS. Questo può interrompere i canary se la struttura della pagina Web cambia in seguito. Abbiamo in programma di fornire alcune opzioni di configurazione (ad esempio l'uso di data-id) in una versione futura del registratore.
- Il registratore non supporta azioni quali doppio clic o copia/incolla e non supporta combinazioni di tasti ad esempio CMD+0.
- Per verificare la presenza di un elemento o di un testo nella pagina, gli utenti devono aggiungere asserzioni dopo la generazione dello script. Il registratore non supporta la verifica di un elemento senza eseguire alcuna operazione su tale elemento. Questo è simile alle opzioni “Verify text” (Verifica testo) o “Verify element” (Verifica elemento) nel generatore di flussi di lavoro del canary. Abbiamo in programma di aggiungere il supporto delle asserzioni in una versione futura del registratore.
- Il registratore registra tutte le operazioni nella scheda in cui viene avviata la registrazione. Non registra i popup (ad esempio, per consentire il tracciamento della posizione) o la navigazione a pagine diverse dai popup.

Versioni di runtime Synthetics

Quando crei o aggiorni un Canary, scegli una versione di runtime Synthetics per il Canary. Un runtime Synthetics è una combinazione del codice Synthetics che chiama il gestore di script e dei livelli Lambda delle dipendenze in bundle.

CloudWatch Synthetics attualmente supporta runtime che utilizzano Node.js per gli script e il framework Puppeteer e runtime che utilizzano Python per lo scripting e Selenium Webdriver per il framework.

Ti consigliamo di utilizzare sempre la versione di runtime più recente per i Canary in modo da poter utilizzare le funzionalità e gli aggiornamenti più recenti apportati alla libreria Synthetics.

Quando create un canarino, uno dei livelli creati è un livello Synthetics a cui fa preposto. `Synthetics` Questo livello è di proprietà dell'account del servizio Synthetics e contiene il codice di runtime.

> **Note**
>
> Ogni volta che aggiorni un canary per utilizzare una nuova versione del runtime Synthetics, tutte le funzioni della libreria Synthetics utilizzate dal canary vengono aggiornate automaticamente alla stessa versione di NodeJS supportata dal runtime Synthetics.

Argomenti

CloudWatch Politica di supporto per il runtime di Synthetics

Le versioni runtime di Synthetics sono soggette ad aggiornamenti di manutenzione e sicurezza. Quando un componente di una versione runtime non è più supportato, la versione runtime di Synthetics viene resa obsoleta.

Non è possibile creare nuovi canary utilizzando versioni runtime obsolete. I canary che utilizzano runtime obsoleti continueranno a funzionare. Puoi interrompere, avviare ed eliminare questi Canary. Puoi aggiornare un canary esistente che utilizza versioni runtime obsolete aggiornando il canary per utilizzare una versione di runtime supportata.

CloudWatch Synthetics ti avvisa via e-mail se hai canari che utilizzano runtime programmati per essere obsoleti nei prossimi 60 giorni. Ti consigliamo di eseguire la migrazione dei canary a una versione runtime supportata per trarre vantaggio dai nuovi miglioramenti alle funzionalità, alla sicurezza e alle prestazioni inclusi nelle versioni più recenti.

Come posso aggiornare un canary a una nuova versione di runtime?

Puoi aggiornare la versione di runtime di Canary utilizzando la console, o l'SDK. CloudWatch AWS CloudFormation AWS CLI AWS Quando usi la CloudWatch console, puoi aggiornare fino a cinque canarini contemporaneamente selezionandoli nella pagina dell'elenco dei canarini e quindi scegliendo Azioni, Aggiorna Runtime.

Puoi verificare l'aggiornamento clonando prima il canarino utilizzando la CloudWatch console e aggiornando la sua versione di runtime. Questo crea un altro canary che è un clone del tuo canary originale. Dopo aver verificato il canary con la nuova versione di runtime, puoi aggiornare la versione di runtime del canary originale ed eliminare il canary clone.

Puoi anche aggiornare più canary utilizzando uno script di aggiornamento. Per ulteriori informazioni, consulta Script di aggiornamento del runtime del canary.

Se aggiorni un canary e l'operazione non va a buon fine, consulta Risoluzione dei problemi di un canary fallito.

Date di deprecazione del runtime

Versione di runtime	Data di deprecazione
`syn-nodejs-puppeteer-6.1`	8 marzo 2024
`syn-nodejs-puppeteer-6.0`	8 marzo 2024
`syn-nodejs-puppeteer-5.1`	8 marzo 2024
`syn-nodejs-puppeteer-5.0`	8 marzo 2024
`syn-nodejs-puppeteer-4.0`	8 marzo 2024
`syn-nodejs-puppeteer-3.9`	8 gennaio 2024
`syn-nodejs-puppeteer-3.8`	8 gennaio 2024
`syn-python-selenium-2.0`	8 marzo 2024

Versione di runtime	Data di deprecazione
`syn-python-selenium-1.3`	8 marzo 2024
`syn-python-selenium-1.2`	8 marzo 2024
`syn-python-selenium-1.1`	8 marzo 2024
`syn-python-selenium-1.0`	8 marzo 2024
`syn-nodejs-puppeteer-3.7`	8 gennaio 2024
`syn-nodejs-puppeteer-3.6`	8 gennaio 2024
`syn-nodejs-puppeteer-3.5`	8 gennaio 2024
`syn-nodejs-puppeteer-3.4`	13 novembre 2022
`syn-nodejs-puppeteer-3.3`	13 novembre 2022
`syn-nodejs-puppeteer-3.2`	13 novembre 2022
`syn-nodejs-puppeteer-3.1`	13 novembre 2022
`syn-nodejs-puppeteer-3.0`	13 novembre 2022
`syn-nodejs-2.2`	28 maggio 2021

Versione di runtime	Data di deprecazione
`syn-nodejs-2.1`	28 maggio 2021
`syn-nodejs-2.0`	28 maggio 2021
`syn-nodejs-2.0-beta`	8 febbraio 2021
`syn-1.0`	28 maggio 2021

Script di aggiornamento del runtime del canary

Per aggiornare uno script canary a una versione di runtime supportata, utilizza lo script seguente.

```
const AWS = require('aws-sdk');

// You need to configure your AWS credentials and Region.
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-
credentials-node.html
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-
region.html

const synthetics = new AWS.Synthetics();

const DEFAULT_OPTIONS = {
  /**
   * The number of canaries to upgrade during a single run of this script.
   */
  count: 10,
  /**
   * No canaries are upgraded unless force is specified.
   */
  force: false
};

/**
 * The number of milliseconds to sleep between GetCanary calls when
 * verifying that an update succeeded.
 */
const SLEEP_TIME = 5000;
```

```
(async () => {
  try {
    const options = getOptions();

    const versions = await getRuntimeVersions();
    const canaries = await getAllCanaries();
    const upgrades = canaries
      .filter(canary => !versions.isLatestVersion(canary.RuntimeVersion))
      .map(canary => {
        return {
          Name: canary.Name,
          FromVersion: canary.RuntimeVersion,
          ToVersion: versions.getLatestVersion(canary.RuntimeVersion)
        };
      });

    if (options.force) {
      const promises = [];

      for (const upgrade of upgrades.slice(0, options.count)) {
        const promise = upgradeCanary(upgrade);
        promises.push(promise);
        // Sleep for 100 milliseconds to avoid throttling.
        await usleep(100);
      }

      const succeeded = [];
      const failed = [];
      for (let i = 0; i < upgrades.slice(0, options.count).length; i++) {
        const upgrade = upgrades[i];
        const promise = promises[i];
        try {
          await promise;
          console.log(`The update of ${upgrade.Name} succeeded.`);
          succeeded.push(upgrade.Name);
        } catch (e) {
          console.log(`The update of ${upgrade.Name} failed with error: ${e}`);
          failed.push({
            Name: upgrade.Name,
            Reason: e
          });
        }
      }
```

```
      if (succeeded.length) {
        console.group('The following canaries were upgraded successfully.');
        for (const name of succeeded) {
          console.log(name);
        }
        console.groupEnd()
      } else {
        console.log('No canaries were upgraded successfully.');
      }

      if (failed.length) {
        console.group('The following canaries were not upgraded successfully.');
        for (const failure of failed) {
          console.log('\x1b[31m', `${failure.Name}: ${failure.Reason}`, '\x1b[0m');
        }
        console.groupEnd();
      }
    } else {
      console.log('Run with --force [--count <count>] to perform the first <count>
 upgrades shown. The default value of <count> is 10.')
      console.table(upgrades);
    }
  } catch (e) {
    console.error(e);
  }
})();

function getOptions() {
  const force = getFlag('--force', DEFAULT_OPTIONS.force);
  const count = getOption('--count', DEFAULT_OPTIONS.count);
  return { force, count };

  function getFlag(key, defaultValue) {
    return process.argv.includes(key) || defaultValue;
  }
  function getOption(key, defaultValue) {
    const index = process.argv.indexOf(key);
    if (index < 0) {
      return defaultValue;
    }
    const value = process.argv[index + 1];
    if (typeof value === 'undefined' || value.startsWith('-')) {
      throw `The ${key} option requires a value.`;
    }
```

```
    return value;
  }
}

function getAllCanaries() {
  return new Promise((resolve, reject) => {
    const canaries = [];

    synthetics.describeCanaries().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
        if (data === null) {
          resolve(canaries);
        } else {
          canaries.push(...data.Canaries);
        }
      }
    });
  });
}

function getRuntimeVersions() {
  return new Promise((resolve, reject) => {
    const jsVersions = [];
    const pythonVersions = [];
    synthetics.describeRuntimeVersions().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
        if (data === null) {
          jsVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          pythonVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          resolve({
            isLatestVersion(version) {
              const latest = this.getLatestVersion(version);
              return latest === version;
            },
            getLatestVersion(version) {
              if (jsVersions.some(v => v.VersionName === version)) {
                return jsVersions[jsVersions.length - 1].VersionName;
              } else if (pythonVersions.some(v => v.VersionName === version)) {
                return pythonVersions[pythonVersions.length - 1].VersionName;
              } else {
```

```
                throw Error(`Unknown version ${version}`);
              }
            }
          });
        } else {
          for (const version of data.RuntimeVersions) {
            if (version.VersionName === 'syn-1.0') {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-2.')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-puppeteer-')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-python-selenium-')) {
              pythonVersions.push(version);
            } else {
              throw Error(`Unknown version ${version.VersionName}`);
            }
          }
        }
      });
  });
}

async function upgradeCanary(upgrade) {
  console.log(`Upgrading canary ${upgrade.Name} from ${upgrade.FromVersion} to
 ${upgrade.ToVersion}`);
  await synthetics.updateCanary({ Name: upgrade.Name, RuntimeVersion:
 upgrade.ToVersion }).promise();
  while (true) {
    await usleep(SLEEP_TIME);
    console.log(`Getting the state of canary ${upgrade.Name}`);
    const response = await synthetics.getCanary({ Name: upgrade.Name }).promise();
    const state = response.Canary.Status.State;
    console.log(`The state of canary ${upgrade.Name} is ${state}`);
    if (state === 'ERROR' || response.Canary.Status.StateReason) {
      throw response.Canary.Status.StateReason;
    }
    if (state !== 'UPDATING') {
      return;
    }
  }
}
```

```
function usleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}
```

Versioni runtime che utilizzano Node.js e Puppeteer

La prima versione di runtime per Node.js e Puppeteer è stata denominata `syn-1.0`.
Le versioni di runtime successive hanno la convenzione di denominazione `syn-`*`language`*`-`*`majorversion`*`.`*`minorversion`*. Iniziando con `syn-nodejs-puppeteer-3.0`, la convenzione di denominazione è `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*

Un ulteriore suffisso `-beta` indica che la versione di runtime è attualmente in una versione di anteprima beta.

Le versioni di runtime con lo stesso numero di versione principale sono sempre compatibili con le versioni precedenti.

Important

È previsto che le seguenti CloudWatch versioni di runtime di Synthetics diventino obsolete l'8 marzo 2024.

- `syn-nodejs-puppeteer-6.1`
- `syn-nodejs-puppeteer-6.0`
- `syn-nodejs-puppeteer-5.1`
- `syn-nodejs-puppeteer-5.0`
- `syn-nodejs-puppeteer-4.0`

Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Important

IMPORTANTE: l' AWS SDK incluso per la dipendenza JavaScript v2 verrà rimosso e aggiornato per utilizzare AWS SDK per JavaScript v3 in una futura versione di runtime. Quando ciò accade, puoi aggiornare i riferimenti al codice del canary. In alternativa, puoi

continuare a fare riferimento e utilizzare l' AWS SDK incluso per la dipendenza JavaScript v2 aggiungendolo come dipendenza al file zip del codice sorgente.

Note per tutte le versioni di runtime

Quando utilizzi la versione di runtime `syn-nodejs-puppeteer-3.0` assicurati che lo script canary sia compatibile con Node.js 12.x. Se usi una versione precedente di un runtime `syn-nodejs`, assicurati che lo script sia compatibile con Node.js 10.x.

Il codice Lambda in un Canary è configurato per avere una memoria massima di 1 GB. Ogni esecuzione di Canary esegue il timeout dopo il valore di timeout configurato. Se non viene specificato alcun valore di timeout per un canarino, CloudWatch sceglie un valore di timeout basato sulla frequenza del canarino. Se si configura un valore di timeout, non è inferiore a 15 secondi per consentire l'avvio a freddo Lambda e il tempo necessario per avviare la strumentazione canary.

Note

Le seguenti CloudWatch versioni di runtime di Synthetics sono state dichiarate obsolete l'8 gennaio 2024. Questo perché il 4 AWS Lambda dicembre 2023 ha reso obsoleto il runtime Lambda Node.js 14.

- `syn-nodejs-puppeteer-3.9`
- `syn-nodejs-puppeteer-3.8`
- `syn-nodejs-puppeteer-3.7`
- `syn-nodejs-puppeteer-3.6`
- `syn-nodejs-puppeteer-3.5`

Le seguenti CloudWatch versioni di runtime di Synthetics sono state dichiarate obsolete il 13 novembre 2022. Questo perché il runtime Lambda Node.js 12 è stato reso AWS Lambda obsoleto il 14 novembre 2022.

- `syn-nodejs-puppeteer-3.4`
- `syn-nodejs-puppeteer-3.3`
- `syn-nodejs-puppeteer-3.2`
- `syn-nodejs-puppeteer-3.1`
- `syn-nodejs-puppeteer-3.0`

Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

syn-nodejs-puppeteer-7,0

Il `syn-nodejs-puppeteer-7.0` runtime è la versione di runtime più recente per il runtime Lambda Node.js 18.x. Utilizza Node.js e Puppeteer.

Principali dipendenze:

- Runtime Lambda Node.js 18.x
- Puppeteer-core versione 21.9.0
- Versione Chromium 121.0.6167.139

Dimensione del codice:

La dimensione del codice e delle dipendenze che è possibile inserire in questo runtime è di 80 MB.

Nuove funzionalità in syn-nodejs-puppeteer -7.0:

- Versioni aggiornate delle librerie in bundle in Puppeteer e Chromium: le dipendenze Puppeteer e Chromium vengono aggiornate alle nuove versioni.

> **Important**
>
> Il passaggio da Puppeteer 19.7.0 a Puppeteer 21.9.0 introduce modifiche importanti per quanto riguarda test e filtri. Per ulteriori informazioni, consulta le sezioni BREAKING CHANGES in puppeteer: v20.0.0 e puppeteer-core: v21.0.0.

Aggiornamento AWS consigliato a SDK v3

Il runtime Lambda nodejs18.x non supporta SDK v2. AWS Ti consigliamo vivamente di migrare a SDK v3. AWS

syn-nodejs-puppeteer-6.2

Principali dipendenze:

- Runtime Lambda Node.js 18.x
- Puppeteer-core versione 19.7.0
- Chromium versione 111.0.5563.146

Nuove funzionalità in -6.2: syn-nodejs-puppeteer

- Versioni aggiornate delle librerie in bundle in Chromium
- Monitoraggio temporaneo dello storage: questo runtime aggiunge il monitoraggio temporaneo dello storage negli account dei clienti.
- Correzioni di bug

syn-nodejs-puppeteer-5.2

Il `syn-nodejs-puppeteer-5.2` runtime è la versione di runtime più recente per il runtime Lambda Node.js 16.x. Utilizza Node.js e Puppeteer.

Principali dipendenze:

- Runtime Lambda Node.js 16.x
- Puppeteer-core versione 19.7.0
- Chromium versione 111.0.5563.146

Nuove funzionalità in -5.2 syn-nodejs-puppeteer:

- Versioni aggiornate delle librerie in bundle in Chromium
- Correzioni di bug

syn-nodejs-puppeteer-6.1

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 18.x
- Puppeteer-core versione 19.7.0
- Chromium versione 111.0.5563.146

Nuove funzionalità in -6.1: syn-nodejs-puppeteer

- Miglioramenti della stabilità: aggiunta la logica di ripetizione automatica per la gestione degli errori di avvio intermittenti di Puppeteer.
- Aggiornamenti delle dipendenze: aggiornamenti di alcuni pacchetti di dipendenza di terze parti.
- Canary senza autorizzazioni Amazon S3: correzioni di bug, in modo che i canary che non dispongono di autorizzazioni Amazon S3 possano funzionare ugualmente. Questi canary senza autorizzazioni Amazon S3 non saranno in grado di caricare screenshot o altri artefatti in Amazon S3. Per ulteriori informazioni sulle autorizzazioni per i canary, consulta Ruoli e autorizzazioni richiesti per i canary.

> **Important**
>
> IMPORTANTE: l' AWS SDK incluso per la dipendenza JavaScript v2 verrà rimosso e aggiornato per utilizzare AWS SDK per JavaScript v3 in una futura versione di runtime. Quando ciò accade, puoi aggiornare i riferimenti al codice del canary. In alternativa, puoi continuare a fare riferimento e utilizzare l' AWS SDK incluso per la dipendenza JavaScript v2 aggiungendolo come dipendenza al file zip del codice sorgente.

syn-nodejs-puppeteer-6,0

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 18.x
- Puppeteer-core versione 19.7.0

- Chromium versione 111.0.5563.146

Nuove funzionalità nella versione -6.0: syn-nodejs-puppeteer

- Aggiornamento della dipendenza: la dipendenza Node.js è aggiornata alla versione 18.x.
- Supporto della modalità di intercettazione: il supporto della modalità di intercettazione cooperativa Puppeteer è stato aggiunto alla libreria di runtime di canary Synthetics.
- Modifica del comportamento di tracciamento: il comportamento di tracciamento predefinito è stato modificato per tracciare solo le richieste di recupero e xhr senza più tracciare le richieste di risorse. È possibile abilitare il tracciamento delle richieste di risorse configurando l'opzione `traceResourceRequests`.
- Metrica della durata perfezionata: la `Duration` metrica ora esclude il tempo operativo utilizzato da Canary per caricare artefatti, scattare schermate e generare metriche. CloudWatch `Duration`i valori delle metriche vengono riportati a CloudWatch e puoi anche visualizzarli nella console Synthetics.
- Correzione di bug: cancella il core dump generato quando Chromium si blocca durante un'esecuzione di un canary.

> **Important**
>
> IMPORTANTE: l' AWS SDK incluso per la dipendenza JavaScript v2 verrà rimosso e aggiornato per utilizzare AWS SDK per JavaScript v3 in una futura versione di runtime. Quando ciò accade, puoi aggiornare i riferimenti al codice del canary. In alternativa, puoi continuare a fare riferimento e utilizzare l' AWS SDK incluso per la dipendenza JavaScript v2 aggiungendolo come dipendenza al file zip del codice sorgente.

syn-nodejs-puppeteer-5,1

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 16.x
- Puppeteer-core versione 19.7.0
- Chromium versione 111.0.5563.146

Correzioni di bug in -5.1: syn-nodejs-puppeteer

- Correzione di bug: questo runtime corregge un bug in `syn-nodejs-puppeteer-5.0` dove i file HAR creati dai canary non avevano le intestazioni di richiesta.

syn-nodejs-puppeteer-5.0

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 16.x
- Puppeteer-core versione 19.7.0
- Chromium versione 111.0.5563.146

Nuove funzionalità in -5.0: syn-nodejs-puppeteer

- Aggiornamento della dipendenza: la versione di Puppeteer-core è aggiornata alla 19.7.0. La versione Chromium è aggiornata a 111.0.5563.146.

> **Important**
>
> La nuova versione di Puppeteer-core non è compatibile con le versioni precedenti di Puppeteer. Alcune delle modifiche apportate a questa versione possono causare il fallimento dei canary esistenti che utilizzano funzioni Puppeteer obsolete. Per ulteriori informazioni, consulta le ultime modifiche nei log delle modifiche per le versioni di Puppeteer-core da 19.7.0 a 6.0 nei log delle modifiche di Puppeteer.

syn-nodejs-puppeteer-4,0

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 16.x
- Puppeteer-core versione 5.5.0
- Chromium versione 92.0.4512

Nuove funzionalità in -4.0: syn-nodejs-puppeteer

- Aggiornamento della dipendenza: la dipendenza Node.js è aggiornata alla versione 16.x.

Runtime obsoleti per Node.js e Puppeteer

I seguenti runtime per Node.js e Puppeteer sono obsoleti.

syn-nodejs-puppeteer-3,9

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta l'8 gennaio 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 14.x
- Puppeteer-core versione 5.5.0
- Chromium versione 92.0.4512

Nuove funzionalità in -3.9: syn-nodejs-puppeteer

- Aggiornamenti delle dipendenze: aggiorna alcuni pacchetti di dipendenza di terze parti.

syn-nodejs-puppeteer-3,8

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta l'8 gennaio 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 14.x
- Puppeteer-core versione 5.5.0
- Chromium versione 92.0.4512

Nuove funzionalità in -3.8: syn-nodejs-puppeteer

- Pulizia del profilo: i profili Chromium vengono ora puliti dopo ogni esecuzione iella canary.

Correzioni di bug in -3.8 syn-nodejs-puppeteer:

- Correzioni di bug: in precedenza, il monitoraggio visivo per le canary talvolta non funzionava correttamente dopo un'esecuzione senza screenshot. Questo problema è stato risolto.

syn-nodejs-puppeteer-3.7

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta l'8 gennaio 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 14.x
- Puppeteer-core versione 5.5.0

- Chromium versione 92.0.4512

Nuove funzionalità in -3.7: syn-nodejs-puppeteer

- Miglioramento della registrazione: il canary carica i registri su Amazon S3 anche in caso di timeout o arresti anomali.
- Dimensione ridotta per il livello Lambda: la dimensione del livello Lambda utilizzato per i canary è del 34% inferiore.

Correzioni di bug in -3.7 syn-nodejs-puppeteer:

- Correzioni di bug: i caratteri giapponesi, cinesi semplificati e cinesi tradizionali vengono visualizzati correttamente.

syn-nodejs-puppeteer-3.6

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta l'8 gennaio 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 14.x
- Puppeteer-core versione 5.5.0
- Chromium versione 92.0.4512

Nuove funzionalità nella versione -3.6: syn-nodejs-puppeteer

- Timestamp più precisi: l'ora di inizio e l'ora di fine delle corse di canary sono ora precisi al millisecondo.

syn-nodejs-puppeteer-3,5

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta l'8 gennaio 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 14.x
- Puppeteer-core versione 5.5.0
- Chromium versione 92.0.4512

Nuove funzionalità in -3.5: syn-nodejs-puppeteer

- Dependencies aggiornate— Le uniche nuove funzionalità di questo runtime sono le dipendenze aggiornate.

syn-nodejs-puppeteer-3,4

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta il 13 novembre 2022. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 12.x
- Puppeteer-core versione 5.5.0
- Chromium versione 88.0.4298.0

Nuove funzionalità in -3.4: syn-nodejs-puppeteer

- Funzione di gestore personalizzato: ora puoi usare una funzione di gestore personalizzato per gli script del tuo canary. I tempi di esecuzione precedenti hanno richiesto che il punto di ingresso dello script includesse `.handler`.

  Puoi inoltre inserire gli script del canary in qualsiasi cartella e passare il nome della cartella nel gestore. Ad esempio, è possibile utilizzare `MyFolder/MyScriptFile.functionname` come punto di ingresso.
- Informazioni sul file HAR esteso: ora puoi visualizzare richieste errate, in sospeso e incomplete nei file HAR prodotti dai canary.

syn-nodejs-puppeteer-3,3

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta il 13 novembre 2022. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 12.x
- Puppeteer-core versione 5.5.0
- Chromium versione 88.0.4298.0

Nuove funzionalità in -3.3: syn-nodejs-puppeteer

- Altre opzioni per la crittografia degli artefatti: per i canarini che utilizzano questo runtime o versioni successive, invece di utilizzare una chiave AWS gestita per crittografare gli artefatti che Canary archivia in Amazon S3, puoi scegliere di utilizzare AWS KMS una chiave gestita dal cliente o una chiave gestita da Amazon S3. Per ulteriori informazioni, consulta Crittografia di artefatti canary.

syn-nodejs-puppeteer-3.2

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta il 13 novembre 2022. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 12.x
- Puppeteer-core versione 5.5.0
- Chromium versione 88.0.4298.0

Nuove funzionalità in -3.2: syn-nodejs-puppeteer

- monitoraggio visivo con screenshot: i canary che utilizzano questo runtime o versioni successive possono confrontare uno screenshot scattato durante un'esecuzione con una versione di riferimento dello stesso screenshot. Se gli screenshot sono più diversi da una soglia percentuale specificata, il canary fallisce. Per ulteriori informazioni, consulta Monitoraggio visivo o Blueprint di monitoraggio visivo.
- Nuove funzioni relative ai dati sensibili: puoi impedire la visualizzazione di dati sensibili nei registri e nei report dei canary. Per ulteriori informazioni, consulta SyntheticsLogHelper classe.
- Funzione obsolete: la classe `RequestResponseLogHelper` è stata resa obsoleta da altre nuove opzioni di configurazione. Per ulteriori informazioni, consulta RequestResponseLogHelper classe.

syn-nodejs-puppeteer-3,1

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta il 13 novembre 2022. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 12.x

- Puppeteer-core versione 5.5.0
- Chromium versione 88.0.4298.0

Nuove funzionalità in -3.1: syn-nodejs-puppeteer

- Possibilità di configurare le CloudWatch metriche: con questo runtime, puoi disabilitare le metriche che non ti servono. Altrimenti, i canarini pubblicano diverse CloudWatch metriche per ogni corsa canaria.
- Collegamento dello screenshot: puoi collegare uno screenshot a un passaggio del canary dopo che il passaggio è stato completato. Per farlo, procedi nello screenshot tramite il metodo takeScreenshot usando il nome del passaggio a cui vuoi associare lo screenshot. Ad esempio, potresti voler eseguire un passaggio, aggiungere un tempo di attesa e quindi eseguire lo screenshot.
- Heartbeat monitor blueprint può monitorare più URL: puoi utilizzare il blueprint di monitoraggio del battito cardiaco nella CloudWatch console per monitorare più URL e visualizzare lo stato, la durata, le schermate associate e il motivo dell'errore di ciascun URL nel riepilogo dei passaggi del rapporto Canary Run.

syn-nodejs-puppeteer-3,0

> **Important**
>
> Questa versione di runtime è stata dichiarata obsoleta il 13 novembre 2022. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 12.x
- Puppeteer-core versione 5.5.0
- Chromium versione 88.0.4298.0

Nuove funzionalità in -3.0: syn-nodejs-puppeteer

- Dipendenze aggiornate: questa versione utilizza Puppeteer versione 5.5.0, Node.js 12.x e Chromium 88.0.4298.0.

- Accesso al bucket tra più regioni: ora puoi specificare un bucket S3 in un'altra regione come bucket in cui il canary memorizza i file di log, gli screenshot e i file HAR.
- Nuove funzioni disponibili: questa versione aggiunge funzioni di libreria per recuperare il nome del canary e la versione di runtime di Synthetics.

  Per ulteriori informazioni, consulta Classe Synthetics.

syn-nodejs-2.2

Questa sezione contiene informazioni relative alla versione di runtime `syn-nodejs-2.2`.

> **Important**
>
> Questa versione di runtime è stata resa obsoleta il 28 maggio 2021. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 10.x
- Puppeteer-core versione 3.3.0
- Chromium versione 83.0.4103.0

Nuove funzionalità in syn-nodejs-2.2:

- Monitora i tuoi canary come passaggi HTTP: ora puoi testare più API in un singolo canary. Ogni API viene testata come passaggio HTTP separato e CloudWatch Synthetics monitora lo stato di ogni passaggio utilizzando le metriche dei passaggi e il rapporto sui passaggi Synthetics. CloudWatch CloudWatch Synthetics `SuccessPercent` crea `Duration` e metrica per ogni passaggio HTTP.

  Questa funzionalità è implementata dalla funzione executeHttpStep(StepName, RequestOptions, callback, stepConfig). Per ulteriori informazioni, consulta executeHttpStep(StepName, RequestOptions, [callback], [StepConfig]).

  Il blueprint del canary API viene aggiornato per utilizzare questa nuova funzionalità.
- Report di richieste HTTP: ora puoi visualizzare report di richieste HTTP dettagliati che acquisiscono dettagli quali intestazioni della richiesta/risposta, corpo della risposta, codice di stato, intervalli

di errore e prestazioni, tempo di connessione TCP, tempo di handshake TLS, tempo del primo byte e tempo di trasferimento del contenuto. Tutte le richieste HTTP che utilizzano il modulo HTTP/HTTPS dietro le quinte vengono acquisite qui. Le intestazioni e il corpo della risposta non vengono acquisiti per impostazione predefinita, ma possono essere attivati impostando le opzioni di configurazione.

- Configurazione globale e a livello di fase: puoi impostare le configurazioni CloudWatch Synthetics a livello globale, che vengono applicate a tutti i passaggi di Canaries. Puoi anche sovrascrivere queste configurazioni a livello di fase passando coppie chiave/valore di configurazione per abilitare o disabilitare determinate opzioni.

  Per ulteriori informazioni, consulta SyntheticsConfiguration classe.

- Continuazione della configurazione di errore di fase: puoi scegliere di continuare l'esecuzione del canary quando una fase non va a buon fine. Per la funzione `executeHttpStep`, questa funzione è abilitata per impostazione predefinita. Puoi impostare questa opzione una volta a livello globale o impostarla in modo diverso per fase.

syn-nodejs-2.1

> **Important**
>
> Questa versione di runtime è stata resa obsoleta il 28 maggio 2021. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 10.x
- Puppeteer-core versione 3.3.0
- Chromium versione 83.0.4103.0

Nuove funzionalità in syn-nodejs-2.1:

- Comportamento dello screenshot configurabile: offre la possibilità di disattivare l'acquisizione di screenshot da parte dei canary dell'interfaccia utente. Nei canary che utilizzano versioni di runtime precedenti, i canary dell'interfaccia utente acquisiscono sempre screenshot prima e dopo ogni fase. Con `syn-nodejs-2.1`, questo comportamento è configurabile. La disattivazione degli screenshot

può ridurre i costi di stoccaggio di Amazon S3 e può aiutarti a rispettare le normative HIPAA. Per ulteriori informazioni, consulta SyntheticsConfiguration classe.

- Personalizzazione dei parametri di avvio di Google Chrome: ora puoi configurare gli argomenti utilizzati quando un canary avvia una finestra del browser Google Chrome. Per ulteriori informazioni, consulta launch(options).

Può verificarsi un piccolo aumento della durata del canary quando utilizzi syn-nodejs-2.0 o versioni successive, rispetto alle versioni di runtime precedenti del canary.

syn-nodejs-2.0

> **Important**
>
> Questa versione di runtime è stata resa obsoleta il 28 maggio 2021. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 10.x
- Puppeteer-core versione 3.3.0
- Chromium versione 83.0.4103.0

Nuove funzionalità in syn-nodejs-2.0:

- Dipendenze aggiornate: questa versione di runtime utilizza Puppeteer-core versione 3.3.0 e Chromium 83.0.4103.0
- Supporto del tracciamento attivo di X-Ray. Quando un canarino ha il tracciamento abilitato, vengono inviate tracce X-Ray per tutte le chiamate effettuate dal canarino che utilizzano il browser, l' AWS SDK o i moduli HTTP o HTTPS. I canary con tracciamento abilitato vengono visualizzati sulla mappa di tracciamento X-Ray, anche quando non inviano richieste ad altri servizi o applicazioni che hanno il tracciamento abilitato. Per ulteriori informazioni, consulta Canary e tracciamento X-Ray.
- Synthetics Reporting: per ogni corsa di Canary CloudWatch , Synthetics crea un `SyntheticsReport-PASSED.json` rapporto `SyntheticsReport-FAILED.json` denominato o che registra dati come ora di inizio, ora di fine, stato e guasti. Registra anche lo stato PASSED/FAILED di ogni fase dello script canary ed errori e screenshot acquisiti per ogni fase.

- Report sul controllo del collegamento interrotto: la nuova versione del controllo del collegamento interrotto incluso in questo runtime crea un report che include i collegamenti controllati, il codice di stato, il motivo dell'errore (se presente) e gli screenshot della pagina di origine e di destinazione.
- Nuove CloudWatch metriche: Synthetics pubblica metriche `2xx` denominate`4xx`,, `5xx` e nel namespace. `RequestFailed` `CloudWatchSynthetics` Questi parametri mostrano il numero di 200, 400, 500 e errori di richiesta nelle esecuzioni di canary. Con questa versione di runtime, questi parametri vengono segnalati solo per i canary dell'interfaccia utente e non vengono segnalati per i canary delle API. Vengono segnalati anche per i canary delle API a partire dalla versione di runtime `syn-nodejs-puppeteeer-2.2`.
- File HAR ordinabili: ora puoi ordinare i file HAR in base al codice di stato, alle dimensioni della richiesta e alla durata.
- Timestamp delle metriche: le CloudWatch metriche vengono ora riportate in base all'ora di invocazione Lambda anziché all'ora di fine dell'esecuzione Canary.

Correzioni di bug in syn-nodejs-2.0:

- Risolto il problema di errori di caricamento degli artefatti del canary che non venivano segnalati. Tali errori sono ora emersi come errori di esecuzione.
- Risolto il problema di richieste reindirizzate (3xx) che venivano registrate in modo errato come errori.
- Risolto il problema della numerazione degli screenshot a partire da 0. Ora dovrebbero iniziare con 1.
- Risolto il problema di screenshot confusi per caratteri cinesi e giapponesi.

Può verificarsi un piccolo aumento della durata del canary quando utilizzi syn-nodejs-2.0 o versioni successive, rispetto alle versioni di runtime precedenti del canary.

syn-nodejs-2.0-beta

> **Important**
>
> Questa versione di runtime è stata resa obsoleta l'8 febbraio 2021. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Runtime Lambda Node.js 10.x
- Puppeteer-core versione 3.3.0
- Chromium versione 83.0.4103.0

Nuove funzionalità in syn-nodejs-2.0-beta:

- Dipendenze aggiornate: questa versione di runtime utilizza Puppeteer-core versione 3.3.0 e Chromium 83.0.4103.0
- Synthetics Reporting: per ogni corsa di Canary CloudWatch , Synthetics crea un `SyntheticsReport-PASSED.json` rapporto `SyntheticsReport-FAILED.json` denominato o che registra dati come ora di inizio, ora di fine, stato e guasti. Registra anche lo stato PASSED/FAILED di ogni fase dello script canary ed errori e screenshot acquisiti per ogni fase.
- Report sul controllo del collegamento interrotto: la nuova versione del controllo del collegamento interrotto incluso in questo runtime crea un report che include i collegamenti controllati, il codice di stato, il motivo dell'errore (se presente) e gli screenshot della pagina di origine e di destinazione.
- Nuove CloudWatch metriche: Synthetics pubblica metriche `2xx` denominate`4xx`,, `5xx` e nel namespace. `RequestFailed` `CloudWatchSynthetics` Questi parametri mostrano il numero di 200, 400, 500 e errori di richiesta nelle esecuzioni di canary. Questi parametri vengono segnalati solo per i canary dell'interfaccia utente e non vengono segnalati per i canary delle API.
- File HAR ordinabili: ora puoi ordinare i file HAR in base al codice di stato, alle dimensioni della richiesta e alla durata.
- Timestamp delle metriche: le CloudWatch metriche vengono ora riportate in base all'ora di invocazione Lambda anziché all'ora di fine dell'esecuzione Canary.

Correzioni di bug in syn-nodejs-2.0-beta:

- Risolto il problema di errori di caricamento degli artefatti del canary che non venivano segnalati. Tali errori sono ora emersi come errori di esecuzione.
- Risolto il problema di richieste reindirizzate (3xx) che venivano registrate in modo errato come errori.
- Risolto il problema della numerazione degli screenshot a partire da 0. Ora dovrebbero iniziare con 1.
- Risolto il problema di screenshot confusi per caratteri cinesi e giapponesi.

syn-1.0

> **Important**
>
> Questa versione di runtime è pianificata per essere resa obsoleta il 28 maggio 2021. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

La prima versione di runtime di Synthetics è `syn-1.0`.

Principali dipendenze:

- Runtime Lambda Node.js 10.x
- Puppeteer-core versione 1.14.0
- La versione Chromium che corrisponde a Puppeteer-core 1.14.0

Versioni di runtime che utilizzano Python e Selenium Webdriver

Le seguenti sezioni contengono informazioni sulle versioni di runtime di CloudWatch Synthetics per Python e Selenium Webdriver. Selenium è uno strumento open source di automazione del browser. Per ulteriori informazioni su Selenium, consulta www.selenium.dev/

La convenzione di denominazione per queste versioni di runtime è `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*.

> **Important**
>
> È previsto che le seguenti CloudWatch versioni di runtime di Synthetics diventino obsolete l'8 marzo 2024.
>
> - `syn-python-selenium-2.0`
> - `syn-python-selenium-1.3`
> - `syn-python-selenium-1.2`
> - `syn-python-selenium-1.1`
> - `syn-python-selenium-1.0`

Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

syn-python-selenium-3,0

La versione 3.0 è l'ultimo runtime CloudWatch Synthetics per Python e Selenium.

Principali dipendenze:

- Python 3.8
- Selenio 4.15.1
- Versione Chromium 121.0.6167.139

Nuove funzionalità in -3.0: syn-python-selenium

- Versioni aggiornate delle librerie in bundle in Chromium: la dipendenza Chromium viene aggiornata a una nuova versione.

syn-python-selenium-2.1

Principali dipendenze:

- Python 3.8
- Selenio 4.15.1
- Chromium versione 111.0.5563.146

Nuove funzionalità in -2.1: syn-python-selenium

- Versioni aggiornate delle librerie incluse in Chromium: le dipendenze Chromium e Selenium vengono aggiornate alle nuove versioni.

syn-python-selenium-2,0

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Python 3.8
- Selenium 4.10.0
- Chromium versione 111.0.5563.146

Nuove funzionalità in -2.0: syn-python-selenium

- Dipendenze aggiornate: le dipendenze Chromium e Selenium sono aggiornate alle nuove versioni.

Correzioni di bug in -2.0 syn-python-selenium:

- Timestamp aggiunto: è stato aggiunto un timestamp ai log del canary.
- Riutilizzo della sessione: è stato corretto un bug che impediva ai canary di riutilizzare la sessione della precedente esecuzione canary.

syn-python-selenium-1,3

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Python 3.8
- Selenium 3.141.0
- Chromium versione 92.0.4512.0

Nuove funzionalità in -1.3: syn-python-selenium

- Timestamp più precisi: l'ora di inizio e l'ora di fine delle corse di canary sono ora precisi al millisecondo.

syn-python-selenium-1,2

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Python 3.8
- Selenium 3.141.0
- Chromium versione 92.0.4512.0

- Dependencies aggiornate— Le uniche nuove funzionalità di questo runtime sono le dipendenze aggiornate.

syn-python-selenium-1.1

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Python 3.8
- Selenium 3.141.0
- Chromium versione 83.0.4103.0

Funzionalità:

- Funzione di gestore personalizzato: ora puoi usare una funzione di gestore personalizzato per gli script del tuo canary. I tempi di esecuzione precedenti hanno richiesto che il punto di ingresso dello script includesse `.handler`.

  Puoi inoltre inserire gli script del canary in qualsiasi cartella e passare il nome della cartella nel gestore. Ad esempio, è possibile utilizzare `MyFolder/MyScriptFile.functionname` come punto di ingresso.
- Opzioni di configurazione per l'aggiunta di parametri e configurazioni di errori di passaggio: queste opzioni erano già disponibili nei tempi di esecuzione per i canary Node.js. Per ulteriori informazioni, consulta SyntheticsConfiguration classe.
- Argomenti personalizzati in Chrome: ora puoi aprire un browser in modalità di navigazione in incognito o passare alla configurazione del server proxy. Per ulteriori informazioni, consulta Chrome().
- Bucket di artefatti tra regioni: un canary può archiviare i propri artefatti in un bucket Amazon S3 in un'altra regione.
- Correzioni di bug, inclusa una correzione per il problema **`index.py`**: con i tempi di esecuzione precedenti, un file canary denominato `index.py` ha causato eccezioni perché era in conflitto con il nome del file della libreria. Questo problema è stato risolto.

syn-python-selenium-1,0

> **Important**
>
> Questa versione di runtime dovrebbe diventare obsoleta l'8 marzo 2024. Per ulteriori informazioni, consulta CloudWatch Politica di supporto per il runtime di Synthetics.

Principali dipendenze:

- Python 3.8
- Selenium 3.141.0
- Chromium versione 83.0.4103.0

Funzionalità:

- Supporto per Selenium: puoi scrivere script canary utilizzando il framework di test Selenium. Puoi trasferire i tuoi script Selenium da altrove in Synthetics con modifiche minime e CloudWatch funzioneranno con i servizi. AWS

Scrivere uno script canary

Le sezioni seguenti spiegano come scrivere uno script canarino e come integrare un canary con altri AWS servizi e con dipendenze e librerie esterne.

Argomenti

- Scrivere uno script canary Node.js
- Scrivere uno script canary Python
- Modifica di uno script Selenium esistente da utilizzare come canary di Synthetics
- Modifica di uno script Puppeteer Synthetics esistente per autenticare certificati non standard

Scrivere uno script canary Node.js

Argomenti

- Creare un canarino CloudWatch Synthetics da zero
- Impacchettizzazione dei file canary Node.js
- Modifica di uno script Puppeteer esistente da utilizzare come canary di Synthetics
- Variabili di ambiente
- Integrazione del tuo canarino con altri servizi AWS
- Forzare il canary a utilizzare un indirizzo IP statico

Creare un canarino CloudWatch Synthetics da zero

Ecco un esempio minimo di script Synthetics canary. Questo script passa come esecuzione riuscita e restituisce una stringa. Per vedere come appare un canary fallito, passare `let fail = false;` a `let fail = true;`.

È necessario definire una funzione di punto di ingresso per lo script canary. Per vedere come i file vengono caricati nella posizione Amazon S3 specificata come `ArtifactS3Location` del canary, creare questi file nella cartella /tmp. Dopo l'esecuzione dello script, lo stato di pass/fail e le metriche della durata vengono pubblicate su e i file in /tmp vengono CloudWatch caricati su S3.

```
const basicCustomEntryPoint = async function () {

    // Insert your code here

    // Perform multi-step pass/fail check

    // Log decisions made and results to /tmp

    // Be sure to wait for all your code paths to complete
    // before returning control back to Synthetics.
    // In that way, your canary will not finish and report success
    // before your code has finished executing

    // Throw to fail, return to succeed
    let fail = false;
    if (fail) {
        throw "Failed basicCanary check.";
    }

    return "Successfully completed basicCanary checks.";
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

Successivamente, espanderemo lo script per utilizzare la registrazione di Synthetics ed effettuare una chiamata utilizzando l'SDK. AWS A scopo dimostrativo, questo script creerà un client Amazon DynamoDB ed effettuerà una chiamata all'API ListTables DynamoDB. Registra la risposta alla richiesta e i registri passano o falliscono in base all'esito positivo della richiesta.

```
const log = require('SyntheticsLogger');
const AWS = require('aws-sdk');
// Require any dependencies that your script needs
// Bundle additional files and dependencies into a .zip file with folder structure
// nodejs/node_modules/additional files and folders

const basicCustomEntryPoint = async function () {

    log.info("Starting DynamoDB:listTables canary.");

    let dynamodb = new AWS.DynamoDB();
```

```
    var params = {};
    let request = await dynamodb.listTables(params);
    try {
        let response = await request.promise();
        log.info("listTables response: " + JSON.stringify(response));
    } catch (err) {
        log.error("listTables error: " + JSON.stringify(err), err.stack);
        throw err;
    }

    return "Successfully completed DynamoDB:listTables canary.";
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

Impacchettizzazione dei file canary Node.js

Se carichi gli script del canary utilizzando una posizione di Amazon S3, il file zip deve includere lo script in questa struttura di cartelle: `nodejs/node_modules/`*`myCanaryFilename.js file`*.

Se si dispone di più di un singolo file `.js` o si dispone di una dipendenza da cui dipende lo script, è possibile raggrupparli tutti in un unico file ZIP contenente la struttura delle cartelle `nodejs/node_modules/`*`myCanaryFilename.js file and other folders and files`*. Se utilizzi un tempo `syn-nodejs-puppeteer-3.4` o successivo, puoi inserire i file del canary in un'altra cartella e creare una struttura della cartella simile a quella seguente: `nodejs/node_modules/`*`myFolder`*`/`*`myCanaryFilename.js file and other folders and files`*.

Nome del gestore

Assicurati di impostare il punto di ingresso dello script del tuo canary (gestore) in modo che `myCanaryFilename.functionName` corrisponda al nome del file del punto di ingresso dello script. Se utilizzi un runtime precedente a `syn-nodejs-puppeteer-3.4`, `functionName` deve corrispondere a `handler`. Se utilizzi un tempo `syn-nodejs-puppeteer-3.4` o successivo, puoi scegliere qualsiasi nome di funzione come gestore. Se utilizzi un tempo `syn-nodejs-puppeteer-3.4` o successivo, puoi anche memorizzare il canary in una cartella separata, ad esempio `nodejs/node_modules/myFolder/my_canary_filename`. Se lo archivi in una cartella separata, definisci il percorso nel punto di ingresso dello script, ad esempio `myFolder/my_canary_filename.functionName`.

Modifica di uno script Puppeteer esistente da utilizzare come canary di Synthetics

Questa sezione spiega come prendere script Puppeteer e modificarli per essere eseguiti come script canary Synthetics. Per ulteriori informazioni su Puppeteer, consulta Puppeteer API v1.14.0.

Inizieremo con questo esempio di script Puppeteer:

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  await page.screenshot({path: 'example.png'});

  await browser.close();
})();
```

Le fasi di conversione sono le seguenti:

- Creare ed esportare una funzione `handler`. Il gestore è la funzione del punto di ingresso per lo script. Se utilizzi un runtime precedente a `syn-nodejs-puppeteer-3.4`, la funzione del gestore deve essere denominata `handler`. Se utilizzi un tempo `syn-nodejs-puppeteer-3.4` o successivo, la funzione può avere un nome qualsiasi, purché corrisponda a quello utilizzato nello script. Inoltre, se utilizzi un tempo `syn-nodejs-puppeteer-3.4` o successivo, puoi archiviare gli script in qualsiasi cartella e definire quest'ultima nel nome del gestore.

  ```
  const basicPuppeteerExample = async function () {};

  exports.handler = async () => {
      return await basicPuppeteerExample();
  };
  ```

- Usa la dipendenza `Synthetics`.

  ```
  var synthetics = require('Synthetics');
  ```

- Utilizzare la funzione `Synthetics.getPage` per ottenere un oggetto `Page` Puppeteer.

  ```
  const page = await synthetics.getPage();
  ```

L'oggetto pagina restituito dalla funzione Synthetics.getPage ha gli eventi page.on `request`, `response` e `requestfailed` strumentati per la registrazione. Synthetics imposta anche la generazione di file HAR per le richieste e le risposte nella pagina e aggiunge l'ARN canary alle intestazioni user-agent delle richieste in uscita nella pagina.

Lo script è ora pronto per essere eseguito come Canary di Synthetics. Ecco lo script aggiornato:

```
var synthetics = require('Synthetics');  // Synthetics dependency

const basicPuppeteerExample = async function () {
    const page = await synthetics.getPage(); // Get instrumented page from Synthetics
    await page.goto('https://example.com');
    await page.screenshot({path: '/tmp/example.png'}); // Write screenshot to /tmp
 folder
};

exports.handler = async () => {  // Exported handler function
    return await basicPuppeteerExample();
};
```

Variabili di ambiente

Quando crei canary puoi utilizzare le variabili di ambiente. Ciò consente di scrivere un singolo script canary e quindi utilizzare tale script con valori diversi per creare rapidamente più canary che hanno un'attività simile.

Ad esempio, supponiamo che l'organizzazione disponga di endpoint quali `prod`, `dev` e `pre-release` per le diverse fasi dello sviluppo del software ed è necessario creare canary per testare ciascuno di questi endpoint. È possibile scrivere un singolo script canary che testi il software e quindi specificare valori diversi per la variabile di ambiente dell'endpoint quando crei ciascuno dei tre canary. Quindi, quando crei un canary, si specificano lo script e i valori da utilizzare per le variabili di ambiente.

I nomi delle variabili di ambiente possono contenere lettere, numeri e il carattere di sottolineatura. Devono iniziare con una lettera e avere almeno due caratteri. La dimensione totale delle variabili di ambiente non può superare i 4 KB. Non è possibile specificare alcuna variabile di ambiente riservato Lambda come nomi per le variabili di ambiente. Per ulteriori informazioni sulle variabili di ambiente riservate, consulta Variabili di ambiente di runtime.

> **Important**
>
> Le chiavi e i valori delle variabili d'ambiente non sono crittografati. Non archiviare informazioni sensibili al loro interno.

Lo script di esempio seguente utilizza due variabili di ambiente. Questo script è per un canary che controlla se una pagina Web è disponibile. Utilizza variabili di ambiente per parametrizzare sia l'URL che controlla sia il livello di registro Synthetics CloudWatch che utilizza.

La seguente funzione imposta `LogLevel` al valore della variabile di ambiente `LOG_LEVEL`.

```
synthetics.setLogLevel(process.env.LOG_LEVEL);
```

Questa funzione imposta URL al valore della variabile di ambiente URL.

```
const URL = process.env.URL;
```

Questo è lo script completo. Quando crei un canary utilizzando questo script, si specificano i valori per le variabili di ambiente `LOG_LEVEL` e URL.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadEnvironmentVariable = async function () {

    // Setting the log level (0-3)
    synthetics.setLogLevel(process.env.LOG_LEVEL);
    // INSERT URL here
    const URL = process.env.URL;

    let page = await synthetics.getPage();
    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
    }
    //Wait for page to render.
    //Increase or decrease wait time based on endpoint being monitored.
```

```
    await page.waitFor(15000);
    await synthetics.takeScreenshot('loaded', 'loaded');
    let pageTitle = await page.title();
    log.info('Page title: ' + pageTitle);
    log.debug('Environment variable:' + process.env.URL);

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};

exports.handler = async () => {
    return await pageLoadEnvironmentVariable();
};
```

Passare le variabili di ambiente allo script

Per passare le variabili di ambiente allo script quando crei un canary nella console, specifica le chiavi e i valori delle variabili di ambiente nella finestra Environment variables (Variabili di ambiente) della console. Per ulteriori informazioni, consulta Creazione di un Canary.

Per passare le variabili di ambiente tramite l'API oppure AWS CLI, usa il `EnvironmentVariables` parametro nella sezione. `RunConfig` Di seguito è riportato un AWS CLI comando di esempio che crea un canarino che utilizza due variabili di ambiente con le chiavi `Environment` e`Region`.

```
aws synthetics create-canary --cli-input-json '{
   "Name":"nameofCanary",
   "ExecutionRoleArn":"roleArn",
   "ArtifactS3Location":"s3://cw-syn-results-123456789012-us-west-2",
   "Schedule":{
      "Expression":"rate(0 minute)",
      "DurationInSeconds":604800
   },
   "Code":{
      "S3Bucket": "canarycreation",
      "S3Key": "cwsyn-mycanaryheartbeat-12345678-d1bd-1234-
abcd-123456789012-12345678-6a1f-47c3-b291-123456789012.zip",
      "Handler":"pageLoadBlueprint.handler"
   },
   "RunConfig": {
      "TimeoutInSeconds":60,
      "EnvironmentVariables": {
```

```
         "Environment":"Production",
         "Region": "us-west-1"
      }
   },
   "SuccessRetentionPeriodInDays":13,
   "FailureRetentionPeriodInDays":13,
   "RuntimeVersion":"syn-nodejs-2.0"
}'
```

Integrazione del tuo canarino con altri servizi AWS

Tutti i canarini possono utilizzare la AWS libreria SDK. Puoi usare questa libreria quando scrivi il tuo canarino per integrare il canarino con altri servizi. AWS

Per fare ciò, è necessario aggiungere il seguente codice al tuo canary. Per questi esempi, AWS Secrets Manager viene utilizzato come servizio con cui il canarino si sta integrando.

- Importa l'SDK. AWS

```
const AWS = require('aws-sdk');
```

- Crea un client per il AWS servizio con cui ti stai integrando.

```
const secretsManager = new AWS.SecretsManager();
```

- Utilizzare il client per effettuare chiamate API a tale servizio.

```
var params = {
  SecretId: secretName
};
return await secretsManager.getSecretValue(params).promise();
```

Il seguente frammento di codice di script canary mostra un esempio di integrazione con Secrets Manager in modo più dettagliato.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const AWS = require('aws-sdk');
const secretsManager = new AWS.SecretsManager();
```

```
const getSecrets = async (secretName) => {
    var params = {
        SecretId: secretName
    };
    return await secretsManager.getSecretValue(params).promise();
}

const secretsExample = async function () {
    let URL = "<URL>";
    let page = await synthetics.getPage();

    log.info(`Navigating to URL: ${URL}`);
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});

    // Fetch secrets
    let secrets = await getSecrets("secretname")

    /**
    * Use secrets to login.
    *
    * Assuming secrets are stored in a JSON format like:
    * {
    *   "username": "<USERNAME>",
    *   "password": "<PASSWORD>"
    * }
    **/
    let secretsObj = JSON.parse(secrets.SecretString);
    await synthetics.executeStep('login', async function () {
        await page.type(">USERNAME-INPUT-SELECTOR<", secretsObj.username);
        await page.type(">PASSWORD-INPUT-SELECTOR<", secretsObj.password);

        await Promise.all([
          page.waitForNavigation({ timeout: 30000 }),
          await page.click(">SUBMIT-BUTTON-SELECTOR<")
        ]);
    });

    // Verify login was successful
    await synthetics.executeStep('verify', async function () {
        await page.waitForXPath(">SELECTOR<", { timeout: 30000 });
    });
};
```

```
exports.handler = async () => {
    return await secretsExample();
};
```

Forzare il canary a utilizzare un indirizzo IP statico

Puoi configurare un canary in modo che utilizzi un indirizzo IP statico.

Per forzare un canary a utilizzare un indirizzo IP statico

1. Crea un nuovo VPC. Per ulteriori informazioni, vedi Utilizzo del DNS con VPC.
2. Crea un nuovo gateway Internet. Per ulteriori informazioni, consulta la pagina relativa all'Aggiunta di un gateway Internet al VPC.
3. Crea una sottorete pubblica all'interno del tuo nuovo VPC.
4. Aggiungi una nuova tabella di routing al VPC.
5. Aggiungi un routing nella nuova tabella di routing, che va da `0.0.0.0/0` al gateway Internet.
6. Associa la nuova tabella di routing alla sottorete pubblica.
7. Crea un indirizzo IP elastico. Per ulteriori informazioni, consulta Indirizzi IP elastici.
8. Crea un nuovo gateway NAT e assegnalo alla sottorete pubblica e all'indirizzo IP elastico.
9. Per creare una sottorete privata all'interno del VPC.
10. Aggiungere un routing alla tabella di tabella di routing predefinita del VPC, che va da `0.0.0.0/0` al gateway NAT
11. Crea il tuo canary.

Scrivere uno script canary Python

Questo script passa come esecuzione riuscita e restituisce una stringa. Per vedere come appare un canary fallito, passare “fail = False” a “fail = True”

```
def basic_custom_script():
    # Insert your code here
    # Perform multi-step pass/fail check
    # Log decisions made and results to /tmp
    # Be sure to wait for all your code paths to complete
    # before returning control back to Synthetics.
    # In that way, your canary will not finish and report success
    # before your code has finished executing
    fail = False
```

```
    if fail:
        raise Exception("Failed basicCanary check.")
    return "Successfully completed basicCanary checks."
def handler(event, context):
    return basic_custom_script()
```

Impacchettizzazione dei file canary in Python

Se disponi di più di un file .py o lo script ha una dipendenza, puoi raggrupparli tutti in un unico file ZIP. Se utilizzi il runtime `syn-python-selenium-1.1`, il file ZIP deve contenere il file del canary principale con estensione py all'interno di una cartella `python`, ad esempio `python/my_canary_filename.py`. Se utilizzi un tempo `syn-python-selenium-1.1` o successivo, puoi utilizzare una cartella diversa, ad·esempio `python/myFolder/my_canary_filename.py`.

Questo file ZIP deve contenere tutte le cartelle e i file necessari, ma gli altri file non devono essere presenti nella cartella `python`.

Assicurati di impostare il punto di ingresso dello script del tuo canary in modo che `my_canary_filename.functionName` corrisponda al nome del file e al nome della funzione del punto di ingresso dello script. Se utilizzi il runtime `syn-python-selenium-1.0`, `functionName` deve corrispondere·a `handler`. Se utilizzi un tempo `syn-python-selenium-1.1` o successivo, questa limitazione del nome del gestore non si applica e puoi anche archiviare il canary in una cartella separata, ad esempio `python/myFolder/my_canary_filename.py`. Se lo archivi in una cartella separata, definisci il percorso nel punto di ingresso dello script, ad esempio `myFolder/my_canary_filename.functionName`.

Modifica di uno script Selenium esistente da utilizzare come canary di Synthetics

Puoi modificare rapidamente uno script esistente per Python e Selenium da utilizzare come canary. Per ulteriori informazioni su Selenium, consulta www.selenium.dev/.

Per questo esempio, inizieremo con il seguente script Selenium:

```
from selenium import webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

basic_selenium_script()
```

Le fasi di conversione sono le seguenti.

Per convertire uno script selenio da usare come canary

1. Modifica l'istruzione `import` per utilizzare Selenium dal modulo `aws_synthetics`:

   ```
   from aws_synthetics.selenium import synthetics_webdriver as webdriver
   ```

   Il modulo Selenium di `aws_synthetics` assicura che il canarino possa emettere metriche e registri, generare un file HAR e funzionare con altre funzionalità di Synthetics. CloudWatch

2. Crea una funzione del gestore e chiama il tuo metodo Selenium. Il gestore è la funzione del punto di ingresso per lo script.

   Se utilizzi un tempo `syn-python-selenium-1.0`, la funzione del gestore deve essere denominata `handler`. Se utilizzi un tempo `syn-python-selenium-1.1` o successivo, la funzione può avere un nome qualsiasi, purché corrisponda a quello utilizzato nello script. Inoltre, se utilizzi un tempo `syn-python-selenium-1.1` o successivo, puoi archiviare gli script in qualsiasi cartella e definire quest'ultima nel nome del gestore.

   ```
   def handler(event, context):
       basic_selenium_script()
   ```

Lo script è ora aggiornato per essere un canarino CloudWatch Synthetics. Ecco lo script aggiornato:

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

def handler(event, context):
    basic_selenium_script()
```

Modifica di uno script Puppeteer Synthetics esistente per autenticare certificati non standard

Un caso d'uso importante per Synthetics canaries è il monitoraggio dei propri endpoint. Se volete monitorare un endpoint che non è pronto per il traffico esterno, questo monitoraggio a volte può

significare che non avete un certificato adeguato firmato da un'autorità di certificazione terza affidabile.

Le due possibili soluzioni a questo scenario sono le seguenti:

- Per autenticare un certificato client, consulta Come convalidare l'autenticazione utilizzando Amazon Synthetics CloudWatch — Parte 2.
- Per autenticare un certificato autofirmato, consulta Come convalidare l'autenticazione con certificati autofirmati in Amazon Synthetics CloudWatch

Non sei limitato a queste due opzioni quando usi i canarini CloudWatch Synthetics. Puoi estendere queste funzionalità e aggiungere la tua logica aziendale estendendo il codice canarino.

> **Note**
>
> I canary Synthetics in esecuzione su runtime Python hanno per natura il flag abilitato, quindi non dovrebbero `--ignore-certificate-errors` avere problemi a raggiungere siti con configurazioni di certificati non standard.

Funzioni di libreria disponibili per gli script canary

CloudWatch Synthetics include diverse classi e funzioni integrate che è possibile chiamare durante la scrittura di script Node.js da utilizzare come canarini.

Alcune si applicano sia ai canary UI che alle API. Altri si applicano solo ai canary dell'interfaccia utente. Un canary dell'interfaccia utente è un canary che utilizza la funzione `getPage()` e utilizza Puppeteer come driver Web per navigare e interagire con le pagine Web.

> **Note**
>
> Ogni volta che aggiorni un canary per utilizzare una nuova versione del runtime Synthetics, tutte le funzioni della libreria Synthetics utilizzate dal canary vengono aggiornate automaticamente alla stessa versione di NodeJS supportata dal runtime Synthetics.

Argomenti

- Funzioni di libreria disponibili per gli script canary Node.js

- Funzioni di libreria disponibili per gli script canary Python che usano Selenium

Funzioni di libreria disponibili per gli script canary Node.js

In questa sezione sono elencate le funzioni di libreria disponibili per gli script canary Node.js.

Argomenti

- Classi e funzioni di libreria di Node.js che si applicano a tutti i canary
- Classi e funzioni della libreria di Node.js che si applicano solo ai canary dell'interfaccia utente
- Classi e funzioni della libreria di Node.js che si applicano solo ai canary dell'API

Classi e funzioni di libreria di Node.js che si applicano a tutti i canary

Le seguenti funzioni della libreria CloudWatch Synthetics per Node.js sono utili per tutti i canarini.

Argomenti

- Classe Synthetics
- SyntheticsConfiguration classe
- Synthetics logger
- SyntheticsLogHelper classe

Classe Synthetics

Le seguenti funzioni per tutti i canary sono nella classe Synthetics.

addExecutionError(ErrorMessage, ad esempio);

`errorMessage` descrive l'errore e `ex` è l'eccezione che si verifica

Puoi utilizzare `addExecutionError` per impostare gli errori di esecuzione per il tuo canary. Fa fallire il canary senza interrompere l'esecuzione dello script. Inoltre, non influisce sui tuoi parametri `successPercent`.

È necessario tenere traccia degli errori come errori di esecuzione solo se non sono importanti per indicare il successo o il fallimento dello script canary.

Di seguito è illustrato un esempio dell'uso dell'elemento `addExecutionError`. Stai monitorando la disponibilità del tuo endpoint e acquisendo screenshot dopo che la pagina è stata caricata.

Poiché l'errore di acquisizione di uno screenshot non determina la disponibilità dell'endpoint, puoi rilevare eventuali errori riscontrati durante l'acquisizione di screenshot e aggiungerli come errori di esecuzione. I parametri di disponibilità indicheranno comunque che l'endpoint è attivo e funzionante, ma lo stato del canary verrà contrassegnato come non riuscito. Il seguente blocco di codice di esempio rileva tale errore e lo aggiunge come errore di esecuzione.

```
try {
        await synthetics.takeScreenshot(stepName, "loaded");
    } catch(ex) {
        synthetics.addExecutionError('Unable to take screenshot ', ex);
    }
```

getCanaryName();

Restituisce il nome del canary.

getCanaryArn();

Restituisce l'ARN del canary.

getCanaryUserAgentString();

Restituisce l'agente utente personalizzato del canary.

getRuntimeVersion();

Questa funzione è disponibile nella versione di runtime `syn-nodejs-puppeteer-3.0` e versione successiva. Restituisce la versione di runtime di Synthetics del canary. Ad esempio, il valore restituito potrebbe essere `syn-nodejs-puppeteer-3.0`.

getLogLevel();

Recupera il livello di log corrente per la libreria Synthetics. I valori possibili sono i seguenti:

- `0`: Debug
- `1`: Info
- `2`: Warn
- `3`: Error

Esempio:

```
let logLevel = synthetics.getLogLevel();
```

setLogLevel();

Imposta il livello di log per la libreria Synthetics. I valori possibili sono i seguenti:

- `0`: Debug
- `1`: Info
- `2`: Warn
- `3`: Error

Esempio:

```
synthetics.setLogLevel(0);
```

SyntheticsConfiguration classe

Questa classe è disponibile solo nella versione di runtime `syn-nodejs-2.1` o versioni successive.

È possibile utilizzare la SyntheticsConfiguration classe per configurare il comportamento delle funzioni della libreria Synthetics. Ad esempio, puoi utilizzare questa classe per configurare la funzione `executeStep()` per non acquisire screenshot.

È possibile impostare configurazioni CloudWatch Synthetics a livello globale, che vengono applicate a tutti i passaggi di Canaries. Puoi anche sovrascrivere queste configurazioni a livello di fase passando coppie chiave/valore di configurazione.

Puoi passare le opzioni a livello di fase. Per alcuni esempi, consulta async executeStep (StepName,, [StepConfig]); functionToExecute e executeHttpStep(StepName, RequestOptions, [callback], [StepConfig])

Definizioni della funzione:

setConfig(options)

`options` è un oggetto, che è un insieme di opzioni configurabili per il tuo canary. Le seguenti sezioni spiegano i campi possibili in *`options`*.

setConfig(options) per tutti i canary

Per i canary che utilizzano `syn-nodejs-puppeteer-3.2` o versione successiva, (options) per setConfig può includere i seguenti parametri:

- `includeRequestHeaders` (booleano): indica se includere le intestazioni della richiesta nel report. Il valore predefinito è `false`.
- `includeResponseHeaders` (booleano): indica se includere le intestazioni della risposta nel report. Il valore predefinito è `false`.
- `restrictedHeaders` (array): un elenco di valori di intestazione da ignorare, se le intestazioni sono incluse. Questo vale sia per le intestazioni della richiesta che della risposta. Ad esempio, puoi nascondere le tue credenziali passando includeRequestHeadersas e RestrictedHeaders come. **`true`** `['Authorization']`
- `includeRequestBody` (booleano): indica se includere il corpo della richiesta nel report. Il valore predefinito è `false`.
- `includeResponseBody` (booleano): indica se includere il corpo della risposta nel report. Il valore predefinito è `false`.

setConfig (opzioni) relative alle metriche CloudWatch

Per i canary che utilizzano `syn-nodejs-puppeteer-3.1` o versione successiva, (options) per setConfig può includere i seguenti parametri booleani che determinano quali parametri vengono pubblicati dal canary. Il valore predefinito per ciascuna di queste opzioni è `true`. Le opzioni che iniziano con `aggregated` determinano se il parametro viene emesso senza destinazione `CanaryName`. Puoi usare questi parametri per visualizzare i risultati aggregati per tutti i canary. Le altre opzioni determinano se il parametro viene emesso con la dimensione `CanaryName`. Puoi usare questi parametri per visualizzare i risultati per ogni singolo canary.

Per un elenco delle CloudWatch metriche emesse dai canarini, vedi. CloudWatch metriche pubblicate da canaries

- `failedCanaryMetric` (booleano): indica se emettere il parametro `Failed` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `failedRequestsMetric` (booleano): indica se emettere il parametro `Failed requests` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `_2xxMetric` (booleano): indica se emettere il parametro `2xx` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.

- `_4xxMetric` (booleano): indica se emettere il parametro `4xx` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `_5xxMetric` (booleano): indica se emettere il parametro `5xx` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `stepDurationMetric` (booleano): indica se emettere il parametro `Step duration` (con le dimensioni `CanaryName` `StepName`) per questo canary. Il valore predefinito è `true`.
- `stepSuccessMetric` (booleano): indica se emettere il parametro `Step success` (con le dimensioni `CanaryName` `StepName`) per questo canary. Il valore predefinito è `true`.
- `aggregatedFailedCanaryMetric` (booleano): indica se emettere il parametro `Failed` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `aggregatedFailedRequestsMetric` (booleano): indica se emettere il parametro `Failed Requests` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `aggregated2xxMetric` (booleano): indica se emettere il parametro `2xx` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `aggregated4xxMetric` (booleano): indica se emettere il parametro `4xx` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `aggregated5xxMetric` (booleano): indica se emettere il parametro `5xx` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `visualMonitoringSuccessPercentMetric` (booleano): indica se emettere il parametro `visualMonitoringSuccessPercent` per questo canary. Il valore predefinito è `true`.
- `visualMonitoringTotalComparisonsMetric` (booleano): indica se emettere il parametro `visualMonitoringTotalComparisons` per questo canary. Il valore predefinito è `false`.
- `stepsReport` (booleano): indica se segnalare un riepilogo dell'esecuzione delle fasi. Il valore predefinito è `true`.
- `includeUrlPassword` (booleano): indica se includere una password visualizzata nell'URL. Per impostazione predefinita, le password visualizzate negli URL vengono cancellate da log e report, per evitare la divulgazione di dati sensibili. Il valore predefinito è `false`.
- `restrictedUrlParameters` (array): un elenco di parametri del percorso dell'URL o della query da oscurare. Si applica agli URL visualizzati nei log, nei report e negli errori. Il parametro prevede la distinzione tra lettere maiuscole e minuscole. Puoi passare un asterisco (*) come valore per oscurare tutti i valori dei parametri del percorso dell'URL e delle query. L'impostazione predefinita è una matrice vuota.

- `logRequest` (booleano): indica se registrare ogni richiesta nei log del canary. Per i canary dell'interfaccia utente, registra ogni richiesta inviata dal browser. Il valore predefinito è `true`.
- `logResponse` (booleano): indica se registrare ogni risposta nei log del canary. Per i canary dell'interfaccia utente, registra ogni risposta ricevuta dal browser. Il valore predefinito è `true`.
- `logRequestBody` (booleano): indica se registrare i corpi delle richieste insieme alle richieste nei log dei canary. Questa configurazione si applica solo se `logRequest` è `true`. Il valore predefinito è `false`.
- `logResponseBody` (booleano): indica se registrare i corpi delle risposte insieme alle risposte nei log dei canary. Questa configurazione si applica solo se `logResponse` è `true`. Il valore predefinito è `false`.
- `logRequestHeaders` (booleano): indica se registrare le intestazioni delle richieste insieme alle richieste nei log dei canary. Questa configurazione si applica solo se `logRequest` è `true`. Il valore predefinito è `false`.

  Nota che `includeRequestHeaders` abilita le intestazioni negli artefatti.
- `logResponseHeaders` (booleano): indica se registrare le intestazioni delle risposte insieme alle risposte nei log dei canary. Questa configurazione si applica solo se `logResponse` è `true`. Il valore predefinito è `false`.

  Nota che `includeResponseHeaders` abilita le intestazioni negli artefatti.

> **Note**
>
> I parametri `Duration` e `SuccessPercent` vengono sempre emessi per ogni canary, sia con che senza il parametro `CanaryName`.

Metodi per abilitare o disabilitare i parametri

disableAggregatedRequestMetriche ()

Disabilita il canary dall'emettere tutti i parametri delle richieste emessi senza dimensione `CanaryName`.

disableRequestMetrics()

Disabilita tutti i parametri delle richieste, inclusi i parametri dei canary e i parametri aggregati in tutti i canary.

disableStepMetrics()

Disabilita tutti i parametri dei passaggi, inclusi sia quelli di riuscita che di durata.

enableAggregatedRequestMetriche ()

Consente al canary di emettere tutti i parametri della richiesta emessi senza la dimensione `CanaryName`.

enableRequestMetrics()

Abilita tutti i parametri delle richieste, inclusi i parametri dei canary e i parametri aggregati in tutti i canary.

enableStepMetrics()

Abilita tutti i parametri dei passaggi, inclusi sia quelli di riuscita che di durata.

get2xxMetric()

Indica se il canary emette un parametro `2xx` con dimensione `CanaryName`.

get4xxMetric()

Indica se il canary emette un parametro `4xx` con dimensione `CanaryName`.

get5xxMetric()

Indica se il canary emette un parametro `5xx` con dimensione `CanaryName`.

getAggregated2xxMetric()

Indica se il canary emette un parametro `2xx` senza dimensione.

getAggregated4xxMetric()

Indica se il canary emette un parametro `4xx` senza dimensione.

getAggregatedFailedCanaryMetric()

Indica se il canary emette un parametro `Failed` senza dimensione.

getAggregatedFailedRequestsMetric()

Indica se il canary emette un parametro `Failed requests` senza dimensione.

getAggregated5xxMetric()

Indica se il canary emette un parametro `5xx` senza dimensione.

getFailedCanaryMetrica ()

Indica se il canary emette un parametro `Failed` con dimensione `CanaryName`.

getFailedRequestsMetrico ()

Indica se il canary emette un parametro `Failed requests` con dimensione `CanaryName`.

getStepDurationMetrico ()

Indica se il canary emette un parametro `Duration` con dimensione `CanaryName` per questo canary.

getStepSuccessMetrico ()

Indica se il canary emette un parametro `StepSuccess` con dimensione `CanaryName` per questo canary.

with2xxMetric(_2xxMetric)

Accetta un argomento booleano, che specifica se emettere un parametro `2xx` con dimensione `CanaryName` per questo canary.

with4xxMetric(_4xxMetric)

Accetta un argomento booleano, che specifica se emettere un parametro `4xx` con dimensione `CanaryName` per questo canary.

with5xxMetric(_5xxMetric)

Accetta un argomento booleano, che specifica se emettere un parametro `5xx` con dimensione `CanaryName` per questo canary.

withAggregated2xxMetric(aggregated2xxMetric)

Accetta un argomento booleano, che specifica se emettere un parametro `2xx` senza alcuna dimensione per questo canary.

withAggregated4xxMetric(aggregated4xxMetric)

Accetta un argomento booleano, che specifica se emettere un parametro `4xx` senza alcuna dimensione per questo canary.

withAggregated5xxMetric(aggregated5xxMetric)

Accetta un argomento booleano, che specifica se emettere un parametro `5xx` senza alcuna dimensione per questo canary.

withAggregatedFailedCanaryMetric(aggregatedFailedCanaryMetrico)

Accetta un argomento booleano, che specifica se emettere un parametro `Failed` senza alcuna dimensione per questo canary.

withAggregatedFailedRequestsMetric(aggregatedFailedRequestsMetrico)

Accetta un argomento booleano, che specifica se emettere un parametro `Failed requests` senza alcuna dimensione per questo canary.

withFailedCanaryMetrico () failedCanaryMetric

Accetta un argomento booleano, che specifica se emettere un parametro `Failed` con dimensione `CanaryName` per questo canary.

withFailedRequestsMetrico () failedRequestsMetric

Accetta un argomento booleano, che specifica se emettere un parametro `Failed requests` con dimensione `CanaryName` per questo canary.

withStepDurationMetrico () stepDurationMetric

Accetta un argomento booleano, che specifica se emettere un parametro `Duration` con dimensione `CanaryName` per questo canary.

withStepSuccessMetrico () stepSuccessMetric

Accetta un argomento booleano, che specifica se emettere un parametro `StepSuccess` con dimensione `CanaryName` per questo canary.

Metodi per abilitare o disabilitare altre funzionalità

withHarFile()

Accetta un argomento booleano, che specifica se creare un file HAR per questo canary.

withStepsReport()

Accetta un argomento booleano, che specifica se segnalare un riepilogo dell'esecuzione delle fasi per questo canary.

withIncludeUrlPassword ()

Accetta un argomento booleano, che specifica se includere le password visualizzate negli URL nei log e nei report.

withRestrictedUrlParametri ()

Accetta una matrice di parametri del percorso dell'URL o della query da oscurare. Si applica agli URL visualizzati nei log, nei report e negli errori. Puoi passare un asterisco (*) come valore per oscurare tutti i valori dei parametri del percorso dell'URL e delle query

withLogRequest()

Accetta un argomento booleano, che specifica se registrare ogni richiesta nei log del canary.

withLogResponse()

Accetta un argomento booleano, che specifica se registrare ogni risposta nei log del canary.

withLogRequestCorpo ()

Accetta un argomento booleano, che specifica se registrare ogni corpo della richiesta nei log del canary.

withLogResponseCorpo ()

Accetta un argomento booleano, che specifica se registrare ogni corpo della risposta nei log del canary.

withLogRequestIntestazioni ()

Accetta un argomento booleano, che specifica se registrare ogni intestazione della richiesta nei log del canary.

withLogResponseIntestazioni ()

Accetta un argomento booleano, che specifica se registrare ogni intestazione della risposta nei log del canary.

getHarFile()

Restituisce se il canary crea un file HAR.

getStepsReport()

Restituisce se il canary segnala un riepilogo dell'esecuzione della fase.

getIncludeUrlPassword ()

Restituisce se il canary include password visualizzate negli URL nei log e nei report.

getRestrictedUrlParametri ()

Restituisce se il canary oscura i parametri del percorso dell'URL o della query.

getLogRequest()

Restituisce se il canary registra ogni richiesta nei log del canary.

getLogResponse()

Restituisce se il canary registra ogni risposta nei log del canary.

getLogRequestCorpo ()

Restituisce se il canary registra ogni corpo della richiesta nei log del canary.

getLogResponseCorpo ()

Restituisce se il canary registra ogni corpo della risposta nei log del canary.

getLogRequestIntestazioni ()

Restituisce se il canary registra ogni intestazione della richiesta nei log del canary.

getLogResponseIntestazioni ()

Restituisce se il canary registra ogni intestazione della risposta nei log del canary.

Funzioni per tutti i canary

- `withIncludeRequestHeaders`(includeRequestHeaders)
- `withIncludeResponseHeaders`(includeResponseHeaders)
- `withRestrictedHeaders`(restrictedHeaders)
- `withIncludeRequestBody`(includeRequestBody)
- `withIncludeResponseBody`(includeResponseBody)
- `enableReportingOptions`() — Abilita tutte le opzioni di reporting-- includeRequestHeaders, includeResponseHeadersincludeRequestBody, e includeResponseBody,.
- `disableReportingOptions`() — Disattiva tutte le opzioni di reporting-- includeRequestHeaders, includeResponseHeadersincludeRequestBody, e includeResponseBody,.

setConfig(options) per canary dell'interfaccia utente

Per i canary dell'interfaccia utente, setConfig può includere i seguenti parametri booleani:

- `continueOnStepFailure` (booleano): indica se continuare con l'esecuzione dello script canary dopo che una fase ha esito negativo (questo si riferisce alla funzione executeStep). Se le fasi hanno esito negativo, l'esecuzione del canary verrà comunque contrassegnata come fallita. Il valore predefinito è `false`.
- `harFile` (booleano): indica se creare un file HAR. Il valore predefinito è `True`.
- `screenshotOnStepStart` (booleano): indica se fare uno screenshot prima di iniziare una fase.
- `screenshotOnStepSuccess` (booleano): indica se acquisire uno screenshot dopo aver completato una fase riuscita.
- `screenshotOnStepFailure` (booleano): indica se acquisire uno screenshot dopo il fallimento di una fase.

Metodi per abilitare o disabilitare gli screenshot

disableStepScreenshots()

Disattiva tutte le opzioni relative agli screenshot (screenshotOnStepStart, screenshotOnStep Success e screenshotOnStep Failure).

enableStepScreenshots()

Abilita tutte le opzioni relative agli screenshot (screenshotOnStepStart, screenshotOnStep Success e screenshotOnStep Failure). Per impostazione predefinita, tutti questi metodi sono abilitati.

getScreenshotOnStepFailure()

Restituisce se il canary acquisisce uno screenshot dopo che una fase fallisce.

getScreenshotOnStepStart()

Restituisce se il canary acquisisce uno screenshot prima di iniziare una fase.

getScreenshotOnStepSuccess()

Restituisce se il canary acquisisce uno screenshot dopo aver completato correttamente una fase.

withScreenshotOnStepStart(screenshotOnStepInizio)

Accetta un argomento booleano, che indica se eseguire uno screenshot prima di iniziare una fase.

withScreenshotOnStepSuccess(screenshotOnStepSuccesso)

Accetta un argomento booleano, che indica se acquisire uno screenshot dopo aver completato correttamente una fase.

withScreenshotOnStepFailure(screenshotOnStepFallimento)

Accetta un argomento booleano, che indica se acquisire uno screenshot dopo il fallimento di una fase.

Utilizzo in canary dell'interfaccia utente

Innanzitutto, importa la dipendenza di Synthetics e recupera la configurazione.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');
```

```
// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();
```

Quindi, imposta la configurazione per ogni opzione chiamando il metodo setConfig utilizzando una delle opzioni seguenti.

```
// Set configuration values
    synConfig.setConfig({
        screenshotOnStepStart: true,
        screenshotOnStepSuccess: false,
        screenshotOnStepFailure: false
    });
```

Oppure

```
synConfig.withScreenshotOnStepStart(false).withScreenshotOnStepSuccess(true).withScreenshotOnSt
```

Per disabilitare tutti gli screenshot, usa la funzione disableStepScreenshots () come in questo esempio.

```
synConfig.disableStepScreenshots();
```

Puoi abilitare e disabilitare gli screenshot in qualsiasi punto del codice. Ad esempio, per disabilitare gli screenshot solo per una fase, disattivali prima di eseguire tale fase e quindi attivali dopo la fase.

setConfig(options) per canary dell'iAPI

Per i canary dell'API, setConfig può includere i seguenti parametri booleani:

- `continueOnHttpStepFailure`(boolean) — Indica se continuare a eseguire lo script canary dopo un errore in un passaggio HTTP (si riferisce alla funzione). executeHttpStep Se le fasi hanno esito negativo, l'esecuzione del canary verrà comunque contrassegnata come fallita. Il valore predefinito è `true`.

Monitoraggio visivo

Il monitoraggio visivo confronta gli screenshot acquisiti durante un'esecuzione del canary con quelli acquisiti durante un'esecuzione del canary di riferimento. Se la discrepanza tra i due screenshot

supera una percentuale di soglia, il canary fallisce e puoi vedere le aree con differenze di colore evidenziate nel report di esecuzione del canary. Il monitoraggio visivo è supportato nei canaries che eseguono syn-puppeteer-node -3.2 e versioni successive. Attualmente non è supportato nei canary che eseguono Python e Selenium.

Per abilitare il monitoraggio visivo, aggiungi la seguente riga di codice allo script canary. Per ulteriori dettagli, consulta SyntheticsConfiguration classe.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

La prima volta che il canary viene eseguito correttamente dopo che questa riga è stata aggiunta allo script, utilizza gli screenshot acquisiti durante l'esecuzione come riferimento per il confronto. Dopo la prima esecuzione del canarino, puoi usare la CloudWatch console per modificare il canarino per eseguire una delle seguenti operazioni:

- Imposta l'esecuzione successiva del canary come nuovo riferimento.
- Disegna dei limiti sullo screenshot di riferimento corrente per designare le aree dello screenshot da ignorare durante i confronti visivi.
- Rimuovi uno screenshot dall'utilizzo per il monitoraggio visivo.

Per ulteriori informazioni sull'utilizzo della CloudWatch console per modificare un canarino, consulta. Modifica o eliminazione di un canary

Altre opzioni per il monitoraggio visivo

Configurazione sintetica. withVisualVarianceThresholdPercentage(Percentuale desiderata)

Imposta la percentuale accettabile per la varianza dello screenshot nei confronti visivi.

Configurazione sintetica. withVisualVarianceHighlightHexColor(» #fafa00 «)

Imposta il colore di evidenziazione che designa le aree di varianza quando si esaminano i report di esecuzione del canary che utilizzano il monitoraggio visivo.

Configurazione sintetica. withFailCanaryRunOnVisualVariance(Fail Canary)

Imposta se il canary fallisce o meno quando vi è una differenza visiva superiore alla soglia. L'impostazione predefinita è far fallire il canary.

Synthetics logger

SyntheticsLogger scrive i log out sia sulla console che su un file di registro locale allo stesso livello di registro. Questo file di log viene scritto in entrambe le posizioni solo se il livello di log è pari o inferiore al livello di registrazione desiderato della funzione di log chiamata.

Le istruzioni di registrazione nel file di log locale sono precedute da "DEBUG: ", "INFO: " e così via per corrispondere al livello di log della funzione che è stata chiamata.

Puoi usare SyntheticsLogger, supponendo che tu voglia eseguire la Synthetics Library allo stesso livello di registro del tuo Synthetics Canary Logging.

L'utilizzo di non SyntheticsLogger è necessario per creare un file di registro da caricare nella posizione dei risultati S3. Puoi invece creare un file di log diverso nella cartella `/tmp`. Tutti i file creati nella cartella `/tmp` vengono caricati nella posizione dei risultati in S3 come artefatti.

Per utilizzare il logger della libreria Synthetics:

```
const log = require('SyntheticsLogger');
```

Definizioni utili delle funzioni:

log.debug(***message***, ***ex***);

Parametri: *message* è il messaggio da registrare, ed *ex* è l'eccezione, se presente, da registrare

Esempio:

```
log.debug("Starting step - login.");
```

log.error(***message***, ***ex***);

Parametri: *message* è il messaggio da registrare, ed *ex* è l'eccezione, se presente, da registrare

Esempio:

```
try {
  await login();
catch (ex) {
  log.error("Error encountered in step - login.", ex);
}
```

log.info(***message***, ***ex***);

Parametri: *message* è il messaggio da registrare, ed *ex* è l'eccezione, se presente, da registrare

Esempio:

```
log.info("Successfully completed step - login.");
```

log.log(***message***, ***ex***);

Questo è un alias per `log.info`.

Parametri: *message* è il messaggio da registrare, ed *ex* è l'eccezione, se presente, da registrare

Esempio:

```
 log.log("Successfully completed step - login.");
```

log.warn(***message***, ***ex***);

Parametri: *message* è il messaggio da registrare, ed *ex* è l'eccezione, se presente, da registrare

Esempio:

```
log.warn("Exception encountered trying to publish CloudWatch Metric.", ex);
```

SyntheticsLogHelper classe

La classe `SyntheticsLogHelper` è disponibile nel runtime `syn-nodejs-puppeteer-3.2` e nei runtime successivi. È già inizializzato nella libreria CloudWatch Synthetics ed è configurato con la configurazione Synthetics. È possibile aggiungerla come dipendenza nello script. Questa classe consente di sanificare URL, intestazioni e messaggi di errore per oscurare informazioni sensibili.

> **Note**
>
> Synthetics sanifica tutti gli URL e i messaggi di errore registrati prima di includerli in registri, report, file HAR ed errori di esecuzione del canary in base all'impostazione di configurazione Synthetics `restrictedUrlParameters`. Devi usare `getSanitizedUrl` o `getSanitizedErrorMessage` solo se stai registrando URL o errori nello script. Synthetics

non memorizza alcun artefatto canary tranne che per gli errori del canary generati dallo script. Gli artefatti di esecuzione del canary sono archiviati nel tuo account cliente. Per ulteriori informazioni, consulta Considerazioni sulla sicurezza per Canary Synthetics.

getSanitizedUrl(url, stepConfig = null)

Questa funzione è disponibile nella versione `syn-nodejs-puppeteer-3.2` e versione successiva. Restituisce stringhe di URL sanificate in base alla configurazione. Puoi scegliere di oscurare parametri di URL sensibili ad esempio password e access_token impostando la proprietà `restrictedUrlParameters`. Per impostazione predefinita, le password negli URL vengono oscurate. Se necessario, puoi abilitare le password degli URL impostando `includeUrlPassword` su “true”.

Questa funzione genera un errore se l'URL passato non è un URL valido.

Parametri

- *`url`* è una stringa ed è l'URL da sanificare.
- *`stepConfig`* (Optional) sovrascrive la configurazione Synthetics globale per questa funzione. Se `stepConfig` non viene passato, la configurazione globale viene utilizzata per sanificare l'URL.

Esempio

In questo esempio viene utilizzato l'URL di esempio riportato di seguito: `https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`. In questo esempio, `access_token` contiene le tue informazioni sensibili che non devono essere registrate. Ricorda che i servizi Synthetics non memorizzano alcun artefatto di esecuzione del canary. Artefatti ad esempio log, screenshot e report sono tutti memorizzati in un bucket Amazon S3 nel tuo account cliente.

La prima fase consiste nell'impostare la configurazione Synthetics.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');
```

```
// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
});
```

Quindi, sanifica e registra l'URL

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200');
```

Questa procedura registra quanto segue nel log del canary.

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

Puoi ignorare la configurazione Synthetics per un URL passando un parametro facoltativo contenente le opzioni di configurazione Synthetics, come nell'esempio seguente.

```
const urlConfig = {
   restrictedUrlParameters = ['*']
};
const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200', urlConfig);
logger.info('My example url is: ' + sanitizedUrl);
```

Nell'esempio precedente vengono oscurati tutti i parametri di query e vengono registrati come segue:

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=REDACTED&expires_in=REDACTED
```

getSanitizedErrorMessaggio

Questa funzione è disponibile nella versione `syn-nodejs-puppeteer-3.2` e versione successiva. Restituisce stringhe di errore sanificate sanificando qualsiasi URL presente in base

alla configurazione Synthetics. Puoi scegliere di sovrascrivere la configurazione Synthetics globale quando chiami questa funzione passando un parametro `stepConfig`.

Parametri

- *`error`* è l'errore da sanificare. Può essere un oggetto Error o una stringa.
- *`stepConfig`* (Optional) sovrascrive la configurazione Synthetics globale per questa funzione. Se `stepConfig` non viene passato, la configurazione globale viene utilizzata per sanificare l'URL.

Esempio

In questo esempio viene utilizzato l'errore seguente: `Failed to load url: https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`

La prima fase consiste nell'impostare la configurazione Synthetics.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
});
```

Quindi, sanifica e registra il messaggio di errore

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

try {
   // Your code which can throw an error containing url which your script logs
} catch (error) {
    const sanitizedErrorMessage = synthetics.getSanitizedErrorMessage(errorMessage);
    logger.info(sanitizedErrorMessage);
}
```

Questa procedura registra quanto segue nel log del canary.

```
Failed to load url: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

getSanitizedHeaders(intestazioni, stepconfig=null)

Questa funzione è disponibile nella versione `syn-nodejs-puppeteer-3.2` e versione successiva. Restituisce intestazioni sanificate in base alla proprietà `restrictedHeaders` di `syntheticsConfiguration`. Le intestazioni specificate nella proprietà `restrictedHeaders` vengono oscurate da log, file HAR e report.

Parametri

- *`headers`* è un oggetto contenente le intestazioni da sanificare.
- *`stepConfig`* (Optional) sovrascrive la configurazione Synthetics globale per questa funzione. Se `stepConfig` non viene passato, la configurazione globale viene utilizzata per sanificare le intestazioni.

Classi e funzioni della libreria di Node.js che si applicano solo ai canary dell'interfaccia utente

Le seguenti funzioni della libreria CloudWatch Synthetics per Node.js sono utili solo per i canari dell'interfaccia utente.

Argomenti

- Classe Synthetics
- BrokenLinkCheckerReport classe
- SyntheticsLink classe

Classe Synthetics

Le seguenti funzioni sono nella classe Synthetics.

async addUserAgent (pagina,); userAgentString

Questa funzione si aggiunge *`userAgentString`*all'intestazione user-agent della pagina specificata.

Esempio:

```
await synthetics.addUserAgent(page, "MyApp-1.0");
```

Risultati nell'intestazione user-agent della pagina impostata su *browsers-user-agent-header-value*`MyApp-1.0`

async executeStep (StepName,, [StepConfig]); functionToExecute

Esegue il passaggio fornito, con registrazione avviata/riuscita/non riuscita, screenshot avviato/riuscito/non riuscito e parametro riuscito/non riuscito e durata.

> **Note**
>
> Se utilizzi il runtime `syn-nodejs-2.1` o versione successiva, puoi configurare se e quando vengono acquisite le schermate. Per ulteriori informazioni, consulta SyntheticsConfiguration classe.

La funzione `executeStep` svolge anche le operazioni seguenti:

- Registra che il passaggio è iniziato.
- Acquisisce uno screenshot denominato `<stepName>-starting`.
- Avvia un timer.
- Esegue la funzione fornita.
- Se la funzione viene restituita normalmente, conta come passaggio. Se la funzione genera, conta come errore.
- Termina il timer.
- Registra se il passaggio è riuscito o non è riuscito
- Prende uno screenshot chiamato `<stepName>-succeeded` o `<stepName>-failed`.
- Emette la metrica `stepName SuccessPercent`, 100 per passato o 0 per errore.
- Genera il parametro `stepName Duration` con un valore basato sull'ora di inizio e di fine del passaggio.
- Infine, restituisce ciò che il `functionToExecute` ha restituito o rigenera ciò che `functionToExecute` ha generato.

Se il canary usa il runtime `syn-nodejs-2.0` o versioni successive, questa funzione aggiunge anche un riepilogo dell'esecuzione delle fasi al report del canary. Il riepilogo include dettagli su ogni fase, ad

esempio l'ora di inizio, l'ora di fine, lo stato (PASSED/FAILED), il motivo dell'errore (in caso di errore) e le schermate acquisite durante l'esecuzione di ogni fase.

Esempio:

```
await synthetics.executeStep('navigateToUrl', async function (timeoutInMillis = 30000)
 {
           await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});});
```

Risposta:

Restituisce ciò che `functionToExecute` restituisce.

Aggiornamenti con syn-nodejs-2.2

A partire da`syn-nodejs-2.2`, puoi facoltativamente passare le configurazioni dei passaggi per sovrascrivere le configurazioni CloudWatch Synthetics a livello di passaggio. Per un elenco di opzioni che puoi trasferire a `executeStep`, consulta SyntheticsConfiguration classe.

L'esempio riportato di seguito sostituisce la configurazione `false` di default per `continueOnStepFailure` su `true` e specifica quando acquisire screenshot.

```
var stepConfig = {
    'continueOnStepFailure': true,
    'screenshotOnStepStart': false,
    'screenshotOnStepSuccess': true,
    'screenshotOnStepFailure': false
}

await executeStep('Navigate to amazon', async function (timeoutInMillis = 30000) {
      await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});
 }, stepConfig);
```

getDefaultLaunchOpzioni ();

La `getDefaultLaunchOptions()` funzione restituisce le opzioni di avvio del browser utilizzate da CloudWatch Synthetics. Per ulteriori informazioni, consulta Tipo di opzioni di avvio.

```
// This function returns default launch options used by Synthetics.
const defaultOptions = await synthetics.getDefaultLaunchOptions();
```

getPage();

Restituisce la pagina corrente aperta come oggetto Puppeteer. Per ulteriori informazioni, consulta l'API Puppeteer v1.14.0.

Esempio:

```
let page = synthetics.getPage();
```

Risposta:

La pagina (oggetto Puppeteer) attualmente aperta nella sessione corrente del browser.

getRequestResponseLogHelper();

> **Important**
>
> Nei canary che usano il runtime `syn-nodejs-puppeteer-3.2` o versione successiva, questa funzione è resa obsoleta insieme alla classe `RequestResponseLogHelper`. Qualsiasi utilizzo di questa funzione fa apparire un avviso nei log del canary. Questa funzione verrà rimossa nelle future versioni di runtime. Se utilizzi questa funzione, utilizza invece RequestResponseLogHelper classe.

Utilizza questa funzione come modello di builder per modificare i flag di registrazione della richiesta e della risposta.

Esempio:

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper().withLogRequestHeaders(fals
```

Risposta:

```
{RequestResponseLogHelper}
```

launch(options)

Le opzioni per questa funzione sono disponibili solo nella versione di runtime `syn-nodejs-2.1` o versioni successive.

Questa funzione è utilizzata solo per i canary dell'interfaccia utente. Chiude il browser esistente e ne avvia uno nuovo.

> **Note**
>
> CloudWatch Synthetics avvia sempre un browser prima di iniziare a eseguire lo script. Non è necessario chiamare “launch()” a meno che non desideri avviare un nuovo browser con opzioni personalizzate.

(options) è un insieme configurabile di opzioni da impostare sul browser. Per ulteriori informazioni, consulta Tipo di opzioni di avvio.

Se chiami questa funzione senza “options”, Synthetics avvia un browser con argomenti predefiniti, `executablePath` e `defaultViewport`. La finestra di visualizzazione predefinita in CloudWatch Synthetics è 1920 x 1080.

È possibile sovrascrivere i parametri di avvio utilizzati da CloudWatch Synthetics e passare parametri aggiuntivi all'avvio del browser. Ad esempio, il seguente frammento di codice avvia un browser con argomenti predefiniti e un percorso eseguibile predefinito, ma con un'area di visualizzazione di 800 x 600.

```
await synthetics.launch({
        defaultViewport: {
            "deviceScaleFactor": 1,
            "width": 800,
            "height": 600
    }});
```

Il seguente codice di esempio aggiunge un nuovo `ignoreHTTPSErrors` parametro ai parametri di avvio di CloudWatch Synthetics:

```
await synthetics.launch({
        ignoreHTTPSErrors: true
 });
```

Puoi disabilitare la sicurezza web aggiungendo un `--disable-web-security` flag ad args nei parametri di avvio di CloudWatch Synthetics:

```
// This function adds the --disable-web-security flag to the launch parameters
```

```
const defaultOptions = await synthetics.getDefaultLaunchOptions();
const launchArgs = [...defaultOptions.args, '--disable-web-security'];
await synthetics.launch({
     args: launchArgs
  });
```

RequestResponseLogHelper classe

> **Important**
>
> Nei canary che usano il runtime `syn-nodejs-puppeteer-3.2` o versioni successive, questa classe è resa obsoleta. Qualsiasi uso di questa classe fa apparire un avviso nei log del canary. Questa funzione verrà rimossa nelle future versioni di runtime. Se utilizzi questa funzione, utilizza invece RequestResponseLogHelper classe.

Gestisce la configurazione granulare e la creazione di rappresentazioni di stringa di payload di richiesta e risposta.

```
class RequestResponseLogHelper {

    constructor () {
        this.request = {url: true, resourceType: false, method: false, headers: false,
 postData: false};
        this.response = {status: true, statusText: true, url: true, remoteAddress:
 false, headers: false};
    }

    withLogRequestUrl(logRequestUrl);

    withLogRequestResourceType(logRequestResourceType);

    withLogRequestMethod(logRequestMethod);

    withLogRequestHeaders(logRequestHeaders);

    withLogRequestPostData(logRequestPostData);


    withLogResponseStatus(logResponseStatus);

    withLogResponseStatusText(logResponseStatusText);
```

```
    withLogResponseUrl(logResponseUrl);

    withLogResponseRemoteAddress(logResponseRemoteAddress);

    withLogResponseHeaders(logResponseHeaders);
```

Esempio:

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper()
.withLogRequestPostData(true)
.withLogRequestHeaders(true)
.withLogResponseHeaders(true));
```

Risposta:

```
{RequestResponseLogHelper}
```

setRequestResponseLogHelper();

> **Important**
>
> Nei canary che usano il runtime `syn-nodejs-puppeteer-3.2` o versione successiva, questa funzione è resa obsoleta insieme alla classe `RequestResponseLogHelper`. Qualsiasi utilizzo di questa funzione fa apparire un avviso nei log del canary. Questa funzione verrà rimossa nelle future versioni di runtime. Se utilizzi questa funzione, utilizza invece RequestResponseLogHelper classe.

Utilizza questa funzione come modello di builder per impostare i flag di registrazione richiesta e risposta.

Esempio:

```
synthetics.setRequestResponseLogHelper().withLogRequestHeaders(true).withLogResponseHeaders(tru
```

Risposta:

```
{RequestResponseLogHelper}
```

async takeScreenshot(name, suffix);

Acquisisce uno screenshot (.PNG) della pagina corrente con nome e suffisso (opzionale).

Esempio:

```
await synthetics.takeScreenshot("navigateToUrl", "loaded")
```

Questo esempio acquisisce e carica uno screenshot denominato `01-navigateToUrl-loaded.png` nel bucket S3 del canary.

Puoi acquisire uno screenshot per una particolare fase del canary passando `stepName` come primo parametro. Gli screenshot sono collegati alla fase del canary nei report, per aiutarti a tenere traccia di ogni fase durante il debug.

CloudWatch Synthetics canaries acquisisce automaticamente gli screenshot prima di iniziare un passaggio (`executeStep`la funzione) e dopo il completamento del passaggio (a meno che non si configuri il canary per disabilitare gli screenshot). Puoi acquisire più screenshot passando il nome della fase nella funzione `takeScreenshot`.

L'esempio seguente acquisisce screenshot con `signupForm` come valore di `stepName`. Lo screenshot sarà chiamato `02-signupForm-address` e sarà collegato alla fase denominata `signupForm` nel report del canary.

```
await synthetics.takeScreenshot('signupForm', 'address')
```

BrokenLinkCheckerReport classe

Questa classe offre metodi per aggiungere un collegamento Synthetics. È supportata solo sui canary che utilizzano la versione `syn-nodejs-2.0-beta` del runtime o versioni successive.

Per utilizzare `BrokenLinkCheckerReport`, includi le seguenti righe nello script:

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');

const brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

Definizioni utili delle funzioni:

addLink(***syntheticsLink***, isBroken)

`syntheticsLink` è un `SyntheticsLink` che rappresenta un collegamento. Questa funzione aggiunge il collegamento in base al codice di stato. Per impostazione predefinita, considera un collegamento interrotto se il codice di stato non è disponibile o se il codice di stato è 400 o superiore. Puoi ignorare questo comportamento predefinito passando il parametro opzionale `isBrokenLink` con un valore di `true` o `false`.

Questa funzione non ha un valore di restituzione.

getLinks()

Questa funzione restituisce una matrice di oggetti `SyntheticsLink` inclusi nel report di controllo del collegamento interrotto.

getTotalBrokenCollegamenti ()

Questa funzione restituisce un numero che rappresenta il numero totale di collegamenti interrotti.

getTotalLinksControllato ()

Questa funzione restituisce un numero che rappresenta il numero totale di collegamenti inclusi nel report.

Come usare BrokenLinkCheckerReport

Il seguente frammento di codice di script canary mostra un esempio di spostamento a un collegamento e aggiunta al report di controllo del collegamento interrotto.

1. Importa `SyntheticsLink`, `BrokenLinkCheckerReport` e `Synthetics`.

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');
const SyntheticsLink = require('SyntheticsLink');

// Synthetics dependency
const synthetics = require('Synthetics');
```

2. Per aggiungere un collegamento al report, crea un'istanza di `BrokenLinkCheckerReport`.

```
let brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

3. Passa all'URL e aggiungilo al report di controllo del collegamento interrotto.

```
let url = "https://amazon.com";
```

```
let syntheticsLink = new SyntheticsLink(url);

// Navigate to the url.
let page = await synthetics.getPage();

// Create a new instance of Synthetics Link
let link = new SyntheticsLink(url)

try {
    const response = await page.goto(url, {waitUntil: 'domcontentloaded', timeout:
 30000});
} catch (ex) {
    // Add failure reason if navigation fails.
    link.withFailureReason(ex);
}

if (response) {
    // Capture screenshot of destination page
    let screenshotResult = await synthetics.takeScreenshot('amazon-home', 'loaded');

    // Add screenshot result to synthetics link
    link.addScreenshotResult(screenshotResult);

    // Add status code and status description to the link
    link.withStatusCode(response.status()).withStatusText(response.statusText())
}

// Add link to broken link checker report.
brokenLinkCheckerReport.addLink(link);
```

4. Aggiungi il report a Synthetics. Questa procedura crea un file JSON denominato `BrokenLinkCheckerReport.json` nel bucket S3 per ogni esecuzione del canary. Puoi visualizzare un report dei collegamenti nella console per ogni esecuzione del canary insieme a screenshot, log e file HAR.

```
await synthetics.addReport(brokenLinkCheckerReport);
```

SyntheticsLink classe

Questa classe offre metodi per impacchettare le informazioni. È supportata solo sui canary che utilizzano la versione `syn-nodejs-2.0-beta` del runtime o versioni successive.

Per utilizzare `SyntheticsLink`, includi le seguenti righe nello script:

```
const SyntheticsLink = require('SyntheticsLink');

const syntheticsLink = new SyntheticsLink("https://www.amazon.com");
```

Questa funzione restituisce `syntheticsLink`*`Object`*

Definizioni utili delle funzioni:

withUrl(***`url`***)

`url` è una stringa URL. Questa funzione restituisce `syntheticsLink`*`Object`*

withText(***`text`***)

`text` è una stringa che rappresenta il testo di ancoraggio. Questa funzione restituisce `syntheticsLink`*`Object`*. Aggiunge il testo di ancoraggio corrispondente al collegamento.

withParentUrl(***`ParentUrl)`***

`parentUrl` è una stringa che rappresenta l'URL principale (pagina di origine). Questa funzione restituisce `syntheticsLink`*`Object`*

withStatusCode(***`Codice di stato)`***

`statusCode` è una stringa che rappresenta il codice di stato. Questa funzione restituisce `syntheticsLink`*`Object`*

withFailureReason***`(Motivo dell'errore)`***

`failureReason` è una stringa che rappresenta il motivo dell'errore. Questa funzione restituisce `syntheticsLink`*`Object`*

addScreenshotResult(***`Risultato dello screenshot`***)

`screenshotResult` è un oggetto. Si tratta di un'istanza di `ScreenshotResult` restituita dalla funzione `takeScreenshot` di Synthetics. L'oggetto include i seguenti elementi:

- `fileName`: una stringa che rappresenta `screenshotFileName`
- `pageUrl` (facoltativo)

- `error` (facoltativo)

Classi e funzioni della libreria di Node.js che si applicano solo ai canary dell'API

Le seguenti CloudWatch funzioni della libreria Synthetics per Node.js sono utili solo per API canaries.

Argomenti

- executeHttpStep(StepName, RequestOptions, [callback], [StepConfig])

executeHttpStep(StepName, RequestOptions, [callback], [StepConfig])

Esegue la richiesta HTTP fornita come fase e pubblica `SuccessPercent` (pass/fail) e parametri `Duration`.

executeHttpStepUtilizza automaticamente funzioni native HTTP o HTTPS, a seconda del protocollo specificato nella richiesta.

Questa funzione aggiunge anche un riepilogo dell'esecuzione delle fasi al report del canary. Il riepilogo include dettagli su ogni richiesta HTTP, ad esempio:

- Ora di inizio
- Ora di fine
- Stato (PASSED/FAILED)
- Motivo dell'errore, se non è riuscita
- Dettagli delle chiamate HTTP, ad esempio intestazioni di richiesta/risposta, corpo, codice di stato, messaggio di stato e tempi di prestazioni.

Parametri

stepName(***`String`***)

Specifica il nome della fase. Questo nome viene utilizzato anche per pubblicare le CloudWatch metriche relative a questa fase.

requestOptions(***`Object or String`***)

Il valore di questo parametro può essere un URL, una stringa URL o un oggetto. Se si tratta di un oggetto, deve essere un insieme di opzioni configurabili per effettuare una richiesta HTTP. Supporta tutte le opzioni in http.request(options[, callback]) nella documentazione di Node.js.

Oltre a queste opzioni Node.js, requestOptions supporta il parametro aggiuntivo `body`. Puoi utilizzare il parametro `body` per trasferire i dati come corpo della richiesta.

callback(***response***)

(Facoltativo) Questa è una funzione utente che viene richiamata con la risposta HTTP. La risposta è del tipo Class: http. IncomingMessage.

stepConfig(***object***)

(Facoltativo) Utilizza questo parametro per sostituire le configurazioni Synthetics globali con una configurazione diversa per questa fase.

Esempi di utilizzo executeHttpStep

La seguente serie di esempi si basa l'uno sull'altro per illustrare i vari usi di questa opzione.

Questo primo esempio configura i parametri della richiesta. Puoi trasferire un URL come requestOptions:

```
let requestOptions = 'https://www.amazon.com';
```

Oppure puoi trasferire una serie di opzioni:

```
let requestOptions = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };
```

L'esempio seguente crea una funzione di callback che accetta una risposta. Per impostazione predefinita, se non si specifica il callback, CloudWatch Synthetics verifica che lo stato sia compreso tra 200 e 299 inclusi.

```
// Handle validation for positive scenario
    const callback = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }
```

```
            let responseBody = '';
            res.on('data', (d) => {
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };
```

L'esempio successivo crea una configurazione per questo passaggio che sovrascrive la configurazione globale di Synthetics CloudWatch . La configurazione della fase in questo esempio consente le intestazioni della richiesta, le intestazioni della risposta, il corpo della richiesta (dopo i dati) e il corpo della risposta nel report e di limitare i valori di intestazione “X-Amz-Security-Token” e “Authorization”. Per impostazione predefinita, questi valori non vengono inclusi nel report per motivi di sicurezza. Se scegli di includerli, i dati vengono archiviati solo nel bucket S3.

```
// By default headers, post data, and response body are not included in the report for
 security reasons.
// Change the configuration at global level or add as step configuration for individual
 steps
let stepConfig = {
    includeRequestHeaders: true,
    includeResponseHeaders: true,
    restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted header
 values do not appear in report generated.
    includeRequestBody: true,
    includeResponseBody: true
};
```

Questo ultimo esempio passa la richiesta a executeHttpStepe assegna un nome alla fase.

```
await synthetics.executeHttpStep('Verify GET products API', requestOptions, callback,
 stepConfig);
```

Con questo set di esempi, CloudWatch Synthetics aggiunge i dettagli di ogni passaggio del rapporto e produce metriche per ogni passaggio utilizzando StepName.

Vedrai i parametri `successPercent` e `duration` per la fase `Verify GET products API`. Puoi monitorare le prestazioni delle API monitorando i parametri per le fasi delle chiamate API.

Per uno script completo di esempio che utilizza queste funzioni, consulta Canary dell'API in più fasi.

Funzioni di libreria disponibili per gli script canary Python che usano Selenium

Questa sezione elenca le funzioni di libreria di Selenium disponibili per gli script canary Python.

Argomenti

- Classi e funzioni della libreria Python e Selenium che si applicano a tutti i canary
- Classi e funzioni della libreria Python e Selenium che si applicano solo ai canary dell'interfaccia utente

Classi e funzioni della libreria Python e Selenium che si applicano a tutti i canary

Le seguenti funzioni della libreria CloudWatch Synthetics Selenium per Python sono utili per tutti i canarini.

Argomenti

- SyntheticsConfiguration classe
- SyntheticsLogger classe

SyntheticsConfiguration classe

È possibile utilizzare la SyntheticsConfiguration classe per configurare il comportamento delle funzioni della libreria Synthetics. Ad esempio, puoi utilizzare questa classe per configurare la funzione `executeStep()` per non acquisire screenshot.

È possibile impostare le configurazioni CloudWatch Synthetics a livello globale.

Definizioni della funzione:

set_config(options)

```
from aws_synthetics.common import synthetics_configuration
```

`options` è un oggetto, che è un insieme di opzioni configurabili per il tuo canary. Le seguenti sezioni spiegano i campi possibili in *`options`*.

- `screenshot_on_step_start` (booleano): indica se fare uno screenshot prima di iniziare una fase.
- `screenshot_on_step_success` (booleano): indica se acquisire uno screenshot dopo aver completato una fase riuscita.
- `screenshot_on_step_failure` (booleano): indica se acquisire uno screenshot dopo il fallimento di una fase.

with_screenshot_on_step_start(screenshot_on_step_start)

Accetta un argomento booleano, che indica se eseguire uno screenshot prima di iniziare una fase.

with_screenshot_on_step_success(screenshot_on_step_success)

Accetta un argomento booleano, che indica se acquisire uno screenshot dopo aver completato correttamente una fase.

with_screenshot_on_step_failure(screenshot_on_step_failure)

Accetta un argomento booleano, che indica se acquisire uno screenshot dopo il fallimento di una fase.

get_screenshot_on_step_start()

Restituisce se acquisire uno screenshot prima di iniziare una fase.

get_screenshot_on_step_success()

Restituisce se acquisire uno screenshot dopo aver completato correttamente una fase.

get_screenshot_on_step_failure()

Restituisce se acquisire uno screenshot dopo che una fase fallisce.

disable_step_screenshots()

Disabilita tutte le opzioni di screenshot (get_screenshot_on_step_start, get_screenshot_on_step_success e get_screenshot_on_step_failure).

enable_step_screenshots()

Abilita tutte le opzioni di screenshot (get_screenshot_on_step_start, get_screenshot_on_step_success e get_screenshot_on_step_failure). Per impostazione predefinita, tutti questi metodi sono abilitati.

setConfig (opzioni) relative alle metriche CloudWatch

Per i canary che utilizzano `syn-python-selenium-1.1` o versione successiva, (options) per setConfig può includere i seguenti parametri booleani che determinano quali parametri vengono pubblicati dal canary. Il valore predefinito per ciascuna di queste opzioni è `true`. Le opzioni che iniziano con `aggregated` determinano se il parametro viene emesso senza destinazione `CanaryName`. Puoi usare questi parametri per visualizzare i risultati aggregati per tutti i canary. Le altre opzioni determinano se il parametro viene emesso con la dimensione `CanaryName`. Puoi usare questi parametri per visualizzare i risultati per ogni singolo canary.

Per un elenco delle CloudWatch metriche emesse dai canarini, vedi. CloudWatch metriche pubblicate da canaries

- `failed_canary_metric` (booleano): indica se emettere il parametro `Failed` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `failed_requests_metric` (booleano): indica se emettere il parametro `Failed requests` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `2xx_metric` (booleano): indica se emettere il parametro `2xx` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `4xx_metric` (booleano): indica se emettere il parametro `4xx` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `5xx_metric` (booleano): indica se emettere il parametro `5xx` (con la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `step_duration_metric` (booleano): indica se emettere il parametro `Step duration` (con le dimensioni `CanaryName StepName`) per questo canary. Il valore predefinito è `true`.
- `step_success_metric` (booleano): indica se emettere il parametro `Step success` (con le dimensioni `CanaryName StepName`) per questo canary. Il valore predefinito è `true`.
- `aggregated_failed_canary_metric` (booleano): indica se emettere il parametro `Failed` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `aggregated_failed_requests_metric` (booleano): indica se emettere il parametro `Failed Requests` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `aggregated_2xx_metric` (booleano): indica se emettere il parametro `2xx` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.

- `aggregated_4xx_metric` (booleano): indica se emettere il parametro `4xx` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.
- `aggregated_5xx_metric` (booleano): indica se emettere il parametro `5xx` (senza la dimensione `CanaryName`) per questo canary. Il valore predefinito è `true`.

with_2xx_metric(2xx_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `2xx` con dimensione `CanaryName` per questo canary.

with_4xx_metric(4xx_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `4xx` con dimensione `CanaryName` per questo canary.

with_5xx_metric(5xx_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `5xx` con dimensione `CanaryName` per questo canary.

withAggregated2xxMetric(aggregated2xxMetric)

Accetta un argomento booleano, che specifica se emettere un parametro `2xx` senza alcuna dimensione per questo canary.

withAggregated4xxMetric(aggregated4xxMetric)

Accetta un argomento booleano, che specifica se emettere un parametro `4xx` senza alcuna dimensione per questo canary.

with_aggregated_5xx_metric(aggregated_5xx_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `5xx` senza alcuna dimensione per questo canary.

with_aggregated_failed_canary_metric(aggregated_failed_canary_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `Failed` senza alcuna dimensione per questo canary.

with_aggregated_failed_requests_metric(aggregated_failed_requests_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `Failed requests` senza alcuna dimensione per questo canary.

with_failed_canary_metric(failed_canary_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `Failed` con dimensione `CanaryName` per questo canary.

with_failed_requests_metric(failed_requests_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `Failed requests` con dimensione `CanaryName` per questo canary.

with_step_duration_metric(step_duration_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `Duration` con dimensione `CanaryName` per questo canary.

with_step_success_metric(step_success_metric)

Accetta un argomento booleano, che specifica se emettere un parametro `StepSuccess` con dimensione `CanaryName` per questo canary.

Metodi per abilitare o disabilitare i parametri

disable_aggregated_request_metrics()

Disabilita il canary dall'emettere tutti i parametri delle richieste emessi senza dimensione `CanaryName`.

disable_request_metrics()

Disabilita tutti i parametri delle richieste, inclusi i parametri dei canary e i parametri aggregati in tutti i canary.

disable_step_metrics()

Disabilita tutti i parametri dei passaggi, inclusi sia quelli di riuscita che di durata.

enable_aggregated_request_metrics()

Consente al canary di emettere tutti i parametri della richiesta emessi senza la dimensione `CanaryName`.

enable_request_metrics()

Abilita tutti i parametri delle richieste, inclusi i parametri dei canary e i parametri aggregati in tutti i canary.

enable_step_metrics()

Abilita tutti i parametri dei passaggi, inclusi sia quelli di riuscita che di durata.

Utilizzo in canary dell'interfaccia utente

Innanzitutto, importa la dipendenza di Synthetics e recupera la configurazione. Quindi, imposta la configurazione per ogni opzione chiamando il metodo setConfig utilizzando una delle opzioni seguenti.

```
from aws_synthetics.common import synthetics_configuration

synthetics_configuration.set_config(
     {
        "screenshot_on_step_start": False,
        "screenshot_on_step_success": False,
        "screenshot_on_step_failure": True
     }
)

or
```

Oppure

```
synthetics_configuration.with_screenshot_on_step_start(False).with_screenshot_on_step_success(F
```

Per disabilitare tutte le schermate, utilizzate la funzione disableStepScreenshots () come in questo esempio.

```
synthetics_configuration.disable_step_screenshots()
```

Puoi abilitare e disabilitare gli screenshot in qualsiasi punto del codice. Ad esempio, per disabilitare gli screenshot solo per una fase, disattivali prima di eseguire tale fase e quindi attivali dopo la fase.

set_config(options) per canary dell'interfaccia utente

A partire da `syn-python-selenium-1.1`, per i canary dell'interfaccia utente, `set_config` può includere i seguenti parametri booleani:

- `continue_on_step_failure` (booleano): indica se continuare con l'esecuzione dello script canary dopo che una fase ha esito negativo (questo si riferisce alla funzione executeStep). Se le fasi hanno esito negativo, l'esecuzione del canary verrà comunque contrassegnata come fallita. Il valore predefinito è `false`.

SyntheticsLogger classe

`synthetics_logger` scrive i log sia nella console che in un file di log locale allo stesso livello di log. Questo file di log viene scritto in entrambe le posizioni solo se il livello di log è pari o inferiore al livello di registrazione desiderato della funzione di log chiamata.

Le istruzioni di registrazione nel file di log locale sono precedute da "DEBUG: ", "INFO: " e così via per corrispondere al livello di log della funzione che è stata chiamata.

L'utilizzo di `synthetics_logger` non è necessario per creare un file di log che viene caricato nel percorso dei risultati di Amazon S3. Puoi invece creare un file di log diverso nella cartella `/tmp`. Tutti i file creati nella cartella `/tmp` vengono caricati nella posizione dei risultati nel bucket S3 come artefatti.

Per utilizzare `synthetics_logger`:

```
from aws_synthetics.common import synthetics_logger
```

Definizioni utili delle funzioni:

Ottenere il livello di log:

```
log_level = synthetics_logger.get_level()
```

Impostare il livello di log:

```
synthetics_logger.set_level()
```

Registrare un messaggio con un livello specificato. Il livello può essere `DEBUG`, `INFO`, `WARN` oppure `ERROR`, come negli esempi di sintassi seguenti:

```
synthetics_logger.debug(message, *args, **kwargs)
```

```
synthetics_logger.info(message, *args, **kwargs)
```

```
synthetics_logger.log(message, *args, **kwargs)
```

```
synthetics_logger.warn(message, *args, **kwargs)
```

```
synthetics_logger.error(message, *args, **kwargs)
```

Per informazioni sui parametri di debug, consulta la documentazione standard di Python all'indirizzo logging.debug

In queste funzioni di registrazione, `message` è la stringa di formato del messaggio. `args` sono gli argomenti che vengono uniti in `msg` utilizzando l'operatore di formattazione delle stringhe.

Ci sono tre argomenti della parola chiave in `kwargs`:

- `exc_info`: se non viene valutato come false (falso), aggiunge informazioni sulle eccezioni al messaggio di log.
- `stack_info`: imposta il valore predefinito su false (falso). Se true (vero), aggiunge le informazioni sullo stack al messaggio di log, inclusa la chiamata di registrazione effettiva.
- `extra`: il terzo argomento della parola chiave facoltativo, che puoi utilizzare per il passaggio a un dizionario utilizzato per popolare il `__dict__` di `LogRecord` creato per l'evento di log con attributi definiti dall'utente.

Esempi:

Registrare un messaggio con il livello `DEBUG`:

```
synthetics_logger.debug('Starting step - login.')
```

Registrare un messaggio con il livello `INFO`. `logger.log` è sinonimo di `logger.info`:

```
synthetics_logger.info('Successfully completed step - login.')
```

oppure

```
synthetics_logger.log('Successfully completed step - login.')
```

Registrare un messaggio con il livello WARN:

```
synthetics_logger.warn('Warning encountered trying to publish %s', 'CloudWatch Metric')
```

Registrare un messaggio con il livello ERROR:

```
synthetics_logger.error('Error encountered trying to publish %s', 'CloudWatch Metric')
```

Registrare un'eccezione:

```
synthetics_logger.exception(message, *args, **kwargs)
```

Registra un messaggio con il livello ERROR. Le informazioni sulle eccezioni vengono aggiunte al messaggio di log. Dovresti chiamare questa funzione solo da un gestore di eccezioni.

Per informazioni sui parametri di eccezione, consulta la documentazione standard di Python all'indirizzo logging.exception

`message` è la stringa di formato del messaggio. `args` sono gli argomenti che vengono uniti in `msg` utilizzando l'operatore di formattazione delle stringhe.

Ci sono tre argomenti della parola chiave in `kwargs`:

- `exc_info`: se non viene valutato come false (falso), aggiunge informazioni sulle eccezioni al messaggio di log.
- `stack_info`: imposta il valore predefinito su false (falso). Se true (vero), aggiunge le informazioni sullo stack al messaggio di log, inclusa la chiamata di registrazione effettiva.
- `extra`: il terzo argomento della parola chiave facoltativo, che puoi utilizzare per il passaggio a un dizionario utilizzato per popolare il `__dict__` di `LogRecord` creato per l'evento di registrazione con attributi definiti dall'utente.

Esempio:

```
synthetics_logger.exception('Error encountered trying to publish %s', 'CloudWatch
 Metric')
```

Classi e funzioni della libreria Python e Selenium che si applicano solo ai canary dell'interfaccia utente

Le seguenti funzioni della libreria CloudWatch Synthetics Selenium per Python sono utili solo per i canari dell'interfaccia utente.

Argomenti

- SyntheticsBrowser classe
- SyntheticsWebDriver classe

SyntheticsBrowser classe

Quando crei un'istanza del browser chiamando `synthetics_webdriver.Chrome()`, l'istanza del browser restituita è del tipo `SyntheticsBrowser`. La `SyntheticsBrowser` classe controlla e abilita lo script canary per guidare il browser, permettendo a Selenium di funzionare con WebDriver Synthetics. ChromeDriver

Oltre ai metodi standard di Selenium, offre anche i seguenti metodi.

set_viewport_size(width, height)

Imposta l'area di visualizzazione del browser. Esempio:

```
browser.set_viewport_size(1920, 1080)
```

save_screenshot(filename, suffix)

Salva gli screenshot nella directory `/tmp`. Gli screenshot vengono caricati da lì nella cartella degli artefatti del canary nel bucket S3.

filename è il nome del file per lo screenshot, e suffix è una stringa opzionale da utilizzare per nominare lo screenshot.

Esempio:

```
browser.save_screenshot('loaded.png', 'page1')
```

SyntheticsWebDriver classe

Per utilizzare questa classe, utilizza i seguenti elementi nello script:

```
from aws_synthetics.selenium import synthetics_webdriver
```

addExecutionError(errorMessage, ex);

`errorMessage` descrive l'errore e `ex` è l'eccezione che si verifica

Puoi utilizzare `add_execution_error` per impostare gli errori di esecuzione per il tuo canary. Fa fallire il canary senza interrompere l'esecuzione dello script. Inoltre, non influisce sui tuoi parametri `successPercent`.

È necessario tenere traccia degli errori come errori di esecuzione solo se non sono importanti per indicare il successo o il fallimento dello script canary.

Di seguito è illustrato un esempio dell'uso dell'elemento `add_execution_error`. Stai monitorando la disponibilità del tuo endpoint e acquisendo screenshot dopo che la pagina è stata caricata. Poiché l'errore di acquisizione di uno screenshot non determina la disponibilità dell'endpoint, puoi rilevare eventuali errori riscontrati durante l'acquisizione di screenshot e aggiungerli come errori di esecuzione. I parametri di disponibilità indicheranno comunque che l'endpoint è attivo e funzionante, ma lo stato del canary verrà contrassegnato come non riuscito. Il seguente blocco di codice di esempio rileva tale errore e lo aggiunge come errore di esecuzione.

```
try:
    browser.save_screenshot("loaded.png")
except Exception as ex:
   self.add_execution_error("Unable to take screenshot", ex)
```

add_user_agent(user_agent_str)

Aggiunge il valore di `user_agent_str` all'intestazione dell'agente utente del browser. È necessario assegnare `user_agent_str` prima di creare l'istanza del browser.

Esempio:

```
synthetics_webdriver.add_user_agent('MyApp-1.0')
```

execute_step(step_name, function_to_execute)

Elabora una funzione. Inoltre esegue le seguenti operazioni:

- Registra che il passaggio è iniziato.
- Acquisisce uno screenshot denominato `<stepName>-starting`.
- Avvia un timer.
- Esegue la funzione fornita.
- Se la funzione viene restituita normalmente, conta come passaggio. Se la funzione genera, conta come errore.
- Termina il timer.
- Registra se il passaggio è riuscito o non è riuscito
- Prende uno screenshot chiamato `<stepName>-succeeded` o `<stepName>-failed`.
- Emette la metrica `stepName SuccessPercent`, 100 per passato o 0 per errore.
- Genera il parametro `stepName Duration` con un valore basato sull'ora di inizio e di fine del passaggio.
- Infine, restituisce ciò che il `functionToExecute` ha restituito o rigenera ciò che `functionToExecute` ha generato.

Esempio:

```
from selenium.webdriver.common.by import By

def custom_actions():
        #verify contains
        browser.find_element(By.XPATH, "//*[@id=\"id_1\"][contains(text(),'login')]")
        #click a button
        browser.find_element(By.XPATH, '//*[@id="submit"]/a').click()

    await synthetics_webdriver.execute_step("verify_click", custom_actions)
```

Chrome()

Avvia un'istanza del browser Chromium e restituisce l'istanza creata del browser.

Esempio:

```
browser = synthetics_webdriver.Chrome()
browser.get("https://example.com/)
```

Per avviare un browser in modalità di navigazione in incognito, utilizza quanto segue:

```
add_argument('—-incognito')
```

Per aggiungere impostazioni proxy, utilizza quanto segue:

```
add_argument('--proxy-server=%s' % PROXY)
```

Esempio:

```
from selenium.webdriver.chrome.options import Options
chrome_options = Options()
chrome_options.add_argument("—-incognito")
browser = syn_webdriver.Chrome(chrome_options=chrome_options)
```

Pianificazione delle esecuzioni di canary usando cron

L'utilizzo di un'espressione cron ti offre flessibilità quando pianifichi un canary. Le espressioni cron contengono cinque o sei campi nell'ordine elencato nella tabella seguente. I campi sono separati da uno spazio. La sintassi varia a seconda che si utilizzi la CloudWatch console per creare il canarino o gli SDK. AWS CLI AWS Quando utilizzi la console, specifichi solo i primi cinque campi. Quando utilizzi gli AWS CLI o AWS SDK, specifichi tutti e sei i campi e devi specificare per il campo. `* Year`

Campo	Valori consentiti	Caratteri speciali consentiti
Minuti	0-59	, - * /
Ore	0-23	, - * /
D ay-of-month	1-31	, - * ? / L W
Mese	1-12 o JAN-DEC	, - * /
D ay-of-week	1-7 o SUN-SAT	, - * ? L #
Anno	*	

Caratteri speciali

- Il carattere ,(virgola) include più valori nell'espressione di un campo. Ad esempio, nel campo Month (Mese), JAN,FEB,MAR (GEN,FEB,MAR) include gennaio, febbraio e marzo.
- Il carattere speciale - (trattino) specifica gli intervalli. Nel campo Day (Giorno), 1-15 include i giorni dall'1 al 15 del mese specificato.
- Il carattere speciale * (asterisco) include tutti i valori nel campo. Nel campo Hours (Ore), * include ogni ora. Non è possibile utilizzare* in entrambi i ay-of-week campi D ay-of-month e D della stessa espressione. Se viene utilizzato in uno di tali campi, è necessario utilizzare ? nell'altro.
- Il carattere / (barra) specifica gli incrementi. Nel campo Minutes (Minuti), puoi inserire 1/10 per specificare ogni decimo minuto, a partire dal primo minuto dell'ora (ad esempio, l'11°, il 21° e il 31° minuto e così via).
- Il carattere ? (punto interrogativo) specifica un valore o un altro. Se inserisci 7 nel ay-of-month campo D e non ti interessa in che giorno della settimana è il settimo, puoi inserire? nel ay-of-week campo D.
- Il carattere jolly L nei ay-of-week campi D ay-of-month o D specifica l'ultimo giorno del mese o della settimana.
- Il carattere **`W`** jolly nel ay-of-month campo D specifica un giorno della settimana. Nel ay-of-month campo D, **`3W`** specifica il giorno della settimana più vicino al terzo giorno del mese.
- Il carattere jolly # nel ay-of-week campo D specifica una determinata istanza del giorno della settimana specificato all'interno di un mese. Ad esempio, 3#2 è il secondo martedì del mese. Il 3 fa riferimento a martedì perché è il terzo giorno di ogni settimana e il 2 fa riferimento al secondo giorno di questo tipo in un mese.

Limitazioni

- Non è possibile specificare i ay-of-week campi D ay-of-month e D nella stessa espressione cron. Se specifichi un valore o * (asterisco) in uno dei campi, devi usare un carattere ? (punto interrogativo) nell'altro campo.
- Le espressioni Cron che indicano frequenze più rapide di un minuto non sono supportate.
- Non è possibile impostare un canary in modo da attendere più di un anno prima dell'esecuzione, quindi è possibile specificare solo * nel campo `Year`.

Esempi

Quando crei un canary puoi fare riferimento alle seguenti stringhe Cron di esempio. Gli esempi seguenti sono la sintassi corretta per utilizzare gli AWS CLI o AWS SDK per creare o aggiornare un canarino. Se stai usando la CloudWatch console, ometti la finale * in ogni esempio.

Expression	Significato
`0 10 * * ? *`	Esegui ogni giorno alle 10:00 (UTC)
`15 12 * * ? *`	Esegui ogni giorno alle 12:15 (UTC)
`0 18 ? * MON-FRI *`	Esegui dal lunedì al venerdì alle 18:00 (UTC)
`0 8 1 * ? *`	Esegui ogni primo giorno del mese alle 8:00 (UTC)
`0/10 * ? * MON-SAT *`	Esegui dal lunedì al sabato ogni 10 minuti
`0/5 8-17 ? * MON-FRI *`	Esegui dal lunedì al venerdì dalle 8.00 alle 17:55 (UTC) ogni 5 minuti

Gruppi

Puoi creare gruppi per associare i canary tra loro, compresi i canary tra regioni. L'utilizzo dei gruppi può aiutarti a gestire e automatizzare i canary e puoi anche visualizzare i risultati e le statistiche di esecuzione aggregati per tutti i canary di un gruppo.

I gruppi sono risorse globali. Quando crei un gruppo, questo viene replicato in tutte le AWS regioni che supportano i gruppi e puoi aggiungere canarini da una qualsiasi di queste regioni e visualizzarlo in una di queste regioni. Sebbene il formato dell'ARN del gruppo rifletta il nome della regione in cui è stato creato, un gruppo non è vincolato ad alcuna regione. Ciò significa che puoi inserire canary di più regioni nello stesso gruppo e utilizzare quindi tale gruppo per visualizzare e gestire tutti i canary in un'unica vista.

I gruppi sono supportati in tutte le regioni ad eccezione delle regioni che sono disattivate per impostazione predefinita. Per ulteriori informazioni su queste regioni, consulta Abilitazione di una regione.

Ogni gruppo può contenere un massimo di 10 canary. Nel tuo account puoi avere un massimo di 20 gruppi. Ogni singolo canary può essere membro al massimo di 10 gruppi.

Creazione di un gruppo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Synthetics Canaries.
3. Selezionare Create Group (Crea gruppo).
4. Per Group Name (Nome gruppo), specifica un nome per il gruppo.
5. Seleziona i canary da associare a questo gruppo. Per selezionare un canary, digita il suo nome completo in Exact canary name (Nome esatto del canary) e scegli Search (Cerca). Quindi seleziona la casella di controllo accanto al nome del canary. Se ci sono più canary con lo stesso nome in regioni diverse, assicurati di selezionare i canary desiderati.

   Puoi ripetere questa fase per associare un massimo di 10 canary al gruppo.
6. (Facoltativo) In Tags (Tag), aggiungi facoltativamente una o più coppie chiave/valore come tag per questo gruppo. I tag possono aiutarti a identificare e organizzare le tue AWS risorse e tenere traccia dei costi. AWS Per ulteriori informazioni, consulta Taggare le tue risorse Amazon CloudWatch .
7. Selezionare Create Group (Crea gruppo).

Prova un canarino a livello locale

Questa sezione spiega come modificare, testare ed eseguire il debug di CloudWatch Synthetics canaries direttamente all'interno dell'editor di codice o Microsoft Visual Studio dell'editor di codice. JetBrains IDE L'ambiente di debug locale utilizza un contenitore Serverless Application Model (SAM) per simulare una funzione Lambda per emulare il comportamento di un canarino Synthetics.

> **Note**
>
> Non è pratico eseguire localmente il debug dei canarini che si basano sul monitoraggio visivo. Il monitoraggio visivo si basa sull'acquisizione di schermate di base durante un'esecuzione iniziale e sul successivo confronto di queste schermate con le schermate delle esecuzioni successive. In un ambiente di sviluppo locale, le esecuzioni non vengono archiviate o tracciate e ogni iterazione è un'esecuzione indipendente e autonoma. L'assenza di una cronologia di esecuzione dei canarini rende poco pratico il debug dei canarini che si basano sul monitoraggio visivo.

Prerequisiti

1. Scegli o crea un bucket Amazon S3 da utilizzare per archiviare gli artefatti dei test locali di Canary, come file HAR e schermate. Ciò richiede che tu sia dotato di IAM. Se salti la configurazione dei bucket Amazon S3, puoi comunque testare il tuo canarino localmente, ma vedrai un messaggio di errore relativo al bucket mancante e non avrai accesso agli artefatti del canarino.

   Se utilizzi un bucket Amazon S3, ti consigliamo di impostare il ciclo di vita del bucket in modo da eliminare gli oggetti dopo alcuni giorni, per risparmiare sui costi. Per ulteriori informazioni, consulta Gestione del ciclo di vita dell'archiviazione.
2. Configura un profilo predefinito AWS per il tuo account. AWS Per ulteriori informazioni, consulta Impostazioni dei file di configurazione e credenziali.
3. Imposta la AWS regione predefinita dell'ambiente di debug sulla tua regione preferita, ad esempio. `us-west-2`
4. Installa la AWS SAM CLI. Per ulteriori informazioni, consulta Installazione della AWS SAM CLI.
5. Installa Visual Studio Code Editor o. JetBrains IDE Per ulteriori informazioni, consulta Visual Studio Code o JetBrains IDE
6. DockerInstalla per funzionare con la AWS SAM CLI. Assicurati di avviare il demone docker. Per ulteriori informazioni, consulta Installazione Docker da utilizzare con la AWS SAM CLI.

   In alternativa, puoi installare altri software di gestione dei containerRancher, ad esempio, purché utilizzi il Docker runtime.
7. Installa un'estensione del AWS toolkit per il tuo editor preferito. Per ulteriori informazioni, vedere Installazione di AWS Toolkit for Visual Studio Code o Installazione di AWS Toolkit for JetBrains.

Argomenti

Configurare l'ambiente di test e debug

Innanzitutto, clona il repository Github che lo fornisce inserendo il seguente comando. AWS Il repository contiene esempi di codice sia per i canari Node.js che per i canari Python.

```
git clone https://github.com/aws-samples/synthetics-canary-local-debugging-sample.git
```

Quindi esegui una delle seguenti operazioni, a seconda della lingua dei tuoi canarini.

Per i canarini Node.js

1. Vai alla directory dei sorgenti di Node.js canary inserendo il seguente comando.

   ```
   cd synthetics-canary-local-debugging-sample/nodejs-canary/src
   ```

2. Immettere il seguente comando per installare le dipendenze canarie.

   ```
   npm install
   ```

Per i canarini in Python

1. Vai alla directory dei sorgenti di Python canary inserendo il seguente comando.

   ```
   cd synthetics-canary-local-debugging-sample/python-canary/src
   ```

2. Immettere il seguente comando per installare le dipendenze canarie.

   ```
   pip3 install -r requirements.txt -t .
   ```

Utilizzare Visual Studio Code IDE

Il file di configurazione Visual Studio di avvio si trova in. `.vscode/launch.json` Contiene configurazioni per consentire al file modello di essere scoperto dal isual Studio codice V. Definisce un payload Lambda con i parametri richiesti per richiamare correttamente il canarino. Ecco la configurazione di avvio per un canarino Node.js:

```
{
            ...
            ...
```

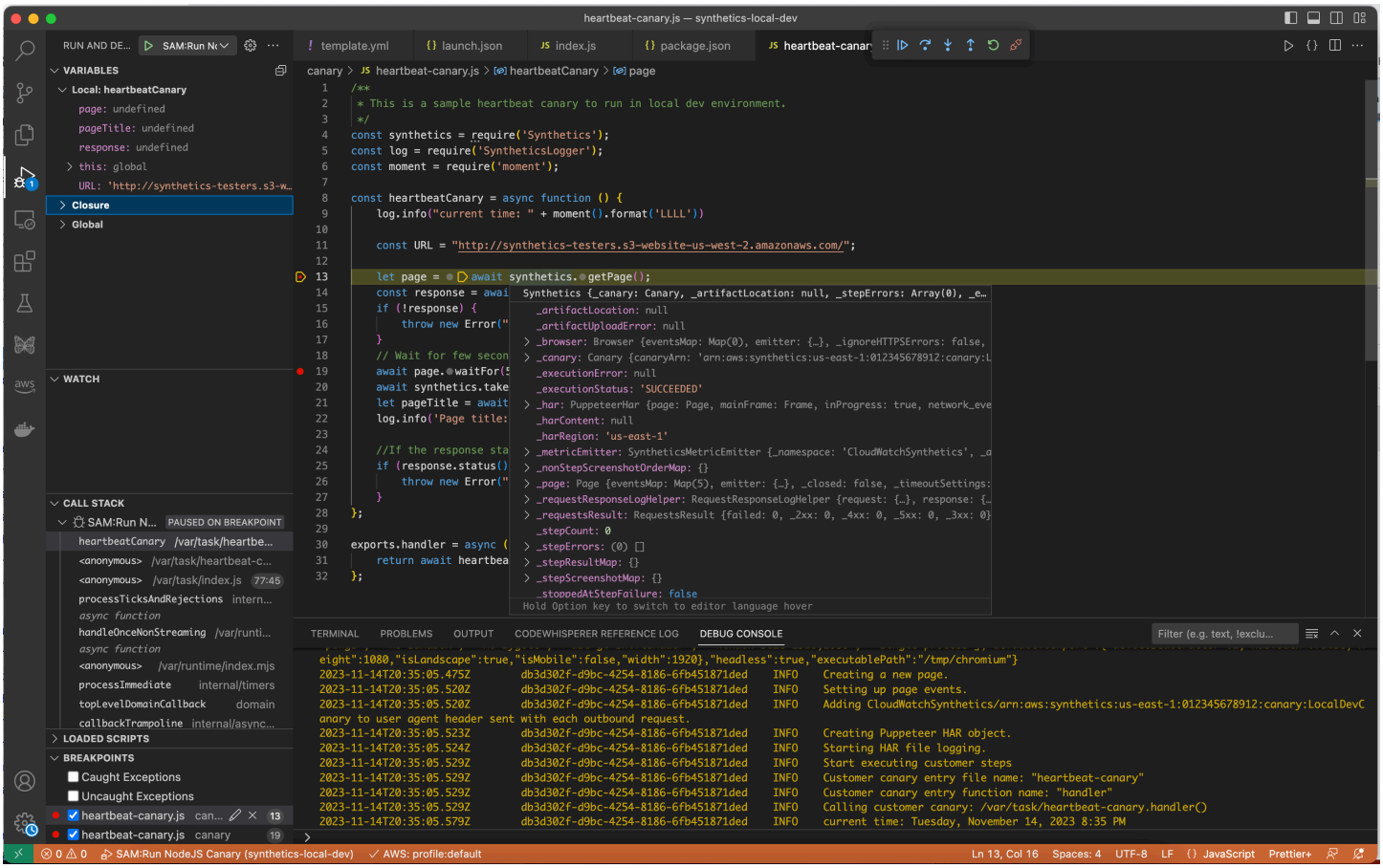
```
            "lambda": {
                "payload": {
                    "json": {
                        // Canary name. Provide any name you like.
                        "canaryName": "LocalSyntheticsCanary",
                        // Canary artifact location
                        "artifactS3Location": {
                            "s3Bucket": "cw-syn-results-123456789012-us-west-2",
                            "s3Key": "local-run-artifacts",
                        },
                        // Your canary handler name
                        "customerCanaryHandlerName": "heartbeat-canary.handler"
                    }
                },
                // Environment variables to pass to the canary code
                "environmentVariables": {}
            }
        }
    ]
}
```

Puoi anche fornire facoltativamente i seguenti campi nel payload JSON:

- `s3EncryptionMode`Valori validi: | `SSE_S3` `SSE_KMS`
- `s3KmsKeyArn`Valore valido: *`KMS Key ARN`*
- `activeTracing`Valori validi: | `true` `false`
- `canaryRunId`Valore valido: *`UUID`* Questo parametro è obbligatorio se la traccia attiva è abilitata.

Per eseguire il debug di Canary inVisual Studio, aggiungi dei punti di interruzione nel codice Canary in cui desideri sospendere l'esecuzione. Per aggiungere un punto di interruzione, scegliete il margine dell'editor e passate alle modalità Esegui e Debug nell'editor. Esegui il canarino facendo clic sul pulsante play. Quando il canarino è in funzione, i log verranno registrati nella console di debug, che ti fornirà informazioni in tempo reale sul comportamento del canarino. Se hai aggiunto punti di interruzione, l'esecuzione di Canary si interromperà in corrispondenza di ogni punto di interruzione, consentendoti di esaminare il codice e controllare i valori delle variabili, i metodi di istanza, gli attributi degli oggetti e lo stack di chiamate alle funzioni.

Non sono previsti costi per l'esecuzione e il debug di Canaries localmente, ad eccezione degli artefatti archiviati nel bucket Amazon S3 e delle metriche generate da ogni esecuzione locale. CloudWatch



Utilizzare JetBrains IDE

Dopo aver installato l' AWS Toolkit for JetBrains estensione, assicuratevi che il plugin e il JavaScript debugger Node.js siano abilitati all'esecuzione, se state eseguendo il debug di un canarino Node.js. Quindi procedi come descritto di seguito.

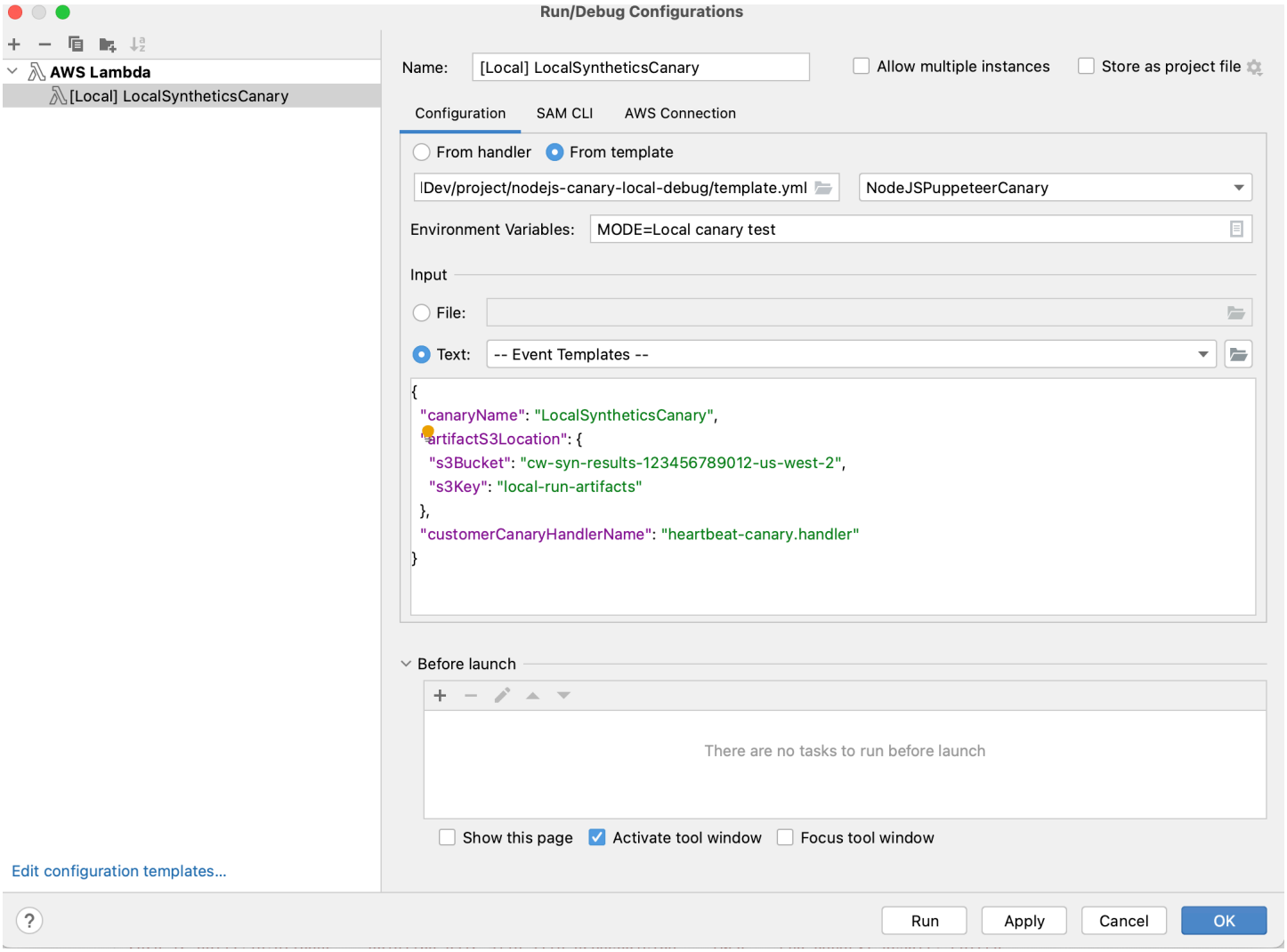
Esegui il debug di un canarino usando JetBrains IDE

1. Nel riquadro di navigazione a sinistra diJetBrains IDE, scegli Lambda, quindi scegli il modello di configurazione locale.
2. Inserisci un nome per la configurazione di esecuzione, ad esempio **`LocalSyntheticsCanary`**
3. Scegli Da template, scegli il file browser nel campo template, quindi seleziona il file template.yml dal progetto, dalla directory nodejs o dalla directory python.
4. Nella sezione Input, inserisci il payload per il canarino come mostrato nella schermata seguente.

```
{
 "canaryName": "LocalSyntheticsCanary",
 "artifactS3Location": {
```

```
    "s3Bucket": "cw-syn-results-123456789012-us-west-2",
    "s3Key": "local-run-artifacts"
 },
 "customerCanaryHandlerName": "heartbeat-canary.handler"
}
```

Puoi anche impostare altre variabili di ambiente nel payload JSON, come elencato in. Utilizzare Visual Studio Code IDE



Esegui un canary localmente con la CLI SAM

Utilizza una delle seguenti procedure per eseguire il tuo canary localmente utilizzando la CLI Serverless Application Model (SAM). Assicurati di specificare il nome del tuo bucket Amazon S3 per in `s3Bucket` `event.json`

Per utilizzare la CLI SAM per eseguire un canary Node.js

1. Vai alla directory dei sorgenti inserendo il seguente comando.

```
cd synthetics-canary-local-debugging-sample/nodejs-canary
```

2. Esegui i comandi seguenti:

```
sam build
sam local invoke -e ../event.json
```

Usare la CLI SAM per eseguire un canary Python

1. Vai alla directory dei sorgenti inserendo il seguente comando.

```
cd synthetics-canary-local-debugging-sample/python-canary
```

2. Esegui i comandi seguenti:

```
sam build
sam local invoke -e ../event.json
```

Integra il tuo ambiente di test locale in un pacchetto canary esistente

Puoi integrare il debug locale di Canary nel tuo pacchetto canary esistente copiando tre file:

- Copia il `template.yml` file nella radice del tuo pacchetto canary. Assicurati di modificare il percorso in modo che punti `CodeUri` alla directory in cui esiste il tuo codice canarino.
- Se state lavorando con un canarino Node.js, copiate il `cw-synthetics.js` file nella directory dei sorgenti di Canary. Se stai lavorando con un canary Python, copialo nella tua directory dei sorgenti `cw-synthetics.py` di Canary.
- Copia il file di configurazione di avvio. `vscode/launch.json`nella radice del pacchetto. Assicurati di metterlo all'interno della `.vscode` cartella; crealo se non esiste già.

Modificare il runtime di CloudWatch Synthetics

Come parte del debug, potresti provare a eseguire un canary con un runtime Synthetics diverso, invece del runtime CloudWatch più recente. A tale scopo, trovate il runtime che desiderate utilizzare da una delle seguenti tabelle. Assicurati di selezionare il runtime per la regione corretta. Quindi incolla l'ARN per quel runtime nella posizione appropriata del `template.yml` file, quindi esegui il canary.

Runtime Node.js

ARN per -7.0 syn-nodejs-puppeteer

La tabella seguente elenca gli ARN da utilizzare per la versione `syn-nodejs-puppeteer-7.0` del runtime CloudWatch Synthetics in AWS ciascuna regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:44`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:46`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:44`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:47`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:44`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:45`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:20`

Regione	ARN
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:26
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:18
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:44
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:30
Asia Pacific (Seul)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:46
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:49
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:44
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:44
Canada (Centrale)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:44
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:76
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:45
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:46
Europa (Francoforte)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:44

Regione	ARN
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:46`
Europa (Londra)	`arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:44`
Europa (Milano)	`arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:45`
Europa (Parigi)	`arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:44`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:20`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:44`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:19`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:17`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:44`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:19`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:45`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:41`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:42`

ARN per -6.2 syn-nodejs-puppeteer

La tabella seguente elenca gli ARN da utilizzare per la versione `syn-nodejs-puppeteer-6.2` del runtime CloudWatch Synthetics in AWS ciascuna regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:41`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:43`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:41`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:44`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:41`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:42`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:41`

Regione	ARN
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27
Asia Pacific (Seul)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:42
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:46
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:41
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:41
Canada (Centrale)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:41
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:42
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:43
Europa (Francoforte)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:41
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:43
Europa (Londra)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:41
Europa (Milano)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:42

Regione	ARN
Europa (Parigi)	`arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:41`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:41`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:41`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:42`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:39`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:39`

ARN per -5.2 syn-nodejs-puppeteer

La tabella seguente elenca gli ARN da utilizzare per la versione `syn-nodejs-puppeteer-5.2` del runtime CloudWatch Synthetics in AWS ciascuna regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:42`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:44`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:42`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:45`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:42`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:43`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:18`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:24`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:16`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:42`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:28`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:44`

Regione	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:47
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:42
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:42
Canada (Centrale)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:42
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:74
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:43
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:44
Europa (Francoforte)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:42
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:44
Europa (Londra)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:42
Europa (Milano)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:43
Europa (Parigi)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:42
Europa (Spagna)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:18

Regione	ARN
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:42`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:17`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:15`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:42`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:17`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:43`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:40`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:40`

Runtime Python

ARN per -3.0 syn-python-selenium

La tabella seguente elenca gli ARN da utilizzare per la versione `syn-python-selenium-3.0` del runtime CloudWatch Synthetics in AWS ciascuna regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`aarn:aws:lambda:us-east-1:378653112637:layer:Synthetics_Selenium:32`

Regione	ARN
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics_Selenium:34`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics_Selenium:32`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics_Selenium:34`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics_Selenium:32`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics_Selenium:32`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics_Selenium:20`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics_Selenium:26`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics_Selenium:18`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics_Selenium:32`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics_Selenium:30`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics_Selenium:34`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics_Selenium:37`

Regione	ARN
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics_Selenium:32
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics_Selenium:32
Canada (Centrale)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics_Selenium:32
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics_Selenium:76
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics_Selenium:32
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics_Selenium:32
Europa (Francoforte)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics_Selenium:32
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics_Selenium:34
Europa (Londra)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics_Selenium:32
Europa (Milano)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics_Selenium:33
Europa (Parigi)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics_Selenium:32
Europa (Spagna)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics_Selenium:20
Europa (Stoccolma)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics_Selenium:32

Regione	ARN
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics_Selenium:19`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics_Selenium:17`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics_Selenium:32`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics_Selenium:19`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics_Selenium:33`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics_Selenium:30`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics_Selenium:31`

ARN per -2.1 syn-python-selenium

La tabella seguente elenca gli ARN da utilizzare per la versione `syn-python-selenium-2.1` del runtime CloudWatch Synthetics in AWS ciascuna regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:29`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:31`

Regione	ARN
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:29`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:31`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:29`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:29`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:29`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:30`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:34`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:29`

Regione	ARN
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:29
Canada (Centrale)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:29
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:29
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:29
Europa (Francoforte)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:29
Europa (Irlanda)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:31
Europa (Londra)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:29
Europa (Milano)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:30
Europa (Parigi)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:29
Europa (Spagna)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17
Europa (Stoccolma)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:29
Europa (Zurigo)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16

Regione	ARN
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:29`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:30`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:29`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:29`

Errori comuni

Errore: l'esecuzione locale di progetti AWS SAM richiede Docker. L'hai installato e funzionante?

Assicurati di avviarlo Docker sul tuo computer.

Richiamata locale SAM non riuscita: si è verificato un errore (ExpiredTokenException) durante la chiamata dell' GetLayerVersion operazione: il token di sicurezza incluso nella richiesta è scaduto

Assicurati che il profilo AWS predefinito sia impostato.

Errori più comuni

Per ulteriori informazioni sugli errori più comuni con SAM, consulta Risoluzione dei problemi della CLI AWS SAM.

Risoluzione dei problemi di un canary fallito

Se il canary fallisce, procedi come descritto di seguito per la risoluzione dei problemi.

Risoluzione dei problemi generali

- Utilizza la pagina dei dettagli del canary per trovare maggiori informazioni. Nella CloudWatch console, scegli Canarie nel pannello di navigazione, quindi scegli il nome del canarino per aprire la pagina dei dettagli del canarino. Nella scheda Disponibilità, controlla la SuccessPercentmetrica per vedere se il problema è costante o intermittente.

  Mentre ancora nella scheda Availability (Disponibilità), scegli un punto dati non riuscito per visualizzare schermate, log e report delle fasi (se disponibili) per l'esecuzione non riuscita.

  Se è disponibile un report di passaggio perché le fasi fanno parte dello script, verifica quale fase non è riuscita e consulta gli screenshot associati per verificare il problema riscontrato dai clienti.

  È inoltre possibile controllare i file HAR per vedere se una o più richieste non vanno a buon fine. Puoi esaminare ulteriori dettagli utilizzando i log per analizzare richieste ed errori non andati a buon fine. Infine, puoi confrontare questi artefatti con gli artefatti di un'esecuzione di un canary andato a buon fine per individuare il problema.

  Per impostazione predefinita, CloudWatch Synthetics acquisisce schermate per ogni passaggio in un'interfaccia utente canary. Tuttavia, lo script potrebbe essere configurato per disabilitare gli screensot. Durante il debug, potrai voler abilitare nuovamente gli screenshot. Allo stesso modo, per i canary dell'API potresti voler vedere le intestazioni e il corpo della richiesta HTTP e della risposta durante il debug. Per informazioni su come includere questi dati nel report, consulta executeHttpStep(StepName, RequestOptions, [callback], [StepConfig]).
- Se disponi di un'implementazione recente per l'applicazione, esegui il rollback e quindi esegui il debug in un secondo momento.
- Connettiti manualmente all'endpoint per verificare se è possibile riprodurre lo stesso problema.

Argomenti

- Canary fallisce dopo l'aggiornamento dell'ambiente Lambda
- Il mio canarino è bloccato da AWS WAF
- Attesa della visualizzazione di un elemento
- Il nodo non è visibile oppure non è un elemento HTML per page.click()
- Impossibile caricare artefatti su S3, Eccezione: impossibile recuperare la posizione del bucket S3: accesso negato
- Errore: errore di protocollo (Runtime). callFunctionOn): Obiettivo chiuso.

- Canary non riuscito. Errore: nessun datapoint - Canary mostra errore di timeout
- Tentativo di accedere a un endpoint interno
- Problemi relativi all'aggiornamento e al downgrade della versione di runtime del canary
- Problema CORS (Cross-Origin Request Sharing)
- Problemi relativi alle condizioni della gara delle Canarie
- Risoluzione dei problemi di un Canary su un VPC

Canary fallisce dopo l'aggiornamento dell'ambiente Lambda

CloudWatch I canarini Synthetics sono implementati come funzioni Lambda nel tuo account. Queste funzioni Lambda sono soggette a regolari aggiornamenti del runtime Lambda contenenti aggiornamenti di sicurezza, correzioni di bug e altri miglioramenti. Lambda si impegna a fornire aggiornamenti di runtime retrocompatibili con le funzioni esistenti. Tuttavia, come nel caso delle patch software, ci sono rari casi in cui un aggiornamento del runtime può influire negativamente su una funzione esistente. Se ritieni che il tuo canarino sia stato influenzato da un aggiornamento del runtime Lambda, puoi utilizzare la modalità manuale di gestione del runtime Lambda (nelle regioni supportate) per ripristinare temporaneamente la versione runtime Lambda. Ciò mantiene attiva la funzione Canary e riduce al minimo le interruzioni, lasciando il tempo necessario per porre rimedio all'incompatibilità prima di tornare all'ultima versione di runtime.

Se il tuo canary non funziona dopo un aggiornamento del runtime Lambda, la soluzione migliore è eseguire l'aggiornamento a uno dei runtime Synthetics più recenti. Per ulteriori informazioni sui runtime più recenti, consulta. Versioni di runtime Synthetics

Come soluzione alternativa, nelle regioni in cui sono disponibili i controlli di gestione del runtime Lambda, puoi ripristinare un canary a un vecchio runtime gestito da Lambda, utilizzando la modalità manuale per i controlli di gestione del runtime. Puoi impostare la modalità manuale utilizzando AWS CLI o utilizzando la console Lambda, seguendo i passaggi riportati di seguito nelle sezioni seguenti.

> **Warning**
>
> Quando modifichi le impostazioni di runtime in modalità manuale, la funzione Lambda non riceverà aggiornamenti di sicurezza automatici finché non tornerà alla modalità Auto. Durante questo periodo, la tua funzione Lambda potrebbe essere soggetta a vulnerabilità di sicurezza.

Prerequisiti

- Installa jq
- Installa la versione più recente di AWS CLI. Per ulteriori informazioni, consulta le istruzioni di AWS CLI installazione e aggiornamento.

Fase 1: Ottenere la funzione Lambda ARN

Esegui il comando seguente per recuperare il `EngineArn` campo dalla risposta. Questo `EngineArn` è l'ARN della funzione Lambda associata al canarino. Utilizzerai questo ARN nei seguenti passaggi.

```
aws synthetics get-canary --name my-canary | jq '.Canary.EngineArn'
```

Esempio di output di`EngingArn`:

```
"arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-dc5015c2-db17-4cb5-
afb1-EXAMPLE991:8"
```

Fase 2: Ottenere l'ultima versione di runtime Lambda valida (ARN)

Per capire se il tuo canarino è stato interessato da un aggiornamento del runtime Lambda, controlla se la data e l'ora in cui le modifiche ARN della versione runtime Lambda nei tuoi registri corrispondono alla data e all'ora in cui hai visto l'impatto sul tuo canarino. Se non corrispondono, probabilmente non è un aggiornamento del runtime Lambda a causare i problemi.

Se il tuo canary è interessato da un aggiornamento del runtime Lambda, devi identificare l'ARN della versione di runtime Lambda funzionante che stavi utilizzando in precedenza. Segui le istruzioni riportate in Identificazione delle modifiche alla versione di runtime per trovare l'ARN del runtime precedente. Registrare l'ARN della versione di runtime e continuare con il passaggio 3 per impostare la configurazione della gestione del runtime.

Se il tuo canary non è ancora stato interessato da un aggiornamento dell'ambiente Lambda, puoi trovare l'ARN della versione di runtime Lambda che stai utilizzando attualmente. Esegui il comando seguente per recuperare `RuntimeVersionArn` la funzione Lambda dalla risposta.

```
aws lambda get-function-configuration \
--function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991:8" | jq '.RuntimeVersionConfig.RuntimeVersionArn'
```

Esempio di output di: `RuntimeVersionArn`

```
"arn:aws:lambda:us-
west-2::runtime:EXAMPLE647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Fase 3: Aggiornamento della configurazione di gestione del runtime Lambda

È possibile utilizzare la console AWS CLI o la console Lambda per aggiornare la configurazione di gestione del runtime.

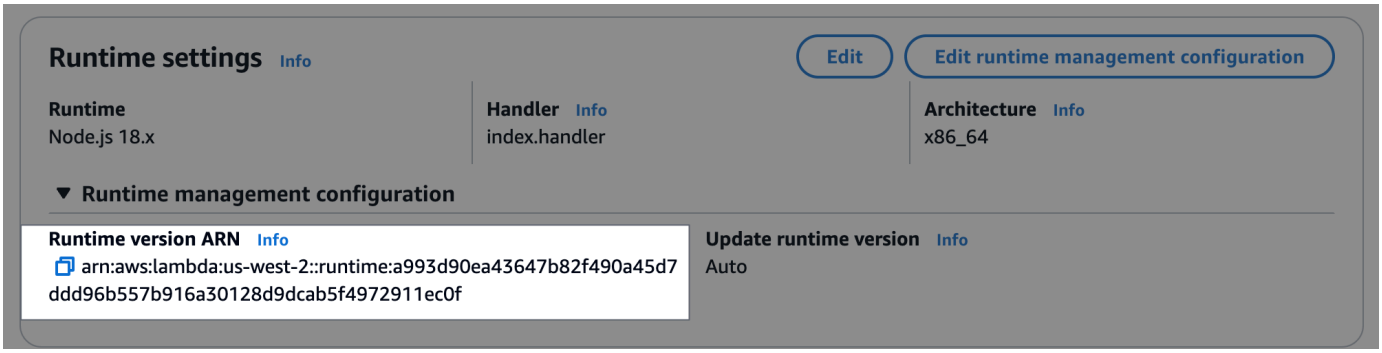
Per impostare la modalità manuale di configurazione della gestione del runtime Lambda utilizzando AWS CLI

Immettere il comando seguente per modificare la gestione del runtime della funzione Lambda in modalità manuale. Assicurati di sostituire il *nome e il qualificatore della funzione Lambda con* l'ARN della funzione Lambda e il numero di versione della funzione Lambda, rispettivamente, utilizzando i valori trovati nel passaggio 1. Sostituisci anche il *runtime-version-arn*con la versione ARN che hai trovato nel passaggio 2.

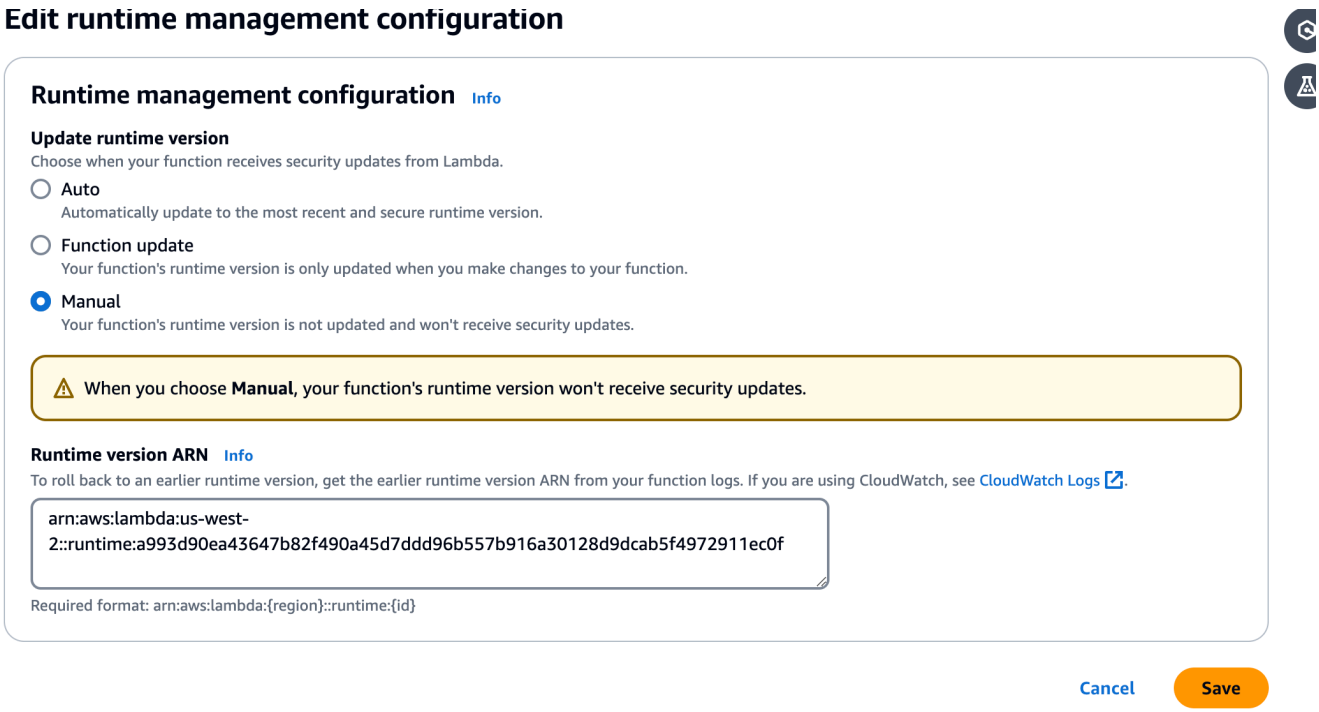
```
aws lambda put-runtime-management-config \
    --function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991" \
    --qualifier 8 \
    --update-runtime-on "Manual" \
    --runtime-version-arn "arn:aws:lambda:us-
west-2::runtime:a993d90ea43647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Per cambiare un canarino in modalità manuale utilizzando la console Lambda

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Scegli la scheda Versioni, scegli il link al numero di versione che corrisponde al tuo ARN e scegli la scheda Codice.
3. Scorri verso il basso fino alle impostazioni di Runtime, espandi la configurazione di gestione del Runtime e copia l'ARN della versione Runtime.



4. Scegli Modifica configurazione di gestione del runtime, scegli Manuale, incolla l'ARN della versione di runtime che hai copiato in precedenza nel campo ARN della versione di runtime. Quindi scegli Save (Salva).



Il mio canarino è bloccato da AWS WAF

Per evitare AWS WAF di bloccare il tuo canarino, imposta una condizione di corrispondenza delle stringhe che consenta la AWS WAF stringa. `CloudWatchSynthetics` Per ulteriori informazioni, consulta Lavorare con le condizioni di corrispondenza delle stringhe nella AWS WAF documentazione.

Attesa della visualizzazione di un elemento

Dopo aver analizzato i log e gli screenshot, se vedi che il tuo script è in attesa che un elemento venga visualizzato sullo schermo e viene raggiunto il timeout, controlla lo screenshot pertinente per vedere se l'elemento appare nella pagina. Verifica il tuo `xpath` per assicurarti che sia corretto.

Per problemi relativi a Puppeteer, consulta la pagina di Puppeteer o i forum su Internet. GitHub

Il nodo non è visibile oppure non è un elemento HTML per page.click()

Se un nodo non è visibile o non è un `HTMLElement` per `page.click()`, prima di tutto verifica l'`xpath` che stai utilizzando per fare clic sull'elemento. Inoltre, se il tuo elemento si trova nella parte inferiore dello schermo, modifica la visualizzazione. CloudWatch Synthetics per impostazione predefinita utilizza una finestra di 1920 * 1080. Puoi impostare un'area di visualizzazione diversa quando avvii il browser o utilizzando la funzione `page.setViewport` di Puppeteer.

Impossibile caricare artefatti su S3, Eccezione: impossibile recuperare la posizione del bucket S3: accesso negato

Se il tuo canary fallisce a causa di un errore di Amazon S3, CloudWatch Synthetics non è stata in grado di caricare schermate, log o report creati per il canarino a causa di problemi di autorizzazione. Verifica quanto segue:

- Controlla che il ruolo IAM del canary abbia autorizzazione `s3:ListAllMyBuckets`, l'autorizzazione `s3:GetBucketLocation` per il bucket Amazon S3 corretto e autorizzazione `s3:PutObject` per il bucket dove il canary immagazzina i suoi artefatti. Se il canary esegue il monitoraggio visivo, per il ruolo è necessaria anche l'autorizzazione `s3:GetObject` per il bucket. Queste stesse autorizzazioni sono richieste anche nella policy degli endpoint gateway Amazon VPC S3, se il canary viene distribuito in un VPC con un endpoint VPC.
- Se il canarino utilizza una chiave gestita AWS KMS dal cliente per la crittografia anziché la chiave AWS gestita standard (impostazione predefinita), il ruolo IAM del canarino potrebbe non avere l'autorizzazione per crittografare o decrittografare utilizzando tale chiave. Per ulteriori informazioni, consulta Crittografia di artefatti canary.
- La tua policy del bucket potrebbe non permettere il meccanismo di crittografia utilizzato da Canary. Ad esempio, se la policy del bucket richiede di utilizzare un meccanismo di crittografia specifico o una chiave KMS, è necessario selezionare la stessa modalità di crittografia per il canary.

Se il canary esegue il monitoraggio visivo, vedere Aggiornamento della posizione e della crittografia degli artifact quando si utilizza il monitoraggio visivo per ulteriori informazioni.

Errore: errore di protocollo (Runtime). callFunctionOn): Obiettivo chiuso.

Questo errore viene visualizzato se sono presenti alcune richieste di rete dopo la chiusura della pagina o del browser. Potresti aver dimenticato di attendere un'operazione asincrona. Dopo aver eseguito lo script, CloudWatch Synthetics chiude il browser. L'esecuzione di qualsiasi operazione asincrona dopo la chiusura del browser potrebbe causare un `target closed error`.

Canary non riuscito. Errore: nessun datapoint - Canary mostra errore di timeout

Significa che l'esecuzione del canary ha superato il timeout. L'esecuzione di Canary si è interrotta prima CloudWatch che Synthetics potesse pubblicare metriche sulla CloudWatch percentuale di successo o aggiornare artefatti come file HAR, registri e schermate. Se il timeout è troppo basso, puoi aumentarlo.

Per impostazione predefinita, un valore di timeout del canary è uguale alla sua frequenza. Puoi regolare manualmente il valore di timeout in modo che sia inferiore o uguale alla frequenza del canary. Se la frequenza del canary è bassa, è necessario aumentare la frequenza per aumentare il timeout. Puoi regolare sia la frequenza che il valore di timeout in Schedule quando crei o aggiorni un canarino utilizzando la console Synthetics CloudWatch .

Assicurati che il valore di timeout canary non sia inferiore a 15 secondi per consentire l'avvio a freddo Lambda e il tempo necessario per avviare la strumentazione canary.

Gli artefatti Canary non sono disponibili per la visualizzazione nella console Synthetics quando si verifica CloudWatch questo errore. Puoi usare CloudWatch Logs per vedere i registri del canarino.

Usare CloudWatch Logs per visualizzare i tronchi di un canarino

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione a sinistra, scegli Log groups (Gruppi di log).
3. Individua il gruppo di log digitando il nome del canary nella casella filtro. I gruppi di log per i canary hanno il nome /aws/lambda/cwsyn-***canaryName***-randomId.

Tentativo di accedere a un endpoint interno

Se desideri che il tuo canary acceda a un endpoint sulla tua rete interna, ti consigliamo di configurare CloudWatch Synthetics per utilizzare VPC. Per ulteriori informazioni, consulta Esecuzione di un Canary su un VPC.

Problemi relativi all'aggiornamento e al downgrade della versione di runtime del canary

Se di recente è stato aggiornato il canary dalla versione di runtime `syn-1.0` a una versione successiva, potrebbe trattarsi di un problema CORS (cross-origin resource sharing). Per ulteriori informazioni, consulta Problema CORS (Cross-Origin Request Sharing).

Se hai recentemente effettuato il downgrade di Canary a una versione di runtime precedente, assicurati che le funzioni Synthetics CloudWatch che stai utilizzando siano disponibili nella versione di runtime precedente a cui hai effettuato il downgrade. Ad esempio, la funzione `executeHttpStep` è disponibile per la versione di runtime `syn-nodejs-2.2` e versioni successive. Per verificare la disponibilità delle funzioni, consulta Scrivere uno script canary.

> **Note**
>
> Quando prevedi di aggiornare o effettuare il downgrade della versione runtime di un canarino, ti consigliamo di clonare prima il canarino e aggiornare la versione runtime nel canarino clonato. Una volta verificato che il clone con la nuova versione di runtime funziona, puoi aggiornare la versione di runtime del canary originale ed eliminare il clone.

Problema CORS (Cross-Origin Request Sharing)

In un canary dell'interfaccia utente, se alcune richieste di rete non vanno a buon fine con `403` o `net::ERR_FAILED`, verifica se il canary ha attivato il tracciamento attivo e utilizza anche la funzione `page.setExtraHTTPHeaders` di Puppeteer per aggiungere intestazioni. In tal caso, le richieste di rete non riuscite potrebbero essere causate da restrizioni CORS (Cross-Origin Request Sharing). Puoi confermare se questo è il caso disabilitando il tracciamento attivo o rimuovendo le intestazioni HTTP aggiuntive.

Perché succede?

Quando viene utilizzato il tracciamento attivo, viene aggiunta un'intestazione aggiuntiva a tutte le richieste in uscita per tracciare la chiamata. La modifica delle intestazioni della richiesta aggiungendo

un'intestazione trace o aggiungendo intestazioni aggiuntive utilizzando Puppeteer causa un controllo CORS per le richieste XML (XHR). `page.setExtraHTTPHeaders` HttpRequest

Se non desideri disabilitare il tracciamento attivo o rimuovere le intestazioni aggiuntive, puoi aggiornare l'applicazione Web per consentire l'accesso multiorigine oppure disabilitare la protezione Web utilizzando il comando `disable-web-security` quando avvii il browser Chrome nello script.

È possibile sovrascrivere i parametri di avvio utilizzati da CloudWatch Synthetics e passare parametri di flag `disable-web-security` aggiuntivi utilizzando la funzione di avvio Synthetics. CloudWatch Per ulteriori informazioni, consulta Funzioni di libreria disponibili per gli script canary Node.js.

> **Note**
>
> È possibile sovrascrivere i parametri di avvio utilizzati da CloudWatch Synthetics quando si utilizza la versione runtime o successiva. `syn-nodejs-2.1`

Problemi relativi alle condizioni della gara delle Canarie

Per una migliore esperienza di utilizzo di CloudWatch Synthetics, assicurati che il codice scritto per i canarini sia idempotente. Altrimenti, in rari casi, i canarini possono incontrare condizioni di gara quando il canarino interagisce con la stessa risorsa su piste diverse.

Risoluzione dei problemi di un Canary su un VPC

Se riscontri problemi dopo la creazione o l'aggiornamento di un Canary su un VPC, una delle sezioni seguenti potrebbe aiutarti a risolvere il problema.

Impossibile aggiornare il Canary o il nuovo Canary è in stato di errore

Se crei un Canary per l'esecuzione in un VPC e questo entra immediatamente in stato di errore oppure se non è possibile aggiornare un Canary per l'esecuzione su un VPC, è possibile che il ruolo del Canary non abbia le autorizzazioni corrette. Per l'esecuzione su un VPC, un canary deve disporre delle autorizzazioni `ec2:CreateNetworkInterface`, `ec2:DescribeNetworkInterfaces` e `ec2:DeleteNetworkInterface`. Queste autorizzazioni sono tutte contenute nella policy `AWSLambdaVPCAccessExecutionRole` gestita. Per ulteriori informazioni, consulta Autorizzazioni del ruolo di esecuzione e dell'utente.

Se il problema si verifica quando crei un Canary, devi eliminare il Canary e crearne uno nuovo. Se usi la CloudWatch console per creare il nuovo canarino, in Autorizzazioni di accesso, seleziona Crea un

nuovo ruolo. Viene creato un nuovo ruolo che include tutte le autorizzazioni necessarie per eseguire il canary.

Se il problema si verifica quando aggiorni un canary, puoi aggiornare nuovamente il canary e fornire un nuovo ruolo con le autorizzazioni necessarie.

Errore "Nessun risultato del test restituito"

Se un Canary visualizza l'errore "Nessun risultato del test restituito", la causa potrebbe una delle seguenti:

- Se il tuo VPC non dispone di accesso a Internet, devi utilizzare gli endpoint VPC per consentire a Canary di accedere ad Amazon S3. CloudWatch Devi abilitare le opzioni DNS resolution (Risoluzione DNS) e DNS hostname (Hostname DNS) nel VPC affinché questi indirizzi di endpoint vengano risolti correttamente. Per ulteriori informazioni, consulta Utilizzo del DNS con il VPC e Utilizzo CloudWatch e CloudWatch Synthetics con gli endpoint VPC di interfaccia.
- I Canary devono essere eseguiti in sottoreti private all'interno di un VPC. Per verificare questa condizione, apri la pagina Subnet (Sottorete) nella console VPC. Controlla le sottoreti selezionate durante la configurazione del Canary. Se hanno un percorso per un gateway Internet (igw-), non sono sottoreti private.

Per risolvere questi problemi, vedi i log per il Canary.

Per visualizzare gli eventi di log da un Canary

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch
2. Nel pannello di navigazione, seleziona Log groups (Gruppi di log).
3. Scegli il nome del gruppo di log del Canary. Il nome del gruppo di log inizia con `/aws/lambda/cwsyn-`*`canary-name`*.

Codice di esempio per gli script canary

Questa sezione contiene esempi di codice che illustrano alcune possibili funzioni per gli script canary di CloudWatch Synthetics.

Esempi per Node.js e Puppeteer

Impostazione dei cookie

I siti Web si basano sui cookie per fornire funzionalità personalizzate o tracciare gli utenti. Impostando i cookie negli script CloudWatch Synthetics, puoi imitare questo comportamento personalizzato e convalidarlo.

Ad esempio, un sito Web potrebbe visualizzare un collegamento Login (Accedi) per un utente che rivisita la pagina invece di un collegamento Register (Registrati).

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {

    let url = "http://smile.amazon.com/";

    let page = await synthetics.getPage();

    // Set cookies.  I found that name, value, and either url or domain are required
 fields.
    const cookies = [{
      'name': 'cookie1',
      'value': 'val1',
      'url': url
    },{
      'name': 'cookie2',
      'value': 'val2',
      'url': url
    },{
      'name': 'cookie3',
      'value': 'val3',
      'url': url
    }];

    await page.setCookie(...cookies);

    // Navigate to the url
    await synthetics.executeStep('pageLoaded_home', async function (timeoutInMillis =
 30000) {
```

```
        var response = await page.goto(url, {waitUntil: ['load', 'networkidle0'],
 timeout: timeoutInMillis});

        // Log cookies for this page and this url
        const cookiesSet = await page.cookies(url);
        log.info("Cookies for url: " + url + " are set to: " +
 JSON.stringify(cookiesSet));
    });

};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Emulazione del dispositivo

Puoi scrivere script che emulano vari dispositivi in modo da poter approssimare l'aspetto e il comportamento di una pagina su tali dispositivi.

L'esempio seguente emula un dispositivo iPhone 6. Per ulteriori informazioni sull'emulazione, consulta page.emulate (options) nella documentazione di Puppeteer.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');
const puppeteer = require('puppeteer-core');

const pageLoadBlueprint = async function () {

    const iPhone = puppeteer.devices['iPhone 6'];

    // INSERT URL here
    const URL = "https://amazon.com";

    let page = await synthetics.getPage();
    await page.emulate(iPhone);

    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
```

```
    }

    await page.waitFor(15000);

    await synthetics.takeScreenshot('loaded', 'loaded');

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Canary dell'API in più fasi

Questo codice di esempio illustra un canary dell'API con due fasi HTTP: testare la stessa API per casi di test positivi e negativi. La configurazione della fase viene passata per abilitare il reporting delle intestazioni della richiesta/risposta. Inoltre, nasconde l'intestazione di autorizzazione e X-Amz-Security-Token, perché contengono credenziali utente.

Quando questo script viene utilizzato come canary, puoi visualizzare i dettagli di ogni fase e le richieste HTTP associate, ad esempio pass/fail, durata e parametri delle prestazioni, come il tempo di ricerca DNS e il tempo del primo byte. Puoi visualizzare il numero di 2xx, 4xx e 5xx per l'esecuzione del canary.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const apiCanaryBlueprint = async function () {

    // Handle validation for positive scenario
    const validatePositiveCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            let responseBody = '';
            res.on('data', (d) => {
```

```
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };

    // Handle validation for negative scenario
    const validateNegativeCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 400) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            resolve();
        });
    };

    let requestOptionsStep1 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };

    let headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep1['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
    let stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
```

```
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };


    await synthetics.executeHttpStep('Verify GET products API with valid name',
 requestOptionsStep1, validatePositiveCase, stepConfig);

    let requestOptionsStep2 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/canary/InvalidName(',
        'port': 443,
        'protocol': 'https:'
    };

    headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep2['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
    stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };

    await synthetics.executeHttpStep('Verify GET products API with invalid name',
 requestOptionsStep2, validateNegativeCase, stepConfig);

};

exports.handler = async () => {
    return await apiCanaryBlueprint();
```

```
};
```

Esempi per Python e Selenium

Il seguente codice di esempio Selenium è un canary che fallisce con un messaggio di errore personalizzato quando un elemento di destinazione non viene caricato.

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver
from aws_synthetics.common import synthetics_logger as logger
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By

def custom_selenium_script():
    # create a browser instance
    browser = webdriver.Chrome()
    browser.get('https://www.example.com/')
    logger.info('navigated to home page')
    # set cookie
    browser.add_cookie({'name': 'foo', 'value': 'bar'})
    browser.get('https://www.example.com/')
    # save screenshot
    browser.save_screenshot('signed.png')
    # expected status of an element
    button_condition = EC.element_to_be_clickable((By.CSS_SELECTOR, '.submit-button'))
    # add custom error message on failure
    WebDriverWait(browser, 5).until(button_condition, message='Submit button failed to
 load').click()
    logger.info('Submit button loaded successfully')
    # browser will be quit automatically at the end of canary run,
    # quit action is not necessary in the canary script
    browser.quit()

# entry point for the canary
def handler(event, context):
    return custom_selenium_script()
```

Canary e tracciamento X-Ray

Puoi scegliere di abilitare il AWS X-Ray tracciamento attivo sui canarini che utilizzano il runtime o successivo. `syn-nodejs-2.0` Con il tracciamento abilitato, vengono inviate tracce per tutte le chiamate effettuate dal canarino che utilizzano il browser, l' AWS SDK o i moduli HTTP o HTTPS. I

canary con tracciamento abilitato vengono visualizzati sulla mappa di tracciamento X-Ray e all'interno di Application Signals una volta abilitata l'opzione per l'applicazione.

> **Note**
>
> L'attivazione del tracciamento X-Ray sui canary non è ancora supportata in Asia Pacifico (Giacarta).

Quando un canary viene visualizzato sulla mappa di tracciamento X-Ray, viene visualizzato come un nuovo tipo di nodo client. Puoi passare il mouse su un nodo canary per visualizzare i dati su latenza, richieste ed errori. Puoi anche scegliere il nodo canary per visualizzare più dati nella parte inferiore della pagina. Da quest'area della pagina, puoi scegliere Visualizza in Synthetics per passare alla console Synthetics per maggiori dettagli sul canarino, oppure scegli Visualizza tracce per vedere maggiori dettagli sulle tracce delle corse di questo canarino. CloudWatch

Un canary con tracciamento abilitato ha anche una scheda Tracing (Tracciamento) nella sua pagina dei dettagli, con dettagli sulle tracce e sui segmenti delle esecuzioni del canary.

L'abilitazione del tracciamento aumenta il tempo di esecuzione dei canary dal 2,5% al 7%.

Un canary con tracciamento abilitato deve utilizzare un ruolo con le seguenti autorizzazioni. Se usi la console per creare il ruolo quando crei il canary, queste autorizzazioni vengono concesse.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Sid230934",
            "Effect": "Allow",
            "Action": [
                "xray:PutTraceSegments"
            ],
            "Resource": "*"
        }
    ]
}
```

Vengono addebitati dei costi per le tracce generate dai canary. Per ulteriori informazioni sui prezzi di X-Ray, consulta Prezzi di AWS X-Ray.

Esecuzione di un Canary su un VPC

Puoi eseguire i canary sugli endpoint di un VPC e sugli endpoint interni pubblici. Per eseguire un Canary su un VPC, devi abilitare entrambe le opzioni DNS Resolution (Risoluzione DNS) e DNS hostnames (Nomi host DNS) sul VPC. Per ulteriori informazioni, vedi Utilizzo del DNS con VPC.

Quando esegui un canary su un endpoint VPC, devi fornire un modo per inviargli i parametri e gli artefatti CloudWatch ad Amazon S3. Se il VPC è già abilitato per l'accesso a Internet, non c'è altro da fare. Il canary verrà eseguito nel VPC, ma può accedere a Internet per caricare parametri e artefatti.

Se il VPC non è già abilitato per l'accesso a Internet, sono disponibili due opzioni:

- Attivarlo per l'accesso a Internet. Per ulteriori informazioni, consulta la sezione seguente Fornire accesso a Internet al tuo canary su un VPC.
- Se desideri mantenere privato il tuo VPC, puoi configurare Canary per inviare i dati ad Amazon CloudWatch S3 tramite endpoint VPC privati. Se non l'hai già fatto, devi creare un endpoint VPC per CloudWatch (com.amazonaws). *region* .monitoring) e un endpoint gateway per Amazon S3. Per ulteriori informazioni, consulta Utilizzo CloudWatch e CloudWatch Synthetics con endpoint VPC di interfaccia ed Endpoint Amazon VPC per Amazon S3.

Fornire accesso a Internet al tuo canary su un VPC

Segui questi passaggi per consentire l'accesso a Internet al tuo canarino VPC o per assegnare al tuo canarino un indirizzo IP statico

Per dare accesso a Internet a un canary su un VPC

1. Creare un gateway NAT in una sottorete pubblica VPC. Per istruzioni, consulta Creazione di un gateway NAT.
2. Aggiungere un nuovo routing alla tabella di routing nella sottorete privata in cui viene lanciato il canary. Specificare le impostazioni seguenti:

   - In Destination (Destinazione), immetti **`0.0.0.0/0`**.
   - Per Destinazione, scegli Gateway NAT e quindi scegli l'ID del gateway NAT creato.
   - Seleziona Save routes (Salva route).

   Per ulteriori informazioni sull'aggiunta di routing, alla tabella di routing, consulta Aggiunta e rimozione di route da una tabella di routing.

> **Note**
>
> Assicurati che i routing verso il gateway NAT siano in uno stato attivo. Se il gateway NAT viene eliminato e non hai aggiornato i routing, si trovano in uno stato blackhole. Per ulteriori informazioni consulta Utilizzo di gateway NAT.

Crittografia di artefatti canary

CloudWatch Synthetics archivia artefatti canarini come schermate, file HAR e report nel tuo bucket Amazon S3. Per impostazione predefinita, questi artefatti vengono crittografati quando sono inattivi utilizzando una chiave gestita. AWS Per ulteriori informazioni, consulta Customer keys and AWS keys.

Puoi scegliere di utilizzare un'opzione di crittografia diversa. CloudWatch Synthetics supporta quanto segue:

- SSE-S3– Crittografia lato server (SSE) con una chiave gestita da Amazon S3.
- SSE-KMS– Crittografia lato server (SSE) con una chiave AWS KMS gestita dal cliente.

Se desideri utilizzare l'opzione di crittografia predefinita con una chiave AWS gestita, non hai bisogno di autorizzazioni aggiuntive.

Per utilizzare la crittografia SSE-S3, devi specificare SSE_S3come modalità di crittografia quando si crea o si aggiorna il canary. Per utilizzare questa modalità di crittografia non sono necessarie autorizzazioni supplementari. Per ulteriori informazioni, consulta Protezione dei dati mediante la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3).

Per utilizzare una chiave gestita AWS KMS dal cliente, devi specificare SSE-KMS come modalità di crittografia quando crei o aggiorni il tuo canary e fornisci anche l'Amazon Resource Name (ARN) della tua chiave. È inoltre possibile utilizzare una chiave KMS a più account.

Per utilizzare una chiave gestita dal cliente, sono necessarie le seguenti impostazioni:

- Il ruolo IAM per il canary deve avere l'autorizzazione di crittografare gli artefatti utilizzando la chiave. Se utilizzi il monitoraggio visivo, è inoltre necessario concedergli l'autorizzazione a decrittare gli artefatti.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": {"Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "Your KMS key ARN"
    }
}
```

- Invece di aggiungere autorizzazioni al ruolo IAM, è possibile aggiungere il proprio ruolo IAM alla policy chiave. Se si usa lo stesso ruolo per più canary, si dovrebbe considerare questo approccio.

```
{
    "Sid": "Enable IAM User Permissions",
    "Effect": "Allow",
    "Principal": {
        "AWS": "Your synthetics IAM role ARN"
    },
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
    ],
    "Resource": "*"
}
```

- Se utilizzi una chiave KMS tra account, consulta Permettere agli utenti in altri account di utilizzare una chiave KMS.

Visualizzazione di artefatti canary crittografati quando si utilizza una chiave gestita dal cliente

Per visualizzare gli artefatti di Canary, aggiorna la tua chiave gestita dal cliente per autorizzare la decrittografia all'utente che visualizza AWS KMS gli artefatti. In alternativa, aggiungi le autorizzazioni di decrittografia all'utente o al ruolo IAM che sta visualizzando gli artefatti.

La AWS KMS policy predefinita abilita le policy IAM nell'account per consentire l'accesso alle chiavi KMS. Se utilizzi una chiave KMS per più account, consulta Perché gli utenti con più account ricevono errori di accesso negato quando tentano di accedere a oggetti Amazon S3 crittografati da una chiave personalizzata? AWS KMS .

Per ulteriori informazioni sulla risoluzione dei problemi di accesso negato a causa di una chiave KMS, consulta Risoluzione dei problemi di accesso alla chiave.

Aggiornamento della posizione e della crittografia degli artifact quando si utilizza il monitoraggio visivo

Per eseguire il monitoraggio visivo, CloudWatch Synthetics confronta gli screenshot con gli screenshot di base acquisiti durante l'esecuzione selezionata come linea di base. Se si aggiorna la posizione dell'artifact o l'opzione di crittografia, è necessario procedere in uno dei seguenti modi:

- Assicurarti che il ruolo IAM disponga di autorizzazioni sufficienti sia per la posizione precedente di Amazon S3 che per la nuova posizione Amazon S3 per gli artefatti. Assicurarti inoltre che disponga dell'autorizzazione per i metodi di crittografia precedenti e nuovi e le chiavi KMS.
- Creare una nuova linea di base selezionando la prossima canary run come nuova baseline. Se si utilizza questa opzione, è solo necessario assicurarti che il ruolo IAM disponga di autorizzazioni sufficienti per la nuova posizione degli artifact e l'opzione di crittografia.

Consigliamo la seconda opzione per selezionare l'esecuzione successiva come nuova baseline. Ciò evita di avere una dipendenza da una posizione di artefatto o da un'opzione di crittografia che non stai più usando per il canary.

Ad esempio, supponiamo che il canary utilizzi la posizione dell'artefatto A e la chiave KMS K per caricare gli artefatti. Se si aggiorna il canary alla posizione dell'artifact B e alla chiave KMS L, ci si può assicurare che il ruolo IAM disponga delle autorizzazioni per entrambe le posizioni degli artifact (A e B) e entrambe le chiavi KMS (K e L). In alternativa, è possibile selezionare l'esecuzione successiva come nuova baseline e assicurarti che il ruolo IAM canary disponga delle autorizzazioni per la posizione dell'artifact B e la chiave KMS L.

Visualizzazione delle statistiche e dei dettagli dei Canary

Puoi visualizzare i dettagli relativi ai Canary e le statistiche delle rispettive esecuzioni.

Per poter visualizzare tutti i dettagli relativi ai risultati della sessione di test canary, devi accedere a un account che dispone di autorizzazioni sufficienti. Per ulteriori informazioni, consulta Ruoli e autorizzazioni richiesti per i canarini CloudWatch.

Per visualizzare statistiche e dettagli dei Canary

1. CloudWatch Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Synthetics Canaries.

   Nei dettagli sui canary che hai creato:

- Status (Stato) mostra visivamente il numero di canary che hanno superato le sessioni di test più recenti.
- Groups (Gruppi) mostra i gruppi che hai creato e riporta quanti di essi hanno canary non funzionanti o allarmanti.
- Slowest performers (Eseguito con i risultati più lenti) visualizza il gruppo e la regione con i canary con le prestazioni più lente. Questi vengono calcolati sommando la durata media di tutti i canary (nell'arco di tempo selezionato) all'interno di un gruppo o di una regione e dividendola per il numero di canary nel gruppo o nella regione. Se scegli il parametro per il gruppo più lento, la tabella viene filtrata per visualizzare solo i gruppi più lenti e i relativi canary. La tabella è ordinata per Durata media.
- Nella parte inferiore della pagina è presente una tabella che visualizza tutti i canary. In una colonna vengono visualizzati gli allarmi creati per ogni canary. Vengono visualizzati solo gli allarmi conformi allo standard di denominazione per gli allarmi del canary. Questo standard è`Synthetics-Alarm-`*`canaryName`*`-`*`index`* . Gli allarmi Canary creati nella sezione Synthetics della CloudWatch console utilizzano automaticamente questa convenzione di denominazione. Se crei gli allarmi Canary nella sezione Allarmi della CloudWatch console o utilizzando AWS CloudFormation, e non usi questa convenzione di denominazione, gli allarmi funzionano ma non compaiono in questo elenco.

3. Per visualizzare più dettagli relativi un singolo canary, seleziona il nome del canary nella tabella Canaries (Canary).

   Nei dettagli su quel canary:

   - La scheda Availability (Disponibilità) mostra informazioni sulle esecuzioni recenti di questo canary.

     Sotto Canary, puoi scegli una delle linee per visualizzare i dettagli su quella esecuzione.

     Nel grafico, per visualizzare i dettagli puoi scegliere Steps (Fasi), Screenshot, Logs (Log) o HAR file (File HAR). Se il canary ha il tracciamento attivo abilitato, puoi anche scegliere Traces (Tracce) per vedere le informazioni di tracciamento delle esecuzioni del canary.

     I log relativi a Canary Running sono archiviati nei bucket S3 e nei Logs. CloudWatch

     Gli screenshot mostrano come i tuoi clienti visualizzano le tue pagine Web. Puoi utilizzare i file HAR (file di archivio HTTP) per visualizzare dati dettagliati sulle prestazioni delle pagine Web. Puoi analizzare l'elenco delle richieste Web e rilevare problemi di prestazioni, ad esempio

il tempo di caricamento di un elemento. I file di log mostrano il record delle interazioni tra il canary e la pagina Web e possono essere utilizzati per identificare i dettagli degli errori.

Se il canary usa la versione di runtime `syn-nodejs-2.0-beta` o versione successiva, puoi ordinare i file HAR in base al codice di stato, alla dimensione della richiesta o alla durata.

La scheda Steps (Fasi) visualizza un elenco delle fasi del canary, lo stato di ogni fase, il motivo dell'errore, l'URL dopo l'esecuzione della fase, gli screenshot e la durata dell'esecuzione della fase. Per i canary delle API con fasi HTTP, puoi visualizzare le fasi e le richieste HTTP corrispondenti se utilizzi il runtime `syn-nodejs-2.2` o versione successiva.

Seleziona HTTP Requests (Richieste HTTP) per visualizzare il log di ogni richiesta HTTP fatta dal canary. Puoi visualizzare le intestazioni della richiesta/risposta, il corpo della risposta, il codice di stato, gli intervalli di errore e le prestazioni (durata totale, tempo di connessione TCP, tempo di handshake TLS, tempo del primo byte e tempo di trasferimento del contenuto). Tutte le richieste HTTP che utilizzano il modulo HTTP/HTTPS dietro le quinte vengono acquisite qui.

Per impostazione predefinita nei canary API, l'intestazione della richiesta, l'intestazione della risposta, il corpo della richiesta e il corpo della risposta non sono inclusi nel report per motivi di sicurezza. Se scegli di includerli, i dati vengono archiviati solo nel bucket S3. Per informazioni su come includere questi dati nel report, consulta executeHttpStep(StepName, RequestOptions, [callback], [StepConfig]).

Sono supportati i tipi di contenuto del corpo della risposta testuale, HTML e JSON. Sono supportati tipi di contenuto come text/HTML, text/plain, Application/JSON e application/ -1.0. x-amz-json Le risposte compresse non sono supportate.

- La scheda Monitoraggio mostra i grafici delle metriche pubblicate da questo canarino. CloudWatch Per ulteriori informazioni su questi parametri, consulta CloudWatch metriche pubblicate da canaries.

  Sotto i CloudWatch grafici pubblicati dal canarino ci sono i grafici delle metriche Lambda relative al codice Lambda del canarino.
- La scheda Configuration (Configurazione) visualizza le informazioni di configurazione e pianificazione sul canary.
- La scheda Groups (Gruppi) mostra gli eventuali gruppi a cui è associato questo canary.
- La scheda Tags (Tag) visualizza i tag associati al canary.

CloudWatch metriche pubblicate da canaries

Canaries pubblica le seguenti metriche nel namespace. CloudWatch `CloudWatchSynthetics` Per ulteriori informazioni sulla visualizzazione CloudWatch delle metriche, consulta. Visualizzazione di parametri disponibili

Parametro	Descrizione
`SuccessPercent`	La percentuale delle esecuzioni di questo canary che riescono e non trovano fallimenti. Dimensioni valide: CanaryName Statistiche valide: media Unità: percentuale
`Duration`	La durata dell'esecuzione del canary, in millisecondi. Dimensioni valide: CanaryName Statistiche valide: media Unità: millisecondi
`Errors`	Il numero di volte in cui il canarino non è riuscito a eseguire lo script completo. Dimensioni valide: CanaryName Statistiche valide: Sum
`2xx`	Numero di richieste di rete eseguite dal canary che ha restituito risposte OK, con codici di risposta compresi tra 200 e 299. Questo parametro viene segnalato per i canary dell'interfaccia utente che utilizzano la versione di runtime `syn-nodejs-2.0` o versione successiva e viene segnalato per i canary dell'API che utilizzano la versione di runtime `syn-nodejs-2.2` o versione successiva. Dimensioni valide: CanaryName

Parametro	Descrizione
	Statistiche valide: Sum Unità: numero
4xx	Numero di richieste di rete eseguite dal canary che ha restituito risposte Error (Errore), con codici di risposta compresi tra 400 e 499. Questo parametro viene segnalato per i canary dell'interfaccia utente che utilizzano la versione di runtime `syn-nodejs-2.0` o versione successiva e viene segnalato per i canary dell'API che utilizzano la versione di runtime `syn-nodejs-2.2` o versione successiva. Dimensioni valide: CanaryName Statistiche valide: Sum Unità: numero
5xx	Numero di richieste di rete eseguite dal canary che ha restituito risposte Fault (Guasto), con codici di risposta compresi tra 500 e 599. Questo parametro viene segnalato per i canary dell'interfaccia utente che utilizzano la versione di runtime `syn-nodejs-2.0` o versione successiva e viene segnalato per i canary dell'API che utilizzano la versione di runtime `syn-nodejs-2.2` o versione successiva. Dimensioni valide: CanaryName Statistiche valide: Sum Unità: numero

Parametro	Descrizione
`Failed`	Il numero di esecuzioni canary che non sono stati eseguiti. Questi fallimenti sono legati al canary stesso. Dimensioni valide: CanaryName Statistiche valide: Sum Unità: numero
`Failed requests`	Numero di richieste HTTP eseguite dal canary sul sito Web di destinazione non riuscite senza risposta. Dimensioni valide: CanaryName Statistiche valide: Sum Unità: numero
`VisualMonitoringSuccessPercent`	Percentuale di confronti visivi che hanno abbinato correttamente gli screenshot di riferimento durante un'esecuzione del canary. Dimensioni valide: CanaryName Statistiche valide: media Unità: percentuale
`VisualMonitoringTotalComparisons`	Il numero totale di confronti visivi verificatisi durante un'esecuzione del canary. Dimensioni valide: CanaryName Unità: numero

> **Note**
>
> Canary che usano il metodo `executeStep()` o `executeHttpStep()` dalla libreria Synthetics pubblicano anche i parametri `SuccessPercent` e `Duration` con le dimensioni `CanaryName` e `StepName` per ogni fase.

Modifica o eliminazione di un canary

Puoi modificare o eliminare un canary esistente.

Modifica di un canary

Quando si modifica un canary, anche se non si modifica la pianificazione, questa viene reimpostata in base a quando modifichi il canary. Ad esempio, se disponi di un canary che viene eseguito ogni ora e modifichi quel canary, il canary verrà eseguito immediatamente dopo il completamento della modifica e poi ogni ora dopo.

Per modificare o aggiornare un canary

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Synthetics Canaries.
3. Seleziona il pulsante accanto al nome del canary, quindi scegli Actions (Operazioni), Edit (Modifica).
4. (Facoltativo) Se questo canary esegue il monitoraggio visivo degli screenshot e desideri impostare l'esecuzione successiva del canary come riferimento, seleziona Set next run as new baseline (Imposta l'esecuzione successiva come nuovo riferimento).
5. (Facoltativo) Se questo canary esegue il monitoraggio visivo degli screenshot e desideri rimuovere uno screenshot dal monitoraggio visivo o desideri designare parti dello screenshot da ignorare durante i confronti visivi, in Visual Monitoring (Monitoraggio visivo) scegli Edit Baseline (Modifica riferimento).

   Viene visualizzato lo screenshot e puoi eseguire una delle seguenti operazioni:

   - Per rimuovere lo screenshot dall'utilizzo per il monitoraggio visivo, seleziona Remove screenshot from visual test baseline (Rimuovi screenshot dal riferimento del test visivo).

- Per designare parti dello screenshot da ignorare durante i confronti visivi, fai clic e trascina per disegnare le aree dello schermo da ignorare. Una volta eseguita questa operazione per tutte le aree che desideri ignorare durante i confronti, scegli Save (Salva).

6. Apporta tutte le altre modifiche desiderate al canary e scegli Save (Salva).

Eliminare un canary

Quando elimini un canary, puoi scegliere se eliminare anche altre risorse utilizzate e create dal canary. Quando elimini un canary, devi eliminare anche quanto segue:

- Le funzioni Lambda e i livelli utilizzati da questo canary. Il loro prefisso è `cwsyn-`*`MyCanaryName`*.
- CloudWatch allarmi creati per questo canarino. Gli allarmi hanno un nome che inizia con `Synthetics-Alarm-`*`MyCanaryName`*. Per ulteriori informazioni sull'eliminazione di allarmi, consulta Modifica o elimina un allarme CloudWatch .
- Oggetti e bucket Amazon S3, ad esempio la posizione dei risultati del Canary e la posizione dell'artefatto.
- I ruoli IAM creati per il Canary. Il loro nome è `role/service-role/CloudWatchSyntheticsRole-`*`MyCanaryName`*.
- Gruppi di CloudWatch log in Registri creati per il canarino. I nomi di questi gruppi di log sono i seguenti: `/aws/lambda/cwsyn-`*`MyCanaryName-randomId`*.

Prima di eliminare un canary, è possibile visualizzare i dettagli del canary e prendere nota di queste informazioni. In questo modo, è possibile eliminare le risorse corrette dopo aver eliminato il canary.

Per eliminare un canary

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Synthetics Canaries.
3. Se il canary è attualmente nello stato `RUNNING`, è necessario arrestarlo. Solo i canary nello stato `STOPPED`, `READY(NOT_STARTED)` o `ERROR` possono essere eliminati.

   Per arrestare il canary seleziona il pulsante accanto al nome del canary, quindi scegli Actions (Operazioni), Stop (Arresta).
4. Seleziona il pulsante accanto al nome del canary, quindi scegli Actions (Operazioni), Delete (Elimina).

5. Scegli se eliminare anche le altre risorse create e utilizzate dal canary. Ciò include la funzione e i livelli Lambda, il ruolo IAM di Canary e la Policy IAM.

   Per eliminare il ruolo IAM di Canary e la Policy IAM, è necessario disporre di autorizzazioni sufficienti. Per ulteriori informazioni, consulta AWS politiche gestite (predefinite) per CloudWatch Synthetics.
6. Inserisci **`Delete`** nella casella e scegli Delete (Elimina).
7. Elimina le altre risorse utilizzate e create per il canary, come indicato in precedenza in questa sezione.

Avvio, interruzione, eliminazione o aggiornamento del runtime di più canary

Puoi interrompere, avviare, eliminare o aggiornare il runtime di cinque canary con una sola operazione. L'aggiornamento del runtime per un canary avviene per l'ultimo runtime disponibile per il linguaggio e il framework utilizzati dal canary.

Se selezioni più canary e solo alcuni di essi si trovano in uno stato valido per l'operazione selezionata, questa viene eseguita solo sui canary in cui tale azione è valida. Ad esempio, se selezioni una combinazione di canary attualmente in esecuzione e altri no, e scegli di avviarli, l'operazione avrà esito positivo soltanto per i canary che non erano in esecuzione, mentre gli altri non saranno interessati.

Se nessuno dei canary selezionati è valido per un'azione, tale operazione non sarà disponibile nel menu.

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Synthetics Canaries.
3. Seleziona le caselle di controllo accanto ai canary da interrompere, avviare o eliminare.
4. Scegli Actions (Operazioni), quindi scegli Start (Avvia), Stop (Interrompi), Delete (Elimina) o Update Runtime (Aggiorna runtime).

Monitoraggio degli eventi delle Canarie con Amazon EventBridge

Le regole EventBridge degli eventi di Amazon possono avvisarti quando i canarini cambiano stato o completano le esecuzioni. EventBridgefornisce un near-real-time flusso di eventi di sistema che descrivono i cambiamenti nelle AWS risorse. CloudWatch Synthetics invia questi eventi con

EventBridge il massimo impegno. La consegna con il massimo impegno significa che CloudWatch Synthetics tenta di inviare tutti gli eventi EventBridge a, ma in alcuni rari casi un evento potrebbe non essere consegnato. EventBridge elabora tutti gli eventi ricevuti almeno una volta. Inoltre, i listener di eventi potrebbero non ricevere gli eventi nell'ordine in cui si sono verificati.

> **Note**
>
> Amazon EventBridge è un servizio di bus di eventi che puoi utilizzare per connettere le tue applicazioni con dati provenienti da una varietà di fonti. Per ulteriori informazioni, consulta What is Amazon EventBridge? nella Amazon EventBridge User Guide.

CloudWatch Synthetics emette un evento quando un canarino cambia stato o completa una corsa. È possibile creare una EventBridge regola che includa uno schema di eventi che corrisponda a tutti i tipi di eventi inviati da CloudWatch Synthetics o che corrisponda solo a tipi di eventi specifici. Quando un canarino attiva una regola, EventBridge richiama le azioni target definite nella regola. In questo modo è possibile inviare notifiche, acquisire informazioni sugli eventi, intraprendere azioni correttive in risposta a una modifica dello stato del canary o al completamento di un'esecuzione del canary. Ad esempio, è possibile creare regole per i seguenti casi d'uso:

- Indagare quando una corsa canary fallisce
- Indagare quando un canary è entrato nello stato `ERROR`
- Tracciamento del ciclo di vita di un canary
- Monitoraggio della riuscita o del fallimento dell'esecuzione del canary in un flusso di lavoro

Eventi di esempio tratti da CloudWatch Synthetics

Questa sezione elenca gli eventi di esempio di CloudWatch Synthetics. Per ulteriori informazioni sul formato degli eventi, vedete Eventi e modelli di eventi in. EventBridge

Modifica dello stato dei canary

In questo tipo di evento, i valori di `current-state` e `previous-state` possono essere i seguenti:

`CREATING` | `READY` | `STARTING` | `RUNNING` | `UPDATING` | `STOPPING` | `STOPPED` | `ERROR`

```
{
                "version": "0",
```

```
                "id": "8a99ca10-1e97-2302-2d64-316c5dedfd61",
                "detail-type": "Synthetics Canary Status Change",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:19:43Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "current-state": "STOPPED",
                                "previous-state": "UPDATING",
                                "source-location": "NULL",
                                "updated-on": 1612909161.767,
                                "changed-config": {
                                                "executionArn": {
                                                                "previous-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                },
                                                "vpcId": {
                                                                "current-value": "NULL"
                                                },
                                                "testCodeLayerVersionArn": {
                                                                "previous-
value": "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                }
                                },
                                "message": "Canary status has changed"
                }
}
```

Esecuzione riuscita del canary completata

```
{
```

```
                "version": "0",
                "id": "989EXAMPLE-f4a5-57a7-1a8f-d9cc768a1375",
                "detail-type": "Synthetics Canary TestRun Successful",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:01Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "989EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "c6c39152-8f4a-471c-9810-989EXAMPLE",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-28ddbe266797/2021/02/09/22/23-41-200",
                                "test-run-status": "PASSED",
                                "state-reason": "null",
                                "canary-run-timeline": {
                                                "started": 1612909421,
                                                "completed": 1612909441
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

Esecuzione non riuscita del canary completata

```
{
                "version": "0",
                "id": "2644b18f-3e67-5ebf-cdfd-bf9f91392f41",
                "detail-type": "Synthetics Canary TestRun Failure",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:27Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "af3e3a05-dc5a-4f5f-96d1-9989EXAMPLE",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "0df3823e-7e33-4da1-8194-
b04e4d4a2bf6",
```

```
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-989EXAMPLE/2021/02/09/22/24-21-275",
                                "test-run-status": "FAILED",
                                "state-reason": "\"Error: net::ERR_NAME_NOT_RESOLVED
 \""
                                "canary-run-timeline": {
                                                "started": 1612909461,
                                                "completed": 1612909467
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

È possibile che gli eventi vengano duplicati o non funzionino. Per determinare l'ordine degli eventi, utilizza la proprietà `time`.

Prerequisiti per la creazione di regole EventBridge

Prima di creare una EventBridge regola per CloudWatch Synthetics, dovete fare quanto segue:

- Acquisite familiarità con eventi, regole e obiettivi in. EventBridge
- Crea e configura gli obiettivi richiamati dalle tue regole. EventBridge Le regole possono richiamare molti tipi di target, tra cui:
  - Argomenti di Amazon SNS
  - AWS Lambda funzioni
  - Flussi Kinesis
  - Code Amazon SQS

Per ulteriori informazioni, consulta What is Amazon EventBridge? e Guida introduttiva ad Amazon EventBridge nella Amazon EventBridge User Guide.

Creare una EventBridge regola (CLI)

I passaggi dell'esempio seguente creano una EventBridge regola che pubblica un argomento di Amazon SNS quando il canarino `my-canary-name` indicato completa un'esecuzione o `us-east-1` cambia lo stato.

1. Crea la regola.

```
aws events put-rule \
  --name TestRule \
  --region us-east-1 \
  --event-pattern "{\"source\": [\"aws.synthetics\"], \"detail\": {\"canary-name\":
 [\"my-canary-name\"]}}"
```

Eventuali proprietà omesse dal modello vengono ignorate.

2. Aggiungi l'argomento come destinazione della regola.

   - Sostituisci *topic-arn* con l'Amazon Resource Name (ARN) dell'argomento Amazon SNS.

```
aws events put-targets \
  --rule TestRule \
  --targets "Id"="1","Arn"="topic-arn"
```

> **Note**
>
> Per consentire EventBridge ad Amazon di indicare il tuo argomento di riferimento, devi aggiungere una politica basata sulle risorse all'argomento. Per ulteriori informazioni, consulta le autorizzazioni di Amazon SNS nella Amazon EventBridge User Guide.

Per ulteriori informazioni, consulta la sezione Eventi e modelli di eventi EventBridge nella Amazon EventBridge User Guide.

Esegui lanci ed esperimenti A/B con Evidently CloudWatch

Puoi usare Amazon CloudWatch Evidently per convalidare nuove funzionalità in modo sicuro offrendole disponibili a una percentuale specifica di utenti durante il rollout della funzionalità. È possibile monitorare le prestazioni della nuova funzionalità per aiutarti a decidere quando aumentare il traffico verso gli utenti. Ciò consente di ridurre i rischi e identificare le conseguenze non intenzionali prima di avviare completamente la funzionalità.

È inoltre possibile condurre esperimenti A/B per prendere decisioni sulla progettazione delle caratteristiche basate su prove e dati. Un esperimento può testare fino a cinque varianti contemporaneamente. Evidently raccoglie i dati degli esperimenti e li analizza utilizzando metodi

statistici. Fornisce inoltre raccomandazioni chiare su quali variazioni funzionano meglio. È possibile testare sia le funzionalità rivolte all'utente che alle funzionalità back-end.

Prezzi di Evidently

Evidently addebita il tuo account in base a eventi evidentemente e alle unità di analisi evidente. Gli eventi di Evidently includono sia eventi dati ad esempio clic e visualizzazioni di pagina, sia eventi di assegnazione che determinano la variazione di funzionalità da servire a un utente.

Le unità di analisi di Evidently sono generate da eventi Evidently, in base alle regole create su Evidently. Le unità di analisi sono il numero di regole corrispondenti agli eventi. Ad esempio, un evento clic dell'utente potrebbe produrre una singola unità di analisi di Evidently, un numero di clic. Un altro esempio è un evento di checkout utente che potrebbe produrre due unità di analisi di Evidently, il valore di checkout e il numero di articoli nel carrello. Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

CloudWatch Evidentemente è attualmente disponibile nelle seguenti regioni:

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- US West (Oregon)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Stoccolma)

Argomenti

- Visualizzare i risultati di avvio nel pannello di controllo
- Visualizzare i risultati degli esperimenti nel pannello di controllo
- How CloudWatch Evidently raccoglie e archivia i dati
- Utilizzo di ruoli collegati ai servizi per Evidently
- CloudWatch Evidentemente quote
- Esercitazione: test A/B con l'applicazione Evidently di esempio

Policy IAM per utilizzare Evidently

Per gestire completamente CloudWatch Evidently, devi accedere come utente o ruolo IAM con le seguenti autorizzazioni:

- La policy AmazonCloudWatchEvidentlyFullAccess
- La policy ResourceGroupsandTagEditorReadOnlyAccess

Inoltre, per poter creare un progetto che memorizzi gli eventi di valutazione in Amazon S3 o CloudWatch Logs, sono necessarie le seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPolicy",
                "s3:PutBucketPolicy",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:PutResourcePolicy"
            ],
```

```
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Autorizzazioni aggiuntive per l'integrazione RUM CloudWatch

Inoltre, se intendi gestire lanci o esperimenti di Evidently che si integrano con Amazon CloudWatch RUM e utilizzare i parametri CloudWatch RUM per il monitoraggio, hai bisogno della AmazonCloudWatch policy RUM. FullAccess Per creare un ruolo IAM che consenta al client web CloudWatch RUM di inviare dati a CloudWatch RUM, sono necessarie le seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*",
                "arn:aws:iam::*:policy/service-role/CloudWatchRUMEvidentlyPolicy-*"
            ]
        }
    ]
}
```

Autorizzazioni per l'accesso in sola lettura a Evidently

Per gli altri utenti che hanno bisogno di visualizzare i dati di Evidently ma non hanno bisogno di creare risorse Evidently, puoi concedere la policy. AmazonCloudWatchEvidentlyReadOnlyAccess

Crea progetti, funzionalità, lanci ed esperimenti

Per iniziare a usare CloudWatch Evidently, sia per il lancio di una funzionalità che per un esperimento A/B, devi prima creare un progetto. Un progetto è un raggruppamento logico di risorse. All'interno del

progetto, si creano caratteristiche che hanno variazioni da testare o lanciare. E' possibile creare una funzione prima di creare un lancio o un esperimento o contemporaneamente.

Argomenti

Crea un nuovo progetto

Usa questi passaggi per configurare un nuovo progetto CloudWatch Evidently.

Per creare un nuovo progetto CloudWatch Evidently

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Seleziona Crea progetto.
4. Per Nome progetto, inserisci un nome da utilizzare per identificare questo progetto nella console CloudWatch Evidently.

   E' anche possbilie anche aggiungere una descrizione opzionale.
5. Per Archiviazione eventi di valutazione, scegli se desideri memorizzare gli eventi di valutazione che si raccolgono con Evidently. Anche se non si memorizzano questi eventi, Evidently li aggrega per creare metriche e altri dati sperimentali che è possibile visualizzare nel pannello di controllo Evidently. Per ulteriori informazioni, consulta Archiviazione dati di progetto.
6. Per Use client-side evaluation (Utilizza la valutazione lato client), scegli se abilitare la valutazione lato client per questo progetto. Con la valutazione lato client, l'applicazione può assegnare variazioni alle sessioni utente localmente anziché richiamare l'operazione. EvaluateFeature In questo modo si riducono i rischi di latenza e disponibilità associati a una chiamata API. Per ulteriori informazioni, consulta Utilizzo della valutazione lato client con tecnologia AWS AppConfig.

Per creare un progetto con una valutazione lato client, è necessario disporre dell'autorizzazione `evidently:ExportProjectAsConfiguration`.

Quando abiliti la valutazione lato client, esegui anche le operazioni seguenti:

a. Scegliete se utilizzare un' AWS AppConfig applicazione esistente o crearne una nuova.

b. Scegli se utilizzare un AWS AppConfig ambiente esistente o crearne uno nuovo.

Per ulteriori informazioni su applicazioni e ambienti in AWS AppConfig, vedi Come AWS AppConfig funziona.

7. (Facoltativo) Per aggiungere tag a questo progetto, scegli Tag,Aggiungi nuovo tag.

Poi per Chiave, inserire un nome per il tag. È possibile aggiungere un valore facoltativo al tag in Value (Valore).

Per aggiungere un altro tag, scegli nuovamente Add tag (Aggiungi tag).

Per ulteriori informazioni, consulta Tagging AWS Resources.

8. Seleziona Crea progetto.

Utilizzo della valutazione lato client con tecnologia AWS AppConfig

È possibile utilizzare la valutazione lato client - powered by AWS AppConfig (valutazione lato client) in un progetto, che consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché assegnare variazioni richiamando l'operazione. EvaluateFeature In questo modo si riducono i rischi di latenza e disponibilità associati a una chiamata API.

Per utilizzare la valutazione lato client, collega l'estensione AWS AppConfig Lambda come livello alle funzioni Lambda e configura le variabili di ambiente. La valutazione lato client viene eseguita come processo secondario sull'host locale. Quindi, puoi richiamare le operazioni and contro. EvaluationFeaturePutProjectEvent`localhost` Il processo di valutazione lato client gestisce l'assegnazione delle varianti, l'archiviazione nella cache e la sincronizzazione dei dati. Per ulteriori informazioni su AWS AppConfig, consulta How AWS AppConfig works.

Quando esegui l'integrazione con AWS AppConfig, specifichi un ID AWS AppConfig dell'applicazione e un ID di AWS AppConfig ambiente su Evidently. Puoi utilizzare lo stesso ID applicazione e lo stesso ID ambiente in tutti i progetti Evidently.

Quando crei un progetto con la valutazione lato client abilitata, Evidently crea un profilo di AWS AppConfig configurazione per quel progetto. Il profilo di configurazione sarà diverso per ogni progetto.

Controllo degli accessi per la valutazione lato client

La valutazione lato client di Evidently utilizza un meccanismo di controllo degli accessi diverso rispetto alle altre funzionalità di Evidently. Ti consigliamo di acquisire familiarità con questo meccanismo in modo da poter implementare le misure di sicurezza adeguate.

Con Evidently puoi creare policy IAM che limitano le operazioni che un utente può eseguire sulle singole risorse. Ad esempio, puoi creare un ruolo utente che impedisca a un utente di eseguire l'azione. EvaluateFeature Per ulteriori informazioni sulle azioni Evidently che possono essere controllate con le policy IAM, consulta Actions defined by Amazon CloudWatch Evidently.

Il modello di valutazione lato client consente valutazioni locali delle funzionalità di Evidently che utilizzano i metadati del progetto. Un utente di un progetto con la valutazione lato client abilitata può chiamare l'EvaluateFeatureAPI su un endpoint host locale e questa chiamata API non raggiunge Evidently e non è autenticata dalle politiche IAM del servizio Evidently. Questa chiamata ha esito positivo anche se l'utente non dispone dell'autorizzazione IAM per utilizzare l'azione. EvaluateFeature Tuttavia, un utente necessita comunque dell'PutProjectEventsautorizzazione dell'agente per memorizzare nel buffer gli eventi di valutazione o gli eventi personalizzati e per scaricare i dati su Evidently in modo asincrono.

L'utente deve inoltre disporre dell'autorizzazione `evidently:ExportProjectAsConfiguration` per poter creare un progetto che utilizza la valutazione lato client. Ciò consente di controllare l'accesso alle EvaluateFeatureazioni richiamate durante la valutazione lato client.

Se non si procede con attenzione, il modello di sicurezza della valutazione lato client può sovvertire le policy impostate sul resto di Evidently. Un utente che dispone dell'`evidently:ExportProjectAsConfiguration`autorizzazione può creare un progetto con la valutazione lato client abilitata e quindi utilizzare l'EvaluateFeatureazione per la valutazione lato client con quel progetto anche se gli viene espressamente negata l'azione in una policy IAM. EvaluateFeature

Inizia a usare Lambda

Evidently attualmente supporta la valutazione lato client utilizzando un ambiente AWS Lambda . Per iniziare, decidi innanzitutto quale AWS AppConfig applicazione e ambiente utilizzare. Scegli un'applicazione e un ambiente esistenti o creane di nuovi.

I seguenti AWS AppConfig AWS CLI comandi di esempio creano un'applicazione e un ambiente.

```
aws appconfig create-application --name YOUR_APP_NAME
```

```
aws appconfig create-environment --application-id YOUR_APP_ID --
name YOUR_ENVIRONMENT_NAME
```

Quindi, crea un progetto Evidently utilizzando queste AWS AppConfig risorse. Per ulteriori informazioni, consulta Crea un nuovo progetto.

La valutazione lato client è supportata in Lambda tramite un livello Lambda. Questo è un livello pubblico che fa parte di`AWS-AppConfig-Extension`, un' AWS AppConfig estensione pubblica creata dal AWS AppConfig servizio. Per ulteriori informazioni sui livelli Lambda, consulta la sezione Livello.

Per utilizzare la valutazione lato client, devi aggiungere questo livello alla funzione Lambda e configurare le autorizzazioni e le variabili di ambiente.

Aggiunta e configurazione del livello Lambda per la valutazione lato client di Evidently alla funzione Lambda

1. Se non lo hai già fatto, crea una funzione Lambda.
2. Aggiungi il livello per la valutazione lato client alla tua funzione. Puoi specificarne l'ARN o selezionarlo dall'elenco dei AWS livelli, se non l'hai già fatto. Per ulteriori informazioni, consulta Configurazione delle funzioni per l'uso dei livelli e Versioni disponibili dell'estensione AWS AppConfig Lambda.
3. Crea una policy IAM denominata EvidentlyAppConfigCachingAgentPolicycon i seguenti contenuti e collegala al ruolo di esecuzione della funzione. Per ulteriori informazioni, consulta Ruolo di esecuzione Lambda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "appconfig:GetLatestConfiguration",
                "appconfig:StartConfigurationSession",
                "evidently:PutProjectEvents"
            ],
```

```
            "Resource": "*"
        }
    ]
}
```

4. Aggiungi la variabile di ambiente `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` richiesta alla funzione Lambda. Questa variabile di ambiente specifica la mappatura tra il progetto Evidently e le risorse. AWS AppConfig

   Se stai usando questa funzione per un progetto Evidently, imposta il valore della variabile di ambiente su `applications/`*`APP_ID`*`/environments/`*`ENVIRONMENT_ID`*`/configurations/`*`PROJECT_NAME`*.

   Se stai usando questa funzione per più progetti Evidently, usa una virgola per separare i valori, come nell'esempio seguente: `applications/`*`APP_ID_1`*`/environments/`*`ENVIRONMENT_ID_1`*`/configurations/`*`PROJECT_NAME_1`*`,applications/`*`APP_ID_2`*`/environments/`*`ENVIRONMENT_ID_2`*`/configurations/`*`PROJECT_NAME_2`*

5. (Facoltativo) Imposta altre variabili di ambiente. Per ulteriori informazioni, consulta Configurazione dell'estensione AWS AppConfig Lambda.

6. Nella tua applicazione, ottieni le valutazioni di Evidently a livello locale inviando `EvaluateFeature` a `localhost`.

   Esempio di Python:

```
import boto3
from botocore.config import Config

def lambda_handler(event, context):
    local_client = boto3.client(
        'evidently',
        endpoint_url="http://localhost:2772",
        config=Config(inject_host_prefix=False)
    )
    response = local_client.evaluate_feature(
        project=event['project'],
        feature=event['feature'],
        entityId=event['entityId']
    )
```

```
    print(response)
```

Esempio di Node.js:

```
const AWS = require('aws-sdk');
const evidently = new AWS.Evidently({
    region: "us-west-2",
    endpoint: "http://localhost:2772",
    hostPrefixEnabled: false
});

exports.handler = async (event) => {

    const evaluation = await evidently.evaluateFeature({
        project: 'John_ETCProject_Aug2022',
        feature: 'Feature_IceCreamFlavors',
        entityId: 'John'
    }).promise()

    console.log(evaluation)
    const response = {
        statusCode: 200,
        body: evaluation,
    };
    return response;
};
```

Esempio di Kotlin:

```
String localhostEndpoint = "http://localhost:2772/"
public AmazonCloudWatchEvidentlyClient getEvidentlyLocalClient() {
    return AmazonCloudWatchEvidentlyClientBuilder.standard()

 .withEndpointConfiguration(AwsClientBuilder.EndpointConfiguration(localhostEndpoint,
 region))

 .withClientConfiguration(ClientConfiguration().withDisableHostPrefixInjection(true))
        .withCredentials(credentialsProvider)
        .build();
}

AmazonCloudWatchEvidentlyClient evidently = getEvidentlyLocalClient();
```

```
// EvaluateFeature via local client.
EvaluateFeatureRequest evaluateFeatureRequest = new
 EvaluateFeatureRequest().builder()
   .withProject(${YOUR_PROJECT}) //Required.
   .withFeature(${YOUR_FEATURE}) //Required.
   .withEntityId(${YOUR_ENTITY_ID}) //Required.
   .withEvaluationContext(${YOUR_EVAL_CONTEXT}) //Optional: a JSON object of
 attributes that you can optionally pass in as part of the evaluation event sent to
 Evidently.
   .build();

EvaluateFeatureResponse evaluateFeatureResponse =
 evidently.evaluateFeature(evaluateFeatureRequest);

// PutProjectEvents via local client.
PutProjectEventsRequest putProjectEventsRequest = new
 PutProjectEventsRequest().builder()
    .withData(${YOUR_DATA})
    .withTimeStamp(${YOUR_TIMESTAMP})
    .withType(${YOUR_TYPE})
    .build();

PutProjectEvents putProjectEventsResponse =
 evidently.putProjectEvents(putProjectEventsRequest);
```

Configurazione della frequenza con cui il client invia i dati a Evidently

Per specificare la frequenza con cui la valutazione lato client invia i dati a Evidently, puoi facoltativamente configurare due variabili di ambiente.

- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_EVENT_BATCH_SIZE` specifica il numero di eventi per progetto da raggruppare in batch prima di inviarli a Evidently. I valori validi sono numeri interi compresi tra 1 e 50 e il valore predefinito è 40.
- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_BATCH_COLLECTION_DURATION` specifica il tempo di attesa in secondi prima dell'invio degli eventi a Evidently. Il valore predefinito è 30.

Risoluzione dei problemi

Utilizza le seguenti informazioni per risolvere i problemi relativi all'utilizzo di CloudWatch Evidently con la valutazione lato client, fornita da. AWS AppConfig

Si è verificato un errore (BadRequestException) durante la chiamata dell' EvaluateFeature operazione: metodo HTTP non supportato per il percorso fornito

Le variabili di ambiente potrebbero essere configurate in modo errato. Ad esempio, potresti aver usato `EVIDENTLY_CONFIGURATIONS` come nome della variabile di ambiente invece di `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS`.

ResourceNotFoundException: Distribuzione non trovata

L'aggiornamento dei metadati del progetto non è stato implementato su AWS AppConfig. Verifica la presenza di una distribuzione attiva nell' AWS AppConfig ambiente utilizzato per la valutazione lato client.

ValidationException: Nessuna configurazione evidentemente per il progetto

La variabile di ambiente `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` potrebbe essere configurata con il nome del progetto errato.

Aggiunta di una funzionalità a un progetto

Una funzionalità in CloudWatch Evidently rappresenta una funzionalità che desideri avviare o di cui desideri testare le varianti.

Per poter aggiungere una funzionalità, è necessario creare un progetto. Per ulteriori informazioni, consulta Crea un nuovo progetto.

Per aggiungere una funzionalità a un progetto

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto.
4. Scegli Aggiungi funzionalità.
5. Per Nome delle funzionalità, immetti un nome da utilizzare per identificare questa funzionalità all'interno di questo progetto.

   Facoltativamente, è possibile anche aggiungere una descrizione della funzionalità.
6. Per Variazioni di funzionalità, per Tipo di variazione scegli booleano,lungo,doppio, oppure stringa. Per ulteriori informazioni, consulta Tipi di varianti.
7. Aggiungere fino a cinque varianti per la funzionalità. Il valore per ogni variante deve essere valido per il Tipo di variazione selezionato.

Definire una delle varianti come di default. Questa è la linea di base con cui verranno confrontate le altre varianti e dovrebbe essere la variante che viene servita ai tuoi utenti ora. Questa è anche la variante che viene servita agli utenti che non vengono aggiunti a un lancio o a un esperimento per questa funzionalità.

8. Scegli Codice di esempio. L'esempio di codice mostra cosa è necessario aggiungere all'applicazione per impostare le varianti e assegnarle le sessioni utente. Puoi scegliere tra JavaScript Java e Python per il codice.

   Non è necessario aggiungere il codice alla tua applicazione in questo momento, ma devi farlo prima di iniziare un lancio o un esperimento.

   Per ulteriori informazioni, consulta Aggiungere un codice all'applicazione.

9. (Facoltativo) Per definire che alcuni utenti vedono sempre una certa variazione, scegli Sovrascritture, Aggiungi sovrascrittura. Quindi, definire un utente inserendo il proprio ID utente, l'ID account o un altro identificatore in Identifier (Identificatore) e specificare quale variazione dovrebbero vedere.

   Questo può essere utile per i membri del proprio team di test o di altri utenti interni quando vuoi essere sicuro che vedano una variazione specifica. Le sessioni degli utenti a cui sono assegnate le sostituzioni non contribuiscono al lancio o all'esperimento di parametri.

   Puoi ripetere questa operazione per un massimo di 20 utenti selezionando nuovamente Aggiungi override.

10. (Facoltativo) Per aggiungere tag a questa funzione, scegli Tag,Aggiungi nuovo tag.

    Poi per Chiave, inserire un nome per il tag. È possibile aggiungere un valore facoltativo al tag in Value (Valore).

    Per aggiungere un altro tag, scegli nuovamente Add tag (Aggiungi tag).

    Per ulteriori informazioni, consulta Tagging AWS Resources.

11. Scegli Aggiungi funzionalità.

Tipi di varianti

Quando si crea una funzionalità e si descrivono le varianti, è necessario selezionare un tipo di variazione. Le tipologie possibili sono:

- Booleano
- numero intero lungo
- Numero in virgola mobile a precisione doppia
- Stringa

Il tipo di variante imposta il modo in cui le diverse varianti vengono differenziate nel codice. È possibile utilizzare il tipo di variante per semplificare l'implementazione di CloudWatch Evidently e anche per semplificare il processo di modifica delle funzionalità nei lanci e negli esperimenti.

Ad esempio, se si definisce una funzionalità con il tipo di variazione intera lunga, i numeri interi specificati per differenziare le varianti possono essere numeri passati direttamente nel codice. Un esempio potrebbe essere il test della dimensione in pixel di un pulsante. I valori per i tipi di variazione possono essere il numero di pixel utilizzati in ciascuna variante. Il codice per ogni variante può leggere il valore del tipo di variante e utilizzarlo come dimensione del pulsante. Per testare una nuova dimensione del pulsante, è possibile modificare il numero utilizzato per il valore della variante, senza apportare altre modifiche al codice.

Quando impostate i valori per i tipi di variazione all'interno di una funzionalità, dovreste evitare di assegnare gli stessi valori a più varianti, a meno che non vogliate eseguire test A/A per provare inizialmente CloudWatch Evidently o non abbiate altri motivi per farlo.

Evidently non ha il supporto nativo per JSON come tipo, ma è possibile passare JSON nel tipo di variante di stringa e analizzare JSON nel codice.

Usa i segmenti per focalizzare il tuo pubblico

È possibile definire i segmenti di pubblico e usarli per gli avviamenti e gli esperimenti. Un segmento è una parte del pubblico che condivide una o più caratteristiche. Ad esempio gli utenti del browser Chrome, gli utenti in Europa o gli utenti del browser Firefox in Europa che soddisfano anche altri criteri raccolti dall'applicazione, come l'età.

L'utilizzo di un segmento in un esperimento limita tale esperimento per valutare solo gli utenti che corrispondono ai criteri del segmento. Quando utilizzi uno o più segmenti in un avvio puoi definire diverse divisioni di traffico per i diversi segmenti di pubblico.

Sintassi di modelli di regole di segmento

Per creare un segmento, definisci un modello di regole di segmento che specifica gli attributi da utilizzare per valutare se una sessione utente sarà nel segmento. Il modello creato viene confrontato

con il valore di `evaluationContext` che Evidently trova in una sessione utente. Per ulteriori informazioni, consulta Usando EvaluateFeature.

Per creare un modello di regole di segmento, specificare i campi a cui si desidera che il modello corrisponda. Puoi anche usare la logica nel tuo modello, ad esempio `And`, `Or`, `Not` ed `Exists`.

Affinché `evaluationContext` corrisponda a un modello, `evaluationContext` deve corrispondere a tutte le parti del modello di regole. Evidentemente ignora i campi nel `evaluationContext` che non sono inclusi nel modello di regole.

I valori che corrispondono ai modelli di regole seguono le regole JSON. È possibile includere stringhe racchiuse tra virgolette ("), numeri e parole chiave `true`, `false`, e `null`.

Per le stringhe, Evidently utilizza la character-by-character corrispondenza esatta senza ripiegare tra maiuscole e minuscole o qualsiasi altra normalizzazione delle stringhe. Le corrispondenze delle regole fanno quindi distinzione tra maiuscole e minuscole. Ad esempio, se `evaluationContext` include un attributo `browser` ma il modello di regole verifica la presenza di `Browser`, non troverà una corrispondenza.

Per i numeri Evidently utilizza anche la rappresentazione di stringhe. Ad esempio, 300, 300.0 e 3.0e2 non sono considerati uguali.

Quando si scrivono modelli di regole di regole per la corrispondenza con `evaluationContext`, è possibile utilizzare l'API `TestSegmentPattern` o il comando CLI `test-segment-pattern` per verificare che il modello corrisponda al JSON corretto. Per ulteriori informazioni, vedere. TestSegmentPattern

Il riepilogo seguente mostra tutti gli operatori di confronto disponibili nei modelli di segmento di Evidently.

Confronto	Esempio	Sintassi delle regole
Null	UserID è nullo	`{ "UserID": [ null ] }`
Empty	LastName è vuoto	`{ "LastName": [""]`

Confronto	Esempio	Sintassi delle regole
		`}`
Equals	Il browser è "Chrome"	`{ "Browser": [ "Chrome" ] }`
And	Il paese è "Francia" e il dispositivo è "Mobile"	`{ "Country": [ "France" ], "Device": ["Mobile"] }`
Oppure (più valori di un singolo attributo)	Il browser è "Chrome" o "Firefox"	`{ "Browser": ["Chrome", "Firefox"] }`
Oppure (attributi diversi)	Il browser è "Safari" o il dispositivo è "Tablet"	`{ "$or": [ {"Browser": ["Safari"]}, {"Device": ["Tablet" }] ] }`
Not	Il browser è tutto tranne che "Safari"	`{ "Browser": [ { "anything-but": [ "Safari" ] } ] }`

Confronto	Esempio	Sintassi delle regole
Numeric (uguale)	Il prezzo è 100	`{ "Price": [ { "numeric": [ "=", 100 ] } ] }`
Numeric (intervallo)	Il prezzo è superiore a 10 e inferiore o uguale a 20	`{ "Price": [ { "numeric": [ ">", 10, "<=", 20 ] } ] }`
Exists	Il campo Età esiste	`{ "Age": [ { "exists": true } ] }`
Does not exist	Il campo Età non esiste	`{ "Age": [ { "exists": false } ] }`
Inizia con un prefisso	La regione è negli Stati Uniti	`{ "Region": [ {"prefix": "us-" } ] }`
Termina con un suffisso	La posizione ha il suffisso "West"	`{ "Region": [ {"suffix": "West" } ] }`

Esempi di regole di segmento

Tutti questi esempi seguenti presuppongono che si passino i valori per `evaluationContext` con le stesse etichette di campo e gli stessi valori utilizzati nei modelli di regole.

L'esempio seguente corrisponde se `Browser` è Chrome o Firefox e `Location` è US-West.

```
{
  "Browser": ["Chrome", "Firefox"],
  "Location": ["US-West"]
}
```

L'esempio seguente corrisponde se `Browser` è qualsiasi browser tranne Chrome, `Location` inizia con US ed esiste un campo `Age`.

```
{
  "Browser": [ {"anything-but": ["Chrome"]}],
  "Location": [{"prefix": "US"}],
  "Age": [{"exists": true}]
}
```

L'esempio seguente corrisponde se `Location` è il Giappone e `Browser` è Safari o `Device` è Tablet.

```
{
 "Location": ["Japan"],
 "$or": [
   {"Browser": ["Safari"]},
   {"Device": ["Tablet"]}
 ]
}
```

Creazione di un segmento

Dopo aver creato un segmento, puoi usarlo in qualsiasi avvio o esperimento di qualsiasi progetto.

Per creare un segmento

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Seleziona la scheda Segments (Segmenti).
4. Scegli Create segment (Crea segmento).

5. Per Segment name (Nome del segmento), immetti un nome da usare per identificare questo segmento.

   Facoltativamente puoi aggiungere una descrizione.

6. Per Modelli di segmenti, immetti un blocco JSON che definisca il modello di regole. Per ulteriori informazioni sulla sintassi del modello di regole, consulta Sintassi di modelli di regole di segmento.

Creazione di un avvio

Per esporre una nuova funzionalità o modificare una percentuale specificata di utenti, creare un avvio. È quindi possibile monitorare i parametri chiave ad esempio i tempi di caricamento delle pagine e le conversioni prima di distribuire la funzionalità a tutti gli utenti.

Prima di poter aggiungere un avvio, è necessario aver creato un progetto. Per ulteriori informazioni, consulta Crea un nuovo progetto.

Quando si aggiunge un avvio, è possibile utilizzare una funzionalità già creata o creare una nuova funzionare durante la creazione dell'avvio.

Aggiunta di un avvio a un progetto

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Seleziona il pulsante accanto al nome del progetto e scegli Azioni progetto, Crea avvio.
4. Per Nome avvio, immetti un nome da utilizzare per identificare questa funzionalità all'interno di questo progetto.

   Facoltativamente, è possibile anche aggiungere una descrizione.

5. Scegli Seleziona tra le funzionalità esistenti o Aggiungi una nuova funzionalità.

   Se utilizzi una funzione esistente, selezionarla sotto Nome delle funzionalità.

   Se si sceglie Aggiunta di una nuova funzionalità, eseguire le seguenti operazioni:

   a. Per Nome delle funzionalità, immetti un nome da utilizzare per identificare questa funzionalità all'interno di questo progetto. Facoltativamente, è possibile anche aggiungere una descrizione.

b. Per Variazioni di funzionalità, per Tipo di variazione scegli booleano,lungo,doppio, oppure stringa. Per ulteriori informazioni, consulta Tipi di varianti.

c. Aggiungere fino a cinque varianti per la funzionalità. Il valore per ogni variante deve essere valido per il Tipo di variazione selezionato.

   Definire una delle varianti come di default. Questa è la linea di base con cui verranno confrontate le altre varianti e dovrebbe essere la variante che viene servita ai tuoi utenti ora. Se si interrompe un esperimento, questa variante di default verrà servita a tutti gli utenti.

d. Scegli Codice di esempio. L'esempio di codice mostra cosa è necessario aggiungere all'applicazione per impostare le varianti e assegnarle le sessioni utente. Puoi scegliere tra JavaScript Java e Python per il codice.

   Non è necessario aggiungere il codice alla propria applicazione in questo momento, ma è necessario farlo prima di iniziare l'avvio.

   Per ulteriori informazioni, consulta Aggiungere un codice all'applicazione.

6. Per Configurazione di avvio, scegli se avviare immediatamente l'avvio o programmarlo in un secondo momento.

7. (Facoltativo) Per specificare suddivisioni di traffico diverse per i segmenti di pubblico che hai definito, invece della suddivisione del traffico che utilizzerai per il pubblico generale, scegli Add Segment Overrides (Aggiungi sostituzioni di segmenti).

   Nello Segment Overrides (Sostituzioni di segmenti), seleziona un segmento e definire la suddivisione del traffico da utilizzare per quel segmento.

   Facoltativamente, è possibile definire più segmenti per definire i subdivisioni del traffico scegliendoAdd Segment Overrides (Aggiungi sostituzioni di segmenti). Un avvio può avere fino a sei sostituzioni di segmenti.

   Per ulteriori informazioni, consulta Usa i segmenti per focalizzare il tuo pubblico.

8. Per Traffic confirguration (Configurazione del traffico), seleziona la percentuale di traffico da assegnare a ciascuna variante per il pubblico generale che non corrisponde alle sostituzioni di segmenti. Puoi inoltre scegliere di escludere che le variazioni vengano inviate agli utenti.

   Il Riepilogo traffico mostra la quantità di traffico globale disponibile per questo avvio.

9. Se si sceglie di pianificare l'avvio in un secondo momento, è possibile aggiungere più passaggi all'avvio. Ogni passaggio può utilizzare percentuali diverse per servire le variazioni. Per farlo,

scegli Aggiunta di un'altra fase e quindi definisci la pianificazione e le percentuali di traffico per il passaggio successivo. È possibile includere fino a cinque passaggi in un avvio.

10. Se desideri monitorare le prestazioni delle funzionalità con i parametri durante l'avvio, scegli Parametri,Aggiungi parametro. È possibile utilizzare metriche CloudWatch RUM o metriche personalizzate.

    Per utilizzare una metrica personalizzata, puoi creare la metrica qui utilizzando una regola Amazon EventBridge . Per creare un parametro personalizzato, eseguire le seguenti operazioni:

    - Scegli Parametri personalizzati e immetti un nome per il parametro.
    - In Regola parametro, perID entità, inserire il modo di identificazione dell'entità. Può trattarsi di un utente o di una sessione che esegue un'azione che causa la registrazione del un valore di un parametro. Un esempio è `userDetails.userID`.
    - Per Chiave valore, immetti il valore che deve essere monitorato per produrre il parametro.
    - Facoltativamente, immetti un nome per le unità per il parametro. Il nome dell'unità è solo a scopo di visualizzazione, da utilizzare sui grafici della console Evidently.

    Quando inserisci questi campi, la casella mostra esempi di come codificare la EventBridge regola per creare la metrica. Per ulteriori informazioni su EventBridge, consulta What Is Amazon EventBridge?

    Per utilizzare i parametri RUM, è necessario disporre di un monitor dell'app RUM configurato per la propria applicazione. Per ulteriori informazioni, consulta Configura un'applicazione per utilizzare CloudWatch RUM.

    > **Note**
    >
    > Se si utilizzano metriche RUM e il monitor dell'app non è configurato per campionare il 100% delle sessioni utente, non tutte le sessioni utente che partecipano all'avvio invieranno parametri a Evidently. Per garantire che i parametri di avvio siano accurati, si consiglia che il monitor dell'app utilizzi il 100% delle sessioni utente per il campionamento.

11. (Facoltativo) Se crei almeno una metrica per il lancio, puoi associare un CloudWatch allarme esistente a questo lancio. Per farlo, scegli Associa CloudWatch allarmi.

Quando associ un allarme a un lancio, CloudWatch Evidently devi aggiungere dei tag all'allarme con il nome del progetto e il nome del lancio. In questo modo CloudWatch Evidently può visualizzare gli allarmi corretti nelle informazioni di avvio della console.

Per confermare che CloudWatch Evidently aggiungerà questi tag, scegli Consenti Evidently per etichettare la risorsa di allarme identificata di seguito con questa risorsa di avvio. Quindi, scegli Allarme associato e immetti il nome dell'allarme.

Per informazioni sulla creazione di CloudWatch allarmi, consulta. Utilizzo degli CloudWatch allarmi Amazon

12. (Facoltativo) Per aggiungere tag a questo avvio, scegli Tag,Aggiungi nuovo tag.

    Poi per Chiave, inserire un nome per il tag. È possibile aggiungere un valore facoltativo al tag in Value (Valore).

    Per aggiungere un altro tag, scegli nuovamente Add tag (Aggiungi tag).

    Per ulteriori informazioni, consulta Tagging AWS Resources.

13. Scegliere Crea avvio.

Creare un esperimento

Utilizzare esperimenti per testare diverse versioni di una funzionalità o di un sito Web e raccogliere dati da sessioni utente reali. In questo modo, è possibile effettuare scelte per la propria applicazione in base a prove e dati.

Prima di poter aggiungere un esperimento, è necessario aver creato un progetto. Per ulteriori informazioni, consulta Crea un nuovo progetto.

Quando si aggiunge un esperimento, è possibile usare una funzionalità già creata o creare una nuova funzionalità mentre si crea l'esperimento.

Per aggiungere un esperimento a un progetto

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Seleziona il pulsante accanto al nome del progetto e scegli Azioni di progetto, Crea esperimenti.

4. Per Nome dell'esperimento, immetti un nome da utilizzare per identificare questa funzionalità all'interno di questo progetto.

   Facoltativamente, è possibile anche aggiungere una descrizione.

5. Scegli Seleziona tra le funzionalità esistenti o Aggiungi una nuova funzionalità.

   Se utilizzi una funzione esistente, selezionarla sotto Nome delle funzionalità.

   Se si sceglie Aggiunta di una nuova funzionalità, eseguire le seguenti operazioni:

   a. Per Nome delle funzionalità, immetti un nome da utilizzare per identificare questa funzionalità all'interno di questo progetto. Facoltativamente, è possibile inserire una descrizione.

   b. Per Variazioni di funzionalità, perTipo di variazione scegli booleano,lungo,doppio, oppure stringa. Il tipo definisce il tipo di valore utilizzato per ogni variante. Per ulteriori informazioni, consulta Tipi di varianti.

   c. Aggiungere fino a cinque varianti per la funzionalità. Il valore per ogni variante deve essere valido per il Tipo di variazione selezionato.

      Definire una delle varianti come di default. Questa è la linea di base con cui verranno confrontate le altre varianti e dovrebbe essere la variante che viene servita ai tuoi utenti ora. Se si interrompe un esperimento che utilizza questa funzionalità, la variazione di default viene quindi servita alla percentuale di utenti che erano presenti nell'esperimento in precedenza.

   d. Scegli Codice di esempio. L'esempio di codice mostra cosa è necessario aggiungere all'applicazione per impostare le varianti e assegnarle le sessioni utente. Puoi scegliere tra JavaScript Java e Python per il codice.

      Non è necessario aggiungere il codice alla tua applicazione in questo momento, ma è necessario farlo prima di iniziare l'esperimento. Per ulteriori informazioni, consulta Aggiungere un codice all'applicazione.

6. Per Audience (Pubblico), seleziona facoltativamente un segmento creato se si desidera che questo esperimento si applichi solo agli utenti che corrispondono a quel segmento. Per ulteriori informazioni sui segmenti, consulta Usa i segmenti per focalizzare il tuo pubblico.

7. Per Traffic split for the experiment (Suddivisione del traffico per l'esperimento), specificare la percentuale del pubblico selezionato le cui sessioni saranno utilizzate nell'esperimento. Quindi allocare il traffico per le diverse varianti utilizzate dall'esperimento.

Se un avvio e un esperimento sono entrambi in esecuzione contemporaneamente per la stessa funzionalità, i destinatari vengono prima indirizzati all'avvio. Quindi, la percentuale di traffico specificata per l'avvio viene presa dai destinatari generali. Dopodiché, la percentuale definita qui è la percentuale dei destinatari rimanente utilizzata per l'esperimento. Qualsiasi traffico rimanente dopo che viene servito con la variazione di default.

8. Per Parametri, scegli i parametri da utilizzare per valutare le variazioni durante l'esperimento. È necessario utilizzare almeno un parametro per la valutazione.

   a. Per Metric source, scegli se utilizzare metriche CloudWatch RUM o metriche personalizzate.

   b. Immetti un nome per il parametro. Per Obiettivo, scegli Increase (Aumenta) se si desidera un valore più alto per il parametro per indicare una variazione migliore. Scegli Decrease (Riduci) se desideri un valore inferiore per il parametro per indicare una variazione migliore.

   c. Se utilizzi una metrica personalizzata, puoi creare la metrica qui utilizzando una regola Amazon EventBridge . Per creare un parametro personalizzato, eseguire le seguenti operazioni:

      - In Regola parametro, perID entità, immetti un modo per identificare l'entità, Questo può essere un utente o una sessione che esegue un'azione che causa la registrazione di un valore di parametro. Un esempio è `userDetails.userID`.
      - Per Chiave valore, immetti il valore che deve essere monitorato per produrre il parametro.
      - Facoltativamente, immetti un nome per le unità per il parametro. Il nome dell'unità è solo a scopo di visualizzazione, da utilizzare sui grafici della console Evidently.

      È possibile utilizzare i parametri RUM solo se è stato impostato RUM per monitorare questa applicazione. Per ulteriori informazioni, consulta Usa CloudWatch RUM.

      > **Note**
      >
      > Se si utilizzano parametri RUM e il monitor dell'app non è configurato per campionare il 100% delle sessioni utente, non tutte le sessioni utente nell'esperimento invieranno parametri a Evidently. Per garantire che i parametri dell'esperimento siano accurate, raccomandiamo che il monitor dell'app utilizzi il 100% delle sessioni utente per il campionamento.

   d. (Facoltativo) Per aggiungere altri parametri da valutare, scegli Aggiungi parametro. È possibile valutare fino a tre parametri durante l'esperimento.

9. (Facoltativo) Per creare CloudWatch allarmi da utilizzare con questo esperimento, scegli allarmi. CloudWatch Gli allarmi possono controllare se la differenza di risultati tra ciascuna variante e quella di default è maggiore di una soglia specificata. Se le prestazioni di una variazione sono peggiori della variante di default e la differenza è maggiore della soglia, entra in stato di allarme e ti avvisa.

   La creazione di un allarme qui crea un allarme per ogni variante che non è la variante di default.

   Se ne crei uno nuovo allarme, definisci quanto segue:

   - Per Nome parametro, scegli il parametro esperimento da utilizzare per l'allarme.
   - Per Condizioni di allarme scegli quale condizione fa sì che l'allarme entri in stato di allarme, quando i valori dei parametri di variazione vengono confrontati con i valori dei parametri di variazione di default. Ad esempio, scegli Maggiore o Maggiore/Uguale se numeri più alti indicano che la variazione indica che sta funzionando male. Ciò sarebbe appropriato se il parametro sta misurando il tempo di caricamento della pagina, ad esempio.
   - Immetti un numero per la soglia, ovvero la differenza percentuale di prestazioni che causerà l'ingresso dell'allarme nello stato `ALARM`.
   - Per Media nel periodo, scegli la quantità di dati di parametri per ogni variante aggregati prima di essere confrontati.

   È possibile scegliere Aggiungere nuovo allarme per aggiungere altri allarmi all'esperimento.

   Quindi, scegli Impostare le notifiche per l'allarme e seleziona o crea un argomento Amazon Simple Notification Service a cui inviare notifiche di avviso. Per ulteriori informazioni, consulta la sezione Impostazione delle notifiche Amazon SNS,

10. (Facoltativo) Per aggiungere tag a questo esperimento, scegli Tag,Aggiungi nuovo tag.

    Poi per Chiave, inserire un nome per il tag. È possibile aggiungere un valore facoltativo al tag in Value (Valore).

    Per aggiungere un altro tag, scegli nuovamente Add tag (Aggiungi tag).

    Per ulteriori informazioni, consulta AWS Tagging Resources.

11. Seleziona Create experiment (Crea esperimento).
12. Se non lo si è già fatto, crea le varianti di funzionalità nella propria applicazione.
13. Seleziona Done (Fatto). L'esperimento non inizia fino a quando non lo si avvia.

Dopo aver completato i passaggi nella procedura seguente, l'esperimento inizia immediatamente.

Per iniziare un esperimento creato

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto.
4. Seleziona la scheda Esperimenti.
5. Scegli il pulsante accanto al nome dell'esperimento e scegli Operazioni, Iniziare un esperimento.
6. (Facoltativo) Per visualizzare o modificare le impostazioni dell'esperimento che hai creato al momento della creazione, scegli Configurazione degli esperimenti.
7. Scegli un orario per la fine dell'esperimento.
8. Scegli Inizia un esperimento.

   L'esperimento inizierà immediatamente.

Gestire funzionalità, avvii ed esperimenti

Utilizzare le procedure contenute in queste sezioni per gestire le funzionalità, gli avvi e gli esperimenti creati.

Argomenti

Consulta le regole di valutazione attuali e il traffico del pubblico per una funzionalità

Puoi usare la console CloudWatch Evidently per vedere come le regole di valutazione della funzionalità ripartiscono il traffico del pubblico tra i lanci, gli esperimenti e le varianti attuali della funzionalità.

Per visualizzare il traffico del pubblico per una funzionalità

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto contenente la funzionalità.
4. Scegli la scheda Caratteristiche.
5. Scegli il nome della funzionalità.

   Nella scheda Regole di valutazione, è possibile vedere il flusso di traffico di pubblico per la tua funzionalità, come segue:

   - Innanzitutto, vengono valutate le sovrascritture. Questi definiscono che a determinati utenti viene sempre servita una variante specifica. Le sessioni degli utenti a cui sono assegnate le sostituzioni non contribuiscono al lancio o all'esperimento di parametri.
   - Successivamente, il traffico rimanente è disponibile per l'avvio in corso, se ce n'è uno. Se è in corso un lancio, la scheda nella sezione Avvio di visualizza il nome di avvio e il traffico di lancio suddiviso tra le varianti di funzionalità. Sul lato destro della sezione Avviii, un indicatore di Trafficovisualizza la quantità di destinatari disponibili (dopo le sostituzioni) allocata a questo avvio. Il resto del traffico non allocato al lancio scorre all'esperimento (se presente) e quindi alla variazione predefinita.
   - Successivamente, il traffico rimanente è disponibile per l'esperimento in corso, se ce n'è uno. Se c'è un esperimento in corso, la tabella nella sezione Esperimenti visualizza il nome dell'esperimento e lo stato di avanzamento. Sul lato destro della sezione Esperimenti, un indicatore di Traffico visualizza quanta parte dei destinatari disponibili (dopo le sostituzioni e i lanci) viene allocata a questo esperimento. Il resto del traffico non allocato all'avvio o all'esperimento viene servito con la variazione di default della funzionalità.

Modifica del traffico di avvio

È possibile modificare l'allocazione del traffico per un avvio in qualsiasi momento, anche mentre l'avvio è in corso.

Se si dispone di un avvio in corso e di un esperimento in corso per la stessa funzionalità, qualsiasi modifica del traffico delle funzionalità causerà una modifica del traffico dell'esperimento. Questo perché i destinatari disponibili per l'esperimento è la parte dei destinatari totali che non è già stata assegnata al lancio. L'aumento del traffico di lancio diminuirà i destinatari disponibili per l'esperimento

e la diminuzione del traffico di avvio o la fine dell'avvio aumenterà i destinatari disponibili per l'esperimento.

Per modificare l'allocazione del traffico per un avvio

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto contenente l'avvio.
4. Seleziona la scheda Avvio di.
5. Scegli il nome dell'avvio.

   Scegli Modifica del traffico di avvio.
6. Per Servi, seleziona la percentuale di traffico da assegnare a ciascuna variante. Puoi inoltre scegliere di escludere che le variazioni vengano inviate agli utenti. Man mano che si modificano questi valori, è possibile vedere gli effetti aggiornati sul traffico complessivo delle funzionalità in Riepilogo traffico.

   Il Riepilogo traffico mostra la quantità di traffico globale disponibile per questo avvio e quanto del traffico disponibile viene allocato a questo avvio.
7. Scegli Modify (Modifica).

Modificare i passaggi futuri di un avvio

È possibile modificare la configurazione delle fasi di avvio ancora non avvenute e aggiungere altri passaggi a un avvio.

Per modificare i passaggi per un avvio

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto contenente l'avvio.
4. Seleziona la scheda Avvio di.
5. Scegli il nome dell'avvio.

   Scegli Modifica del traffico di avvio.
6. Scegli Avvio della pianificazione.

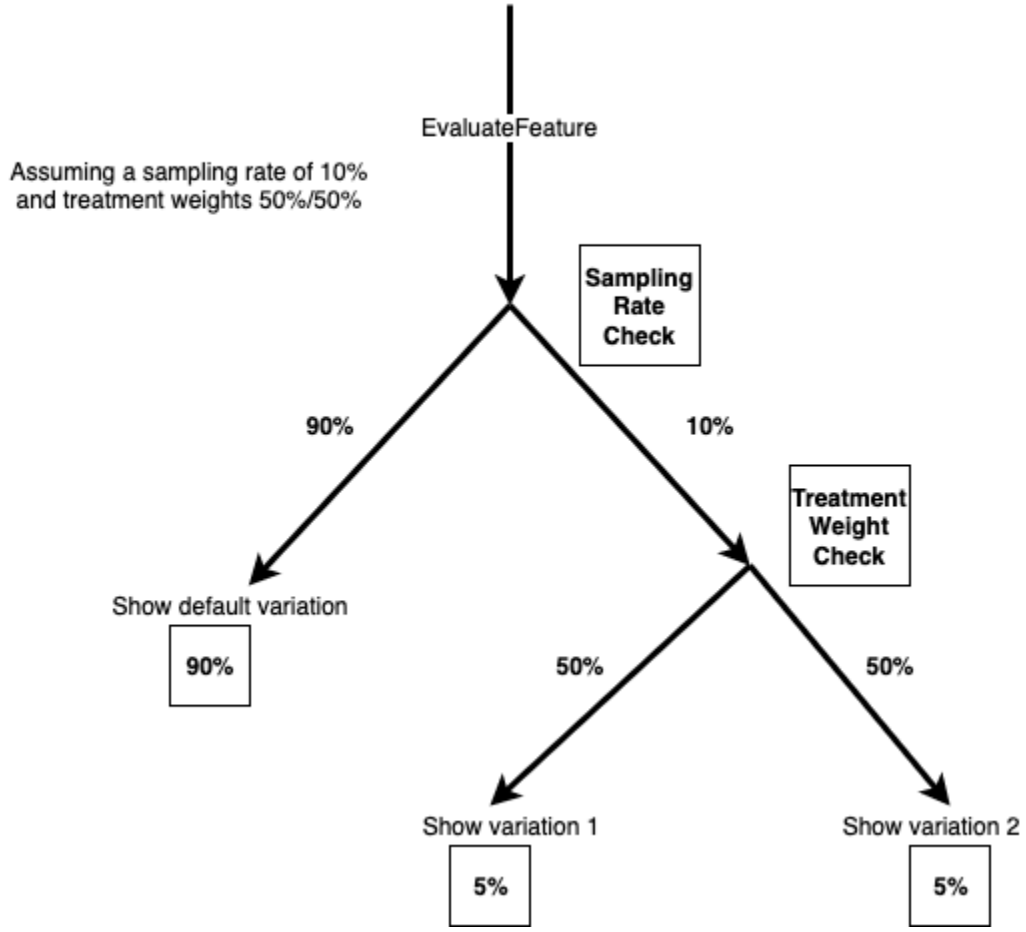
7. Per tutti i passaggi che non sono ancora iniziati, è possibile modificare la percentuale dei destinatari disponibili da utilizzare nell'esperimento. È inoltre possibile modificare il modo in cui il traffico viene allocato tra le varianti.

   È possibile aggiungere altri passaggi all'avvio scegliendoAggiungi di un'altra fase. Un avvio può avere un massimo di cinque passaggi.

8. Scegli Modifica.

Modifica del traffico sperimentale

È possibile modificare la frequenza di campionamento per un esperimento in qualsiasi momento, anche mentre l'esperimento è in corso. Tuttavia, non è possibile aggiornare i pesi del trattamento dopo l'esecuzione di un esperimento. Pertanto, è possibile modificare il traffico totale esposto all'esperimento dopo l'esecuzione di un esperimento, ma non l'allocazione relativa a ciascun trattamento. Se si modifica il traffico di un esperimento in corso, consigliamo di aumentare solo l'allocazione del traffico, in modo da non introdurre distorsioni.



Per modificare l'allocazione del traffico per un esperimento

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Monitoraggio applicazioni,Evidently.
3. Scegli il nome del progetto contenente l'avvio.
4. Seleziona la scheda Esperimenti.
5. Scegli il nome dell'avvio.
6. Scegli Modifica del traffico dell'esperimento.
7. Immetti una percentuale o utilizzare il dispositivo di scorrimento per specificare la quantità di traffico disponibile da allocare a questo esperimento. Il traffico disponibile indica i destinatari totali meno il traffico allocato a un avvio corrente, se ce n'è uno. Il traffico che non è allocato all'avvio all'esperimento viene servito con la variazione di default.
8. Scegli Modifica.

Interrompere un avvio

Se si interrompe un avvio in corso, non sarà possibile riprenderlo o riavviarlo. Inoltre, non verrà valutato come norma per l'allocazione del traffico e il traffico allocato all'avvio sarà invece disponibile per l'esperimento della funzionalità, se ce n'è uno. In caso contrario, tutto il traffico verrà servito con la variazione di default dopo l'arresto dell'avvio.

Per interrompere definitivamente un avvio

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto contenente l'avvio.
4. Scegli la scheda Avvio.
5. Scegli il pulsante a sinistra del nome dell'avvio.
6. Scegli Operazioni, Annulla avvio o Operazioni, Contrassegna come completo.

Interrompere un esperimento

Se si interrompe un esperimento in corso, non sarà possibile riprenderlo o riavviarlo. La parte di traffico precedentemente utilizzata nell'esperimento verrà servita con la variazione di default.

Quando un esperimento non viene arrestato manualmente e supera la data di fine, il traffico non cambia. La parte di traffico allocata all'esperimento va ancora all'esperimento. Per fermare questo e far sì che il traffico dell'esperimento venga invece servito con la variazione di default, contrassegnare l'esperimento come completo.

Quando si interrompe un esperimento, è possibile scegliere di annullarlo o contrassegnarlo come completo. Se viene annullato, verrà mostrato come Annullato nell'elenco degli esperimenti. Se si sceglie di contrassegnarlo come completo, viene visualizzato come Completato.

Per interrompere definitivamente un esperimento

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto contenente l'esperimento.
4. Seleziona la scheda Esperimenti.
5. Scegli il pulsante a sinistra del nome dell'esperimento.
6. Scegli Operazioni, Annullamento dell'esperimento o Operazioni, Contrassegna come completo.

Aggiungere un codice all'applicazione

Per lavorare con CloudWatch Evidently, devi aggiungere codice alla tua applicazione per assegnare una variante a ciascuna sessione utente e inviare metriche a Evidently. Utilizzate l'`EvaluateFeature`operazione CloudWatch Evidently per assegnare variazioni alle sessioni utente e utilizzate l'`PutProjectEvents`operazione per inviare eventi a Evidently da utilizzare per calcolare le metriche per i vostri lanci o esperimenti.

Quando crei varianti o metriche personalizzate, la console CloudWatch Evidently fornisce esempi del codice da aggiungere.

Per un end-to-end esempio, vedi. Esercitazione: test A/B con l'applicazione Evidently di esempio

Usando EvaluateFeature

Quando vengono utilizzate variazioni di funzionalità in un avvio o in un esperimento, l'applicazione utilizza l' EvaluateFeatureoperazione per assegnare una variante a ogni sessione utente. L'assegnazione di una variazione a un utente è un evento di valutazione. Quando si chiama questa operazione, si passa a quanto segue:

- Nome delle funzionalità: obbligatorio. Evidently elabora la valutazione in base alle regole di valutazione delle caratteristiche dell'avvio o dell'esperimento e seleziona una variante per l'entità.
- entityId: obbligatorio. Rappresenta un utente univoco.
- Contesto di valutazione— Opzionale. Un oggetto JSON che rappresenta informazioni aggiuntive su un utente. Evidentemente utilizzerà questo valore per abbinare l'utente a un segmento del tuo pubblico durante le valutazioni delle funzionalità, se hai creato segmenti. Per ulteriori informazioni, consulta Usa i segmenti per focalizzare il tuo pubblico.

  Segue un esempio di un valore `evaluationContext` che puoi inviare a Evidently.

```
{
    "Browser": "Chrome",
    "Location": {
        "Country": "United States",
        "Zipcode": 98007
    }
}
```

Valutazioni permanenti

CloudWatch Evidentemente utilizza valutazioni «appiccicose». Un'unica configurazione di `entityId`, funzionalità, configurazione delle funzionalità e `evaluationContext` riceve sempre la stessa assegnazione di variante. L'unica modifica delle assegnazioni delle variazioni temporali è quando un'entità viene aggiunta a un override o viene composto il traffico sperimentale.

Una configurazione delle funzionalità include:

- Le varianti delle funzionalità
- La configurazione delle varianti (percentuali assegnate a ciascuna variante) per un esperimento attualmente in corso per questa funzionalità, se presente.
- La configurazione della variante per un lancio attualmente in esecuzione per questa funzionalità, se presente. La configurazione della variante include le eventuali sostituzioni dei segmenti definite.

Se l'allocazione del traffico di un esperimento viene aumentata, qualsiasi `entityId` che era stato precedentemente assegnato a un gruppo di trattamento sperimentale continuerà a ricevere lo stesso trattamento. Qualsiasi `entityId` precedentemente assegnato al gruppo di controllo,

potrebbe essere assegnato a un gruppo di trattamento sperimentale, in base alla configurazione della variazione specificata per l'esperimento.

Se l'allocazione del traffico di un esperimento è ridotta, un `entityId` potrebbe passare da un gruppo di trattamento a un gruppo di controllo, ma non a un gruppo di trattamento diverso.

Usando PutProjectEvents

Per codificare una metrica personalizzata per Evidently, si utilizza l' PutProjectEventsoperazione. Di seguito è riportato un semplice esempio di payload.

```
{
    "events": [
        {
            "timestamp": {{$timestamp}},
            "type": "aws.evidently.custom",
            "data": "{\"details\": {\"pageLoadTime\": 800.0}, \"userDetails\":
 {\"userId\": \"test-user\"}}"
        }
    ]
}
```

La `entityIdKey` può essere solo un `entityId` oppure è possibile rinominarlo in qualsiasi altro modo, ad esempio `userId`. Nell'evento reale, `entityId` può essere un nome utente, un ID sessione e così via.

```
"metricDefinition":{
    "name": "noFilter",
    "entityIdKey": "userDetails.userId", //should be consistent with jsonValue in
 events "data" fields
    "valueKey": "details.pageLoadTime"
},
```

Per garantire che gli eventi siano associati all'avvio o all'esperimento corretto, devi passare lo stesso`entityId` quando chiami sia `EvaluateFeature` che `PutProjectEvents`. Assicurati di chiamare `PutProjectEvents` dopo la `EvaluateFeature` chiamata, altrimenti i dati verranno eliminati e non verranno utilizzati da CloudWatch Evidently.

L'operazione `PutProjectEvents` non richiede il nome della caratteristica come parametro di input. In questo modo, puoi utilizzare un unico evento in diversi esperimenti. Ad esempio, supponiamo che chiami `EvaluateFeature` con il `entityId` impostato su `userDetails.userId`. Se hai due o

più esperimenti in esecuzione, puoi fare in modo che un singolo evento della sessione di quell'utente emetta parametri per ciascuno di questi esperimenti. Per fare questo, chiami `PutProjectEvents` una volta per ogni esperimento, usando lo stesso `entityId`.

Timing (Tempo)

Dopo che l'applicazione chiama `EvaluateFeature`, c'è un periodo di tempo di un'ora in cui gli eventi dei parametri da `PutProjectEvents` sono attribuiti basati su tale valutazione. Se si verificano altri eventi dopo il periodo di un'ora, non vengono attribuiti.

Tuttavia, se lo stesso `entityId` è usato per una nuova chiamata `EvaluateFeature` durante la finestra di un'ora di quella chiamata iniziale, viene utilizzato il risultato della successiva `EvaluateFeature` e il timer di un'ora viene riavviato. Ciò può accadere solo in determinate circostanze, ad esempio quando il traffico sperimentale viene composto tra i due incarichi, come spiegato nella sezione precedente Valutazioni permanenti.

Per un end-to-end esempio, vediEsercitazione: test A/B con l'applicazione Evidently di esempio.

Archiviazione dati di progetto

Evidently raccoglie due tipi di eventi:

- Eventi di valutazione sono correlati a quale variazione di funzionalità viene assegnata a una sessione utente. Evidently utilizza questi eventi per produrre parametri e altri dati sperimentali e di avvio, che è possibile visualizzare nella console Evidently.

  Puoi anche scegliere di archiviare questi eventi di valutazione in Amazon CloudWatch Logs o Amazon S3.
- Eventi personalizzati sono utilizzati per generare parametri dalle azioni dell'utente ad esempio click e checkout. Evidently non fornisce un metodo per archiviare eventi personalizzati. Se si desidera salvarli, è necessario modificare il codice dell'applicazione per inviarlo a un'opzione di archiviazione al di fuori di Evidently.

Formato dei registri eventi di valutazione

Se scegli di archiviare gli eventi di valutazione in CloudWatch Logs o Amazon S3, ogni evento di valutazione viene archiviato come evento di registro con il seguente formato:

```
{
    "event_timestamp": 1642624900215,
```

```
    "event_type": "evaluation",
    "version": "1.0.0",
    "project_arn": "arn:aws:evidently:us-east-1:123456789012:project/petfood",
    "feature": "petfood-upsell-text",
    "variation": "Variation1",
    "entity_id": "7",
    "entity_attributes": {},
    "evaluation_type": "EXPERIMENT_RULE_MATCH",
    "treatment": "Variation1",
    "experiment": "petfood-experiment-2"
}
```

Di seguito sono riportati ulteriori dettagli sul precedente formato dell'evento di valutazione:

- Il timestamp è in tempo UNIX con millisecondi
- La variante è il nome della variante della funzione assegnata a questa sessione utente.
- L'ID entità è una stringa.
- Gli attributi di entità sono un hash di valori arbitrari inviati dal client. Ad esempio, se `entityId` è mappato in blu o verde, allora è possibile inviare UserID, dati di sessione o qualsiasi altra cosa desideri dal punto di vista della correlazione e del data warehouse.

Crittografia e policy IAM per lo storage degli eventi di valutazione in Amazon S3

Se si sceglie di utilizzare Amazon S3 per memorizzare eventi di valutazione, devi aggiungere una policy IAM come la seguente per consentire a Evidently di pubblicare i log nel bucket Amazon S3. Questo perché i bucket Amazon S3 e gli oggetti che contengono sono privati e non consentono l'accesso ad altri servizi per impostazione predefinita.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/
*",
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-
control"}}
```

```
        },
        {
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": ["s3:GetBucketAcl", "s3:ListBucket"],
            "Resource": "arn:aws:s3:::bucket_name"
        }
    ]
}
```

Se si memorizzano dati Evidently in Amazon S3, è anche possibile scegliere di crittografarli con Chiavi per la crittografia lato server (SSE-KMS) AWS Key Management Service . Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server.

Se utilizzi una chiave gestita dal cliente di AWS KMS, devi aggiungere quanto segue alla policy IAM relativa alla tua chiave. Ciò consente a Evidently di scrivere nel bucket.

```
{
    "Sid": "AllowEvidentlyToUseCustomerManagedKey",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
    },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

In che modo Evidently calcola i risultati

Puoi utilizzare Amazon CloudWatch Evidently A/B testing come strumento per prendere decisioni basate sui dati. In un test A/B, gli utenti vengono assegnati in modo casuale al gruppo di controllo (chiamato anche variante predefinita) o a uno dei gruppi di trattamento (chiamato anche variante

testata). Ad esempio, gli utenti del gruppo di controllo potrebbero utilizzare il sito Web, il servizio o l'applicazione nello stesso modo in cui lo facevano prima dell'inizio dell'esperimento. Al contempo, gli utenti del gruppo di trattamento potrebbero notare un cambiamento.

CloudWatch Evidentemente supporta fino a cinque diverse varianti in un esperimento. Evidently assegna il traffico a queste varianti in modo casuale. In questo modo, puoi tenere traccia dei parametri aziendali (ad esempio, il fatturato) e delle prestazioni (ad esempio, la latenza) per ogni gruppo. Evidently esegue le operazioni seguenti:

- Confronta il trattamento con il controllo. Ad esempio, puoi verificare se il fatturato aumenta o diminuisce con una nuova procedura di checkout.
- Indica se la differenza osservata tra il trattamento e il controllo è significativa. A tal fine, Evidently offre due approcci: livelli di significatività frequentista e probabilità bayesiane.

Perché usare gli approcci frequentisti e bayesiani?

Immagina come esempio un caso in cui il trattamento non ha alcun effetto sul controllo o un esempio in cui il trattamento è identico al controllo (ad esempio, un test A/A). Osserveresti in ogni caso una piccola differenza tra i dati del trattamento e quelli del controllo. Questo perché i partecipanti al test sono costituiti da un campione finito di utenti, che rappresenta una piccola percentuale di tutti gli utenti del sito Web, del servizio o dell'applicazione. I livelli di significatività frequentista e le probabilità bayesiane consentono di comprendere, tramite informazioni dettagliate, se la differenza osservata è significativa o dovuta al caso.

Per determinare se la differenza osservata è significativa o meno, Evidently prende in considerazione gli elementi seguenti:

- L'entità della differenza
- Il numero di campioni che fanno parte del test
- Il modo in cui vengono distribuiti i dati

Analisi frequentista in Evidently

Evidently utilizza i test sequenziali in modo da evitare le problematiche legate al peeking, un errore comune delle statistiche frequentiste. Con peeking si intende la pratica di controllare i risultati di un test A/B in corso per interromperlo e prendere una decisione in base ai risultati osservati. Per ulteriori informazioni sui test sequenziali, consulta l'articolo Time-uniform, nonparametric, nonasymptotic

confidence sequences (Sequenze di intervalli di confidenza non asintotici, non parametrici e uniformi nel tempo) di Howard e altri autori. (Ann. Statist. 49 (2) 1055 - 1080, 2021).

Dal momento che i risultati di Evidently sono validi in qualsiasi momento (risultati validi in qualsiasi momento), puoi visualizzarli durante l'esperimento e trarre comunque conclusioni valide. Ciò consente di ridurre alcuni costi correlati alla sperimentazione, dal momento che è possibile interrompere un esperimento prima della scadenza prevista se i risultati sono già significativi.

Evidently genera livelli di significatività validi in qualsiasi momento e intervalli di confidenza del 95% validi in qualsiasi momento per la differenza tra la variante testata e la variante predefinita nel parametro target. La colonna Result (Risultato) dell'esperimento indica le prestazioni della variante testata e può presentare uno dei valori seguenti:

- Inconclusive (Inconcludente): il livello di significatività è inferiore al 95%
- Better (Meglio): il livello di significatività è pari o superiore al 95% e una delle seguenti condizioni è vera:
  - Il limite inferiore dell'intervallo di confidenza del 95% è superiore a zero e il parametro dovrebbe aumentare
  - Il limite superiore dell'intervallo di confidenza del 95% è inferiore a zero e il parametro dovrebbe diminuire
- Worse (Peggio): il livello di significatività è pari o superiore al 95% e una delle seguenti condizioni è vera:
  - Il limite superiore dell'intervallo di confidenza del 95% è superiore a zero e il parametro dovrebbe aumentare
  - Il limite inferiore dell'intervallo di confidenza del 95% è inferiore a zero e il parametro dovrebbe diminuire
- Best (Migliore): l'esperimento presenta due o più varianti testate oltre alla variante predefinita e le condizioni seguenti sono soddisfatte:
  - La variante è idonea per la designazione Better (Meglio)
  - Una delle condizioni seguenti è vera:
    - Il limite inferiore dell'intervallo di confidenza del 95% è maggiore del limite superiore degli intervalli di confidenza al 95% di tutte le altre varianti e il parametro dovrebbe aumentare
    - Il limite superiore dell'intervallo di confidenza del 95% è minore del limite inferiore degli intervalli di confidenza al 95% di tutte le altre varianti e il parametro dovrebbe diminuire

Analisi bayesiana in Evidently

Con l'analisi bayesiana puoi calcolare la probabilità che la media nella variante testata sia maggiore o minore della media nella variante predefinita. Evidently esegue l'inferenza bayesiana per la media del parametro target utilizzando i priori coniugati. Grazie ai priori coniugati, Evidently può dedurre in modo più efficiente la distribuzione posteriore necessaria per l'analisi bayesiana.

Evidently attende la data di fine dell'esperimento per calcolare i risultati dell'analisi bayesiana. La pagina dei risultati mostra quanto segue:

- probabilità di aumento: la probabilità che la media del parametro nella variante testata sia superiore di almeno il 3% rispetto alla media nella variante predefinita
- probabilità di diminuzione: la probabilità che la media del parametro nella variante testata sia inferiore di almeno il 3% rispetto alla media nella variante predefinita
- probabilità di invarianza: la probabilità che la media del parametro nella variante testata si trovi entro il ±3% della media della variante predefinita

La colonna Result (Risultato) indica le prestazioni della variante e può presentare uno dei valori seguenti:

- Better (Meglio): la probabilità di aumento è almeno del 90% e il parametro dovrebbe aumentare oppure la probabilità di diminuzione è almeno del 90% e il parametro dovrebbe diminuire
- Worse (Peggio): la probabilità di diminuzione è almeno del 90% e il parametro dovrebbe aumentare oppure la probabilità di aumento è almeno del 90% e il parametro dovrebbe diminuire

Visualizzare i risultati di avvio nel pannello di controllo

È possibile visualizzare i risultati di avanzamento e dei parametri di un esperimento mentre è in corso e dopo il completamento.

Per vedere i progressi e i risultati di un avvio

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto contenente l'avvio.
4. Scegli la scheda Avvio.
5. Scegli il nome dell'avvio.

6. Per vedere le fasi di avvio e le allocazioni del traffico per ogni passaggio, scegli la scheda Avvio.
7. Per visualizzare il numero di sessioni utente assegnate a ciascuna variazione nel tempo e per visualizzare le metriche delle prestazioni per ogni variante nell'avvio, scegli la scheda Monitoraggio.

   Questa vista mostra anche se sono alcuni allarmi di avvio sono andati in stato `ALARM` durante l'avvio.
8. Per vedere le varianti, i parametri, gli allarmi e i tag per questo avvio, scegli la scheda Configurazione.

Visualizzare i risultati degli esperimenti nel pannello di controllo

È possibile visualizzare i risultati statistici di un esperimento mentre è in corso e dopo il completamento. I risultati dell'esperimento sono disponibili fino a 63 giorni dopo l'inizio dell'esperimento. Successivamente non sono disponibili a causa delle politiche di conservazione CloudWatch dei dati.

Nessun risultato statistico viene visualizzato fino a quando ogni variante ha almeno 100 eventi.

Evidently esegue un'ulteriore analisi offline del valore p alla fine dell'esperimento. L'analisi del valore p offline può rilevare la significatività statistica in alcuni casi in cui i valori p utilizzati durante l'esperimento non trovano significatività statistica.

Per ulteriori informazioni su come CloudWatch Evidently calcola i risultati degli esperimenti, vedere. In che modo Evidently calcola i risultati

Per visualizzare i risultati di un esperimento

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il nome del progetto contenente l'esperimento.
4. Seleziona la scheda Esperimenti.
5. Scegli il nome dell'esperimento, quindi scegli la scheda Risultati.
6. Vicino a Variation performance, c'è un controllo in cui è possibile selezionare le statistiche degli esperimenti da visualizzare. Se si seleziona più di una statistica, Evidently visualizza un grafico e una tabella per ogni statistica.

   Ciascun grafico e table mostrano i risultati dell'esperimento fino ad ora.

Ciascun grafico può visualizzare i seguenti risultati. È possibile utilizzare il controllo a destra del grafico per determinare quale dei seguenti elementi è visualizzato:

- Numero di eventi di sessione utente registrati per ogni variante.
- Il valore medio del parametro selezionato nella parte superiore del grafico, per ogni variante.
- Il significato statistico degli esperimenti. Ciò confronta la differenza per il parametro selezionato nella parte superiore del grafico con la variante di default e ciascuna delle altre varianti.
- I limiti di confidenza superiore e inferiore del 95% corrispondono alla differenza del parametro selezionato, tra ciascuna variante e la variante di default.

La tabella mostra una riga per ogni variante. Per ogni variante che non è quella di default, Evidently mostra se ha ricevuto dati sufficienti per dichiarare i risultati statisticamente significativi. Mostra anche se il miglioramento della variazione del valore statistico ha raggiunto un livello di sicurezza del 95%.

Infine, nella colonna Risultato, Evidently fornisce un suggerimento su quale variazione funziona meglio in base a questa statistica, o se i risultati sono inconclusivi.

How CloudWatch Evidently raccoglie e archivia i dati

Amazon CloudWatch evidentemente raccoglie e archivia i dati relativi alle configurazioni dei progetti in modo che i clienti possano eseguire esperimenti e lanci. I dati includono quanto segue:

- Metadati su progetti, funzionalità, avvii ed esperimenti
- Eventi dei parametri
- Dati di valutazione

I metadati delle risorse sono archiviati in Amazon DynamoDB. Per impostazione predefinita, i dati sono crittografati quando sono inattivi, utilizzando. Chiavi di proprietà di AWS Queste chiavi sono una raccolta di AWS KMS chiavi Servizio AWS possedute e gestite per l'utilizzo in più lingue Account AWS. I clienti non possono visualizzare, gestire o verificare l'uso di queste chiavi. Inoltre, i clienti non sono tenuti a intervenire o modificare programmi per proteggere le chiavi che eseguono la crittografia dei dati.

Per ulteriori informazioni, consulta Chiavi di proprietà di AWSla Guida per AWS Key Management Service gli sviluppatori.

Gli eventi dei parametr e gli eventi di valutazione di Evidently vengono consegnati direttamente alle sedi di proprietà del cliente.

I dati in transito vengono crittografati automaticamente con HTTPS. Questi dati verranno consegnati alle sedi di proprietà del cliente.

Puoi anche scegliere di archiviare gli eventi di valutazione in Amazon Simple Storage Service o Amazon CloudWatch Logs. Per ulteriori informazioni su come proteggere i dati in questi servizi, consulta Abilitazione della crittografia dei bucket predefinita di Amazon S3 e Crittografia dei dati di log in Logs using. CloudWatch AWS KMS

Recupero dei dati

Puoi recuperare i tuoi dati utilizzando le API Evidently. CloudWatch Per recuperare i dati del progetto, usa o. GetProjectListProjects

Per recuperare i dati delle feature, utilizzate GetFeatureo. ListFeatures

Per recuperare i dati di lancio, usa GetLauncho. ListLaunches

Per recuperare i dati dell'esperimento, usa GetExperimentListExperiments, o. GetExperimentResults

Modifica ed eliminazione di dati

Puoi modificare ed eliminare i tuoi dati utilizzando le API CloudWatch Evidently. Per i dati del progetto, usa UpdateProjecto. DeleteProject

Per i dati sulle funzionalità, usa UpdateFeatureo DeleteFeature.

Per i dati di lancio, usa UpdateLauncho DeleteLaunch.

Per i dati dell'esperimento, usa UpdateExperimento DeleteExperiment.

Utilizzo di ruoli collegati ai servizi per Evidently

CloudWatch Evidentemente utilizza ruoli AWS Identity and Access Management collegati ai servizi (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente a Evidently.

I ruoli collegati ai servizi sono predefiniti da Evidently e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione di Evidently perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. Evidently definisce le autorizzazioni dei relativi ruoli associati ai servizi e, salvo diversamente definito, solo Evidently potrà assumere i propri ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di Evidently perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Yes (Sì) nella colonna Service-linked roles (Ruoli collegati ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per Evidently

Evidently utilizza il ruolo collegato al servizio denominato AWSServiceRoleForCloudWatchEvidently— Permette a CloudWatch Evidently di gestire le risorse associate per conto del cliente. AWS

Il ruolo AWSServiceRoleForCloudWatchEvidently collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `CloudWatch Evidently`

La politica di autorizzazione dei ruoli denominata AmazonCloudWatchEvidentlyServiceRolePolicy consente Evidently di completare le seguenti azioni sulle risorse specificate:

- Operazioni: `appconfig:StartDeployment`, `appconfig:StopDeployment`, `appconfig:ListDeployments` e `appconfig:TagResource` su thick client di Evidently.

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per Evidently

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando inizi a utilizzare un client Evidently thick nella AWS Management Console, nella o nell' AWS API AWS CLI, Evidenently crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato ai servizi, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando inizi a utilizzare un thick client di Evidently, quest'ultimo crea nuovamente il ruolo collegato ai servizi per tuo conto.

Modifica di un ruolo collegato ai servizi per Evidently

Evidentemente non ti consente di modificare il ruolo collegato al servizio. AWSServiceRoleForCloudWatchEvidently Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Evidently

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente. È necessario eliminare tutti i progetti Evidently che utilizzano thick client.

> **Note**
>
> Se il servizio Evidently utilizza tale ruolo quando tenti di eliminare le risorse, è possibile che l'eliminazione non abbia esito positivo. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare Evidently le risorse utilizzate da AWSServiceRoleForCloudWatchEvidently

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Monitoraggio applicazioni,Evidently.
3. Nell'elenco dei progetti, seleziona la casella di controllo accanto ai progetti che hanno utilizzato thick client.

4. Scegli Project actions (Operazioni di progetto), Delete project (Elimina progetto).

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo AWSServiceRoleForCloudWatchEvidently collegato al servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi Evidently

Evidently supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta Regioni ed endpoint di AWS.

CloudWatch Evidentemente quote

CloudWatch Evidentemente ha le seguenti quote.

Risorsa	Quota predefinita
Progetti	50 per regione per account È possibile richiedere un aumento della quota.
Segmenti	500 per regione per account È possibile richiedere un aumento della quota.
Quote per progetto	• 100 funzionalità totali • 500 avvii totali • 50 avvi in corso • 500 esperimenti totali • 50 esperimenti in corso Puoi richiedere un aumento di quota per tutte queste quote.
Quote API (tutte le quote si intendono per regione)	• PutProjectEvents: 1000 transazioni al secondo (TPS) negli Stati Uniti orientali (Virginia settentrionale), negli

Risorsa	Quota predefinita
	Stati Uniti occidentali (Oregon) e in Europa (Irlanda). 200 TPS in tutte le altre regioni • EvaluateFeature: 1000 TPS negli Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon) ed Europa (Irlanda). 200 TPS in tutte le altre regioni • BatchEvaluateFeature: 50 TPS • Creazione, lettura, aggiornamento, eliminazione API (CRUD): 10 TPS combinati in tutte le API CRUD Puoi richiedere un aumento di quota per tutte queste quote.

Esercitazione: test A/B con l'applicazione Evidently di esempio

Questa sezione fornisce un tutorial per l'utilizzo di Amazon CloudWatch Evidently per i test A/B. Questo tutorial ha come esempio l'applicazione Evidently, che è una semplice applicazione di reazione. L'applicazione di esempio sarà configurata per visualizzare o meno una funzione `showDiscount`. Quando la funzione viene mostrata a un utente, il prezzo visualizzato sul sito web dello shopping è mostrato con uno sconto del 20%.

Oltre a mostrare lo sconto ad alcuni utenti e non ad altri, in questo tutorial hai impostato Evidently per raccogliere i parametri del tempo di caricamento della pagina da entrambe le varianti.

> **Warning**
>
> Questo scenario richiede agli utenti IAM accesso programmatico e credenziali a lungo termine, il che presenta un rischio per la sicurezza. Per contribuire a mitigare questo rischio, ti consigliamo di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari. Le chiavi di accesso possono essere aggiornate se necessario. Per ulteriori informazioni, consulta Updating access keys nella IAM User Guide.

Passaggio 1: Download dell'applicazione di esempio

Inizia scaricando l'applicazione Evidently di esempio.

Per scaricare l'applicazione di esempio

1. scarica l'applicazione di esempio dal seguente bucket Amazon S3:

   ```
   https://evidently-sample-application.s3.us-west-2.amazonaws.com/evidently-sample-shopping-app.zip
   ```

2. Decomprimi il pacchetto.

Passaggio 2: Aggiungere l'endpoint Evidently e configurazione delle credenziali

Quindi, aggiungi la Regione e l'endpoint per Evidently al file`config.js` nella directory `src` nel pacchetto di app di esempio, come nell'esempio seguente:

```
evidently: {
    REGION: "us-west-2",
    ENDPOINT: "https://evidently.us-west-2.amazonaws.com (https://evidently.us-west-2.amazonaws.com/)",
},
```

È inoltre necessario assicurarsi che l'applicazione sia autorizzata a chiamare CloudWatch Evidently.

Per concedere all'app di esempio le autorizzazioni per chiamare Evidently

1. Effettua la federazione sul tuo account. AWS
2. Crea un utente IAM e allega la AmazonCloudWatchEvidentlyFullAccesspolicy a questo utente.
3. Prendere nota dell'ID della chiave di accesso e della chiave di accesso segreta dell'utente IAM poiché sarà necessario nella fase successiva.
4. Nello stesso file `config.js` modificato in precedenza in questa sezione, immetti i valori dell'ID chiave di accesso e la chiave di accesso segreta, come nell'esempio seguente:

   ```
   credential: {
      accessKeyId: "Access key ID",
      secretAccessKey: "Secret key"
   }
   ```

> **Important**
>
> Utilizziamo questo passaggio per rendere l'app di esempio il più semplice possibile da provare. Non è consigliabile inserire le credenziali utente IAM nella tua applicazione di produzione. Invece, ti consigliamo di utilizzare Amazon Cognito per l'autenticazione. Per maggiori informazioni, consulta Integrazione di Amazon Cognito con le app Web e per dispositivi mobili.

Passaggio 3: Impostazione del codice per la valutazione delle funzionalità

Quando usi CloudWatch Evidently per valutare una funzionalità, devi utilizzare l'EvaluateFeatureoperazione per selezionare casualmente una variante della funzionalità per ogni sessione utente. Questa operazione assegna sessioni utente a ciascuna variante della funzionalità, in base alle percentuali specificate nell'esperimento.

Per impostare il codice di valutazione delle funzionalità per l'app demo della libreria

1. Aggiungi il client builder nel file `src/App.jsx` in modo che l'app di esempio possa chiamare Evidently.

```
import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
    endpoint,
    region,
) => {
    const credentials = {
        accessKeyId: config.credential.accessKeyId,
        secretAccessKey: config.credential.secretAccessKey
    }
    return new Evidently({
        endpoint,
        region,
        credentials,
    });
};
```

2. Aggiungi il comando seguente nella sezione codice `const` App per avviare il client.

```
  if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
```

3. Costruisci `evaluateFeatureRequest` aggiungendo il seguente codice. Questo codice pre-riempie il nome del progetto e della funzionalità che consigliamo più avanti in questo tutorial. È possibile sostituire i nomi dei progetti e delle funzionalità, purché si specifichino anche i nomi dei progetti e delle funzionalità nella console Evidently.

```
const evaluateFeatureRequest = {
              entityId: id,
              // Input Your feature name
              feature: 'showDiscount',
              // Input Your project name'
              project: 'EvidentlySampleApp',
            };
```

4. Aggiungi il codice da chiamare Evidently per la valutazione delle funzionalità. Quando la richiesta viene inviata, Evidently assegna in modo casuale la sessione utente per vedere o meno la funzionalità `showDiscount`.

```
client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
    if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
    }
    getPageLoadTime()
 })
```

Passaggio 4: Configura il codice per i parametri dell'esperimento

Per il parametro personalizzato, usare l'API `PutProjectEvents` di Evidently per inviare i risultati deli parametri a Evidently. I seguenti esempi mostrano come configurare il parametro personalizzato e inviare dati sperimentali a Evidently.

Quindi, aggiungere la seguente funzione per calcolare il tempo di caricamento della pagina e utilizzare `PutProjectEvents` per inviare i valori dei parametri a Evidently. Inserire il seguente codice in `Home.tsx` e chiamare questa funzione all'interno dell'API di `EvaluateFeature`:

```
const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
```

Ecco come il file `App.js` dovrebbe apparire dopo le modifiche che hai apportato da quando lo hai scaricato.

```
import React, { useEffect, useState } from "react";
import { BrowserRouter as Router, Switch } from "react-router-dom";
import AuthProvider from "contexts/auth";
import CommonProvider from "contexts/common";
import ProductsProvider from "contexts/products";
import CartProvider from "contexts/cart";
import CheckoutProvider from "contexts/checkout";
import RouteWrapper from "layouts/RouteWrapper";
import AuthLayout from "layouts/AuthLayout";
import CommonLayout from "layouts/CommonLayout";
import AuthPage from "pages/auth";
import HomePage from "pages/home";
import CheckoutPage from "pages/checkout";
import "assets/scss/style.scss";
import { Spinner } from 'react-bootstrap';

import Evidently from 'aws-sdk/clients/evidently';
import config from './config';
```

```
const defaultClientBuilder = (
  endpoint,
  region,
) => {
  const credentials = {
    accessKeyId: config.credential.accessKeyId,
    secretAccessKey: config.credential.secretAccessKey
  }
  return new Evidently({
    endpoint,
    region,
    credentials,
  });
};

const App = () => {
  const [isLoading, setIsLoading] = useState(true);
  const [startTime, setStartTime] = useState(new Date());
  const [showDiscount, setShowDiscount] = useState(false);
  let client = null;
  let id = null;

  useEffect(() => {
    id = new Date().getTime().toString();
    setStartTime(new Date());
    if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
    }
    const evaluateFeatureRequest = {
      entityId: id,
      // Input Your feature name
      feature: 'showDiscount',
      // Input Your project name'
      project: 'EvidentlySampleApp',
    };

    // Launch
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
```

```
    });

    // Experiment
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
      getPageLoadTime()
    })

    setIsLoading(false);
  },[]);

  const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
  return (
    !isLoading? (
    <AuthProvider>
      <CommonProvider>
        <ProductsProvider>
          <CartProvider>
            <CheckoutProvider>
              <Router>
                <Switch>
                  <RouteWrapper
                    path="/"
```

```
                    exact
                    component={() => <HomePage showDiscount={showDiscount}/>}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/checkout"
                    component={CheckoutPage}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/auth"
                    component={AuthPage}
                    layout={AuthLayout}
                  />
                </Switch>
              </Router>
            </CheckoutProvider>
          </CartProvider>
        </ProductsProvider>
      </CommonProvider>
    </AuthProvider> ) : (
      <Spinner animation="border" />
    )
  );
};

export default App;
```

Ogni volta che un utente visita l'app bookstore, viene inviato un parametro personalizzato a Evidently per essere analizzato. Evidently analizza ogni parametro e visualizza i risultati in tempo reale sul pannello di controllo Evidently. Il seguente esempio mostra un payload di parametro:

```
[ {"timestamp": 1637368646.468, "type": "aws.evidently.custom", "data": "{\"details
\":{\"pageLoadTime\":2058.002058},\"userDetails\":{\"userId\":\"1637368644430\",
\"sessionId\":\"1637368644430\"}}" } ]
```

Passaggio 5: Creare il progetto, la funzione e l'esperimento

Successivamente, create il progetto, la funzionalità e l'esperimento nella console Evidently. CloudWatch

Per creare il progetto, la funzionalità e l'esperimento per questo tutorial

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli Crea progetto e compilare i campi. È necessario utilizzare **`EvidentlySampleApp`** per il corretto funzionamento del nome del progetto per l'esempio. Per Archiviazione eventi di valutazione, scegli Non memorizzare eventi di valutazione.

   Dopo aver riempito i campi, scegli Create profile (Crea profilo).

   Per ulteriori dettagli, consulta Crea un nuovo progetto.
4. Dopo la creazione del progetto, creare una funzionalità in quel progetto. Assegnare un nome alla funzionalità **`showDiscount`**. In questa funzionalità, creare due varianti del tipo **`Boolean`**. Assegnare un nome alla prima variante **`disable`** con un valore di **`False`** e nominare la seconda variante **`enable`** con un valore di **`True`**.

   Per ulteriori informazioni sulla creazione di una funzionalità, consulta Aggiunta di una funzionalità a un progetto.
5. Dopo aver finito di creare la funzionalità, creare un esperimento nel progetto. Assegnare un nome all'esperimento **`pageLoadTime`**.

   Questo esperimento utilizzerà un parametro personalizzato chiamato `pageLoadTime` che misura il tempo di caricamento della pagina da testare. Le metriche personalizzate per gli esperimenti vengono create utilizzando Amazon EventBridge. Per ulteriori informazioni su EventBridge, consulta What Is Amazon EventBridge? .

   Per creare quel parametro personalizzato, effettuare le seguenti operazioni quando si crea l'esperimento:

   - In Parametri, per Fonte parametro, scegli Parametri personalizzati.
   - Per Nome parametro, inserire **`pageLoadTime`**.
   - Per Obiettivo scegli Decrease (Riduci). Ciò indica che vogliamo che un valore inferiore di questo parametro indichi la migliore variazione della funzionalità.
   - Per Metric rule (Regola parametro), inserisci quanto segue.
     - Per Entity ID (ID entità), inserisci **`UserDetails.userId`**.
     - PerChiave valore, inserire **`details.pageLoadTime`**.
     - Per Units (Unità), inserisci **`ms`**.

- Scegli Aggiungi parametro.

Per Destinatari, seleziona 100%in modo che tutti gli utenti vengano inseriti nell'esperimento. Impostare la suddivisione del traffico tra le variazioni al 50% ciascuna.

Quindi scegli Create Experiment (Crea esperimento). Dopo averlo creato, non avvia finché non si dice a Evidently di avviarlo.

Fase 6: Iniziate l'esperimento e testatelo CloudWatch evidentemente

Gli ultimi passaggi sono l'avvio dell'esperimento e l'avvio dell'app di esempio.

Per iniziare l'esperimento di tutorial

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, Evidently.
3. Scegli il EvidentlySampleAppprogetto.
4. Seleziona la scheda Esperimenti.
5. Scegli il pulsante accanto a pageLoadTimee scegli Azioni, Avvia esperimento.
6. Scegli un orario per la fine dell'esperimento.
7. Scegli Inizia un esperimento.

   L'esperimento inizierà immediatamente.

Successivamente, avvia l'app Evidently di esempio con il comando seguente:

```
npm install -f && npm start
```

Una volta avviata l'app, ti verrà assegnata una delle due varianti in fase di test. Una variante mostra "20% di sconto» e l'altra no. Continua ad aggiornare la pagina per vedere le diverse varianti.

> **Note**
>
> Evidently ha valutazioni adesive. Le valutazioni delle funzionalità sono deterministiche, il che significa che per gli stessi `entityId` e funzionalità, un utente riceverà sempre la stessa

assegnazione di variazione. L'unica modifica delle assegnazioni delle variazioni temporali è quando un'entità viene aggiunta a un override o viene composto il traffico sperimentale. Tuttavia, per semplificare l'utilizzo del tutorial dell'app di esempio, Evidently riassegna la valutazione della funzionalità dell'app di esempio ogni volta che si aggiorna la pagina, in modo da poter sperimentare entrambe le varianti senza dover aggiungere sostituzioni.

Risoluzione dei problemi

Ti consigliamo di utilizzare la versione 6.14.14 di `npm`. Se riscontri errori relativi alla creazione o all'avvio dell'app di esempio e stai utilizzando una versione diversa di npm, procedi come segue.

Per installare la versione 6.14.14 di **`npm`**

1. Usa un browser per connetterti a https://nodejs.org/download/release/v14.17.5/.
2. Scarica nodo v14.17.5.pkg ed esegui questo pkg per installare npm.

   Se viene visualizzato un errore `webpack not found`, passa alla cartella `evidently-sample-shopping-app` e prova quanto segue:

   a. Eliminazione di `package-lock.json`
   b. Eliminazione di `yarn-lock.json`
   c. Eliminazione di `node_modules`
   d. Eliminare la dipendenza del webpack da `package.json`
   e. Esegui il seguente codice:

   ```
   npm install -f && npm
   ```

Usa CloudWatch RUM

Con CloudWatch RUM, è possibile eseguire il monitoraggio reale degli utenti per raccogliere e visualizzare dati lato client sulle prestazioni delle applicazioni Web provenienti da sessioni utente effettive quasi in tempo reale. I dati che è possibile visualizzare e analizzare includono tempo di caricamento delle pagine, errori lato client e comportamento dell'utente. Quando si visualizzano questi dati, è possibile vederli tutti aggregati insieme e vedere anche i guasti dei browser e dei dispositivi utilizzati dai clienti.

È possibile utilizzare i dati raccolti per identificare ed eseguire rapidamente il debug dei problemi di prestazioni sul lato client. CloudWatch RUM consente di visualizzare le anomalie nelle prestazioni delle applicazioni e di trovare dati di debug pertinenti come messaggi di errore, tracce dello stack e sessioni utente. È inoltre possibile utilizzare RUM per comprendere l'intervallo di impatto dell'utente finale, incluso il numero di utenti, la geolocalizzazione e i browser utilizzati.

I dati degli utenti finali raccolti per CloudWatch RUM vengono conservati per 30 giorni e quindi eliminati automaticamente. Se desideri conservare gli eventi RUM per un periodo più lungo, puoi scegliere di far sì che l'app Monitor invii copie degli eventi ai CloudWatch registri del tuo account. Quindi, è possibile modificare il periodo di conservazione per quel gruppo di log.

Per usare RUM, si crea un app monitor e si forniscono alcune informazioni. RUM genera uno JavaScript snippet da incollare nell'applicazione. Lo snippet inserisce il codice del client Web RUM. Il client Web RUM acquisisce i dati da una percentuale delle sessioni utente dell'applicazione, che viene visualizzata in un pannello di controllo precompilato. È possibile specificare la percentuale di sessioni utente da cui raccogliere i dati.

CloudWatch RUM è integrato con Application Signals, che può scoprire e monitorare i servizi applicativi, i client, i canali Synthetics e le dipendenze dei servizi. Utilizza Application Signals per visualizzare un elenco o una mappa visiva dei tuoi servizi, visualizzare i parametri di integrità in base agli obiettivi del livello di servizio (SLO) e approfondire le tracce X-Ray correlate per una risoluzione dei problemi più dettagliata. Per visualizzare le richieste delle pagine client RUM in Application Signals, attiva il tracciamento attivo X-Ray creando un monitor dell'app o configurando manualmente il client Web RUM. I tuoi client RUM vengono visualizzati nella mappa del servizio connessa ai tuoi servizi e nella pagina dei dettagli dei servizi che chiamano.

Il client Web RUM è open source. Per ulteriori informazioni, consulta CloudWatch il client web RUM.

Considerazioni sulle prestazioni

Questa sezione illustra le considerazioni sulle prestazioni legate all'uso CloudWatch di RUM.

- Impatto sulle prestazioni di caricamento: il client Web CloudWatch RUM può essere installato nell'applicazione Web come JavaScript modulo o caricato nell'applicazione Web in modo asincrono da una rete di distribuzione dei contenuti (CDN). Non blocca il processo di caricamento dell'applicazione. CloudWatch RUM è progettato in modo che non vi sia alcun impatto percettibile sul tempo di caricamento dell'applicazione.
- Impatto sul runtime: il client web RUM esegue l'elaborazione per registrare e inviare i dati RUM al CloudWatch servizio RUM. Poiché gli eventi sono rari e la quantità di elaborazione è ridotta,

CloudWatch RUM è progettato in modo da non avere alcun impatto rilevabile sulle prestazioni dell'applicazione.

- Impatto sulla rete: il client web RUM invia periodicamente dati al CloudWatch servizio RUM. I dati vengono inviati a intervalli regolari mentre l'applicazione è in esecuzione e anche immediatamente prima che il browser scarichi l'applicazione. I dati inviati immediatamente prima che il browser scarichi l'applicazione sono inviati come beacon, che sono progettati per non avere un impatto rilevabile sul tempo di scarico dell'applicazione.

Prezzi di RUM

Con CloudWatch RUM, vengono addebitati costi per ogni evento RUM ricevuto da CloudWatch RUM. Ogni elemento di dati raccolti utilizzando il client Web RUM è considerato un evento RUM. Esempi di eventi RUM includono una visualizzazione di pagina, un JavaScript errore e un errore HTTP. Vi sono opzioni per i tipi di eventi raccolti da ciascun monitor dell'app. È possibile attivare o disattivare le opzioni per raccogliere eventi di telemetria delle prestazioni, JavaScript errori, errori HTTP e tracce X-Ray. Per ulteriori informazioni sula scelta di queste opzioni, consulta Fase 2: creazione di un monitor dell'app e Informazioni raccolte dal client web RUM CloudWatch . Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatchdei prezzi di Amazon.

Disponibilità nelle Regioni

CloudWatch RUM è attualmente disponibile nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Mumbai)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Melbourne)
- Asia Pacifico (Osaka-Locale)
- Asia Pacific (Seul)
- Asia Pacifico (Singapore)

- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Spagna)
- Europa (Stoccolma)
- Europa (Zurigo)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)

Argomenti

- Risoluzione dei problemi RUM CloudWatch

Politiche IAM per l'utilizzo di CloudWatch RUM

Per poter gestire completamente CloudWatch RUM, è necessario accedere come utente o ruolo IAM con la policy AmazonCloudWatchRUM FullAccess IAM. Inoltre, potrebbero essere necessari altri criteri o autorizzazioni:

- Per creare un app monitor che crei un nuovo pool di identità Amazon Cognito per l'autorizzazione, devi disporre del ruolo Admin IAM o della policy AdministratorAccessIAM.
- Per creare un app monitor che invii dati a CloudWatch Logs, devi accedere a un ruolo o a una policy IAM con le seguenti autorizzazioni:

```
{
    "Effect": "Allow",
    "Action": [
        "logs:PutResourcePolicy"
    ],
    "Resource": [
        "*"
    ]
}
```

Agli altri utenti che devono visualizzare i dati CloudWatch RUM ma non devono creare risorse CloudWatch RUM, può essere concessa la politica AmazonCloudWatchRUM ReadOnlyAccess.

Configura un'applicazione per utilizzare CloudWatch RUM

Segui i passaggi descritti in queste sezioni per configurare l'applicazione e iniziare a utilizzare CloudWatch RUM per raccogliere dati sulle prestazioni da sessioni utente reali.

Argomenti

- Passaggio 1: autorizza l'applicazione a inviare dati a AWS
- Fase 2: creazione di un monitor dell'app
- (Opzionale) Fase 3: Modifica manualmente lo snippet di codice per configurare il CloudWatch client web RUM
- Passaggio 4: inserire lo snippet di codice nella propria applicazione

- Passaggio 5: verificare la configurazione del monitor dell'app generando eventi utente

Passaggio 1: autorizza l'applicazione a inviare dati a AWS

Per utilizzare CloudWatch RUM, l'applicazione deve essere autorizzata.

Sono disponibili tre opzioni per impostare l'autorizzazione:

- Consenti a CloudWatch RUM di creare un nuovo pool di identità Amazon Cognito per l'applicazione. Questo metodo richiede il minimo sforzo per la configurazione. Questa è l'opzione di default.

  Il pool di identità conterrà un'identità non autenticata. Ciò consente al client web CloudWatch RUM di inviare dati a CloudWatch RUM senza autenticare l'utente dell'applicazione.

  Il pool di identità di Amazon Cognito ha un ruolo IAM associato. L'identità non autenticata di Amazon Cognito consente al client Web di assumere il ruolo IAM autorizzato a inviare dati a RUM. CloudWatch
- Utilizzare un pool di identità Amazon Cognito esistente. In questo caso, è inoltre necessario modificare il ruolo IAM associato al pool di identità. Utilizza questa opzione per i pool di identità che supportano utenti non autenticati. È possibile utilizzare i pool di identità solo della stessa regione.
- Utilizzare l'autenticazione da un provider di identità esistente già configurato. In questo caso, è necessario ottenere le credenziali dal provider di identità e l'applicazione deve inoltrare queste credenziali al client Web RUM.

  Utilizza questa opzione per i pool di identità che supportano solo utenti autenticati.

Le seguenti sezioni includono maggiori dettagli su queste opzioni.

CloudWatch RUM crea un nuovo pool di identità Amazon Cognito

Questa è l'opzione più semplice da configurare e, se si sceglie questa opzione, non sono necessari ulteriori passaggi di configurazione. È necessario disporre delle autorizzazioni amministrative per utilizzare questa opzione. Per ulteriori informazioni, consulta Politiche IAM per l'utilizzo di CloudWatch RUM.

Con questa opzione, CloudWatch RUM crea le seguenti risorse:

- Pool di identità di Amazon Cognito

- Un'identità Amazon Cognito non autenticata. Ciò consente al client Web RUM di assumere un ruolo IAM senza autenticare l'utente dell'applicazione.
- Il ruolo IAM che assumerà il client Web RUM. La policy IAM allegata a questo ruolo consente di utilizzare l'API di `PutRumEvents` con la risorsa per il monitoraggio dell'app. In altre parole, consente al client Web RUM di inviare dati a RUM.

Il client web RUM utilizza l'identità Amazon Cognito per ottenere AWS le credenziali. Le AWS credenziali sono associate al ruolo IAM. Il ruolo IAM è autorizzato all'uso `PutRumEvents` con la AppMonitor risorsa.

Amazon Cognito invia il token di sicurezza necessario per consentire all'applicazione di inviare dati a CloudWatch RUM. Il frammento di JavaScript codice generato da CloudWatch RUM include le seguenti righe per abilitare l'autenticazione.

```
   {
    identityPoolId: [identity pool id],  // e.g., 'us-west-2:EXAMPLE4a-66f6-4114-902a-
EXAMPLEbad7'
   }
);
```

Utilizzare un pool di identità Amazon Cognito esistente

Se scegli di utilizzare un pool di identità Amazon Cognito esistente, specifichi il pool di identità quando aggiungi l'applicazione a CloudWatch RUM. Il pool deve supportare l'abilitazione dell'accesso a identità non autenticate. Puoi utilizzare pool di identità solo dalla stessa regione.

È inoltre necessario aggiungere le seguenti autorizzazioni alla policy IAM associata al ruolo IAM associato a questo pool di identità.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountid]:appmonitor/[app monitor
 name]"
        }
    ]
```

```
}
```

Amazon Cognito invierà quindi il token di sicurezza necessario per consentire all'applicazione di accedere CloudWatch a RUM.

Fornitore di terza parte

Se si sceglie l'autenticazione privata da un provider di terze parti, è necessario ottenere le credenziali dal provider di identità e inoltrarle a AWS. Il modo migliore per eseguire questa operazione è possibile utilizzare un fornitore di token di sicurezza. Puoi utilizzare qualsiasi fornitore di token di sicurezza, incluso Amazon Cognito AWS Security Token Service con. Per ulteriori informazioni su AWS STS, consulta Welcome to the AWS Security Token Service API Reference.

Se si desidera utilizzare Amazon Cognito come fornitore di token in questo scenario, è possibile configurare Amazon Cognito in modo che funzioni con un provider di autenticazione. Per maggiori informazioni, consulta Nozioni di base sui pool di identità di Amazon Cognito (identità federate)

Dopo aver configurato Amazon Cognito per lavorare con il proprio provider di identità, è necessario anche fare quanto segue:

- Creare un ruolo IAM con le seguenti autorizzazioni. L'applicazione utilizzerà questo ruolo per accedere a AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountID]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

- Aggiungi quanto segue alla tua applicazione per farla passare le credenziali dal tuo provider a CloudWatch RUM. Inserire la riga in modo che venga eseguita dopo che un utente ha effettuato l'accesso all'applicazione e l'applicazione ha ricevuto le credenziali da utilizzare per accedere a AWS.

```
cwr('setAwsCredentials', {/* Credentials or CredentialProvider */});
```

Per ulteriori informazioni sui provider di credenziali nell' AWS JavaScript SDK, consulta Impostazione delle credenziali in un browser Web nella guida per sviluppatori v3 per SDK per JavaScript, Impostazione delle credenziali in un browser Web nella guida per sviluppatori v2 per SDK per e @aws -sdk/credential-providers. JavaScript

Puoi anche utilizzare l'SDK per il client Web RUM per configurare i metodi di autenticazione del client Web. CloudWatch Per ulteriori informazioni sul web client SDK, consulta CloudWatch RUM web client SDK.

Fase 2: creazione di un monitor dell'app

Per iniziare a utilizzare CloudWatch RUM con la tua applicazione, devi creare un app monitor. Quando viene creato l'app monitor, RUM genera uno JavaScript snippet da incollare nell'applicazione. Lo snippet inserisce il codice del client Web RUM. Il client Web RUM acquisisce i dati da una percentuale delle sessioni utente dell'applicazione e li invia a RUM.

Creazione di un monitor dell'app

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, RUM.
3. Scegli Aggiungere un monitor app.
4. Inserire le informazioni e le impostazioni per la propria applicazione:

   - Per il nome del monitor dell'app, inserisci un nome da utilizzare per identificare questo app monitor all'interno della console CloudWatch RUM.
   - Per Dominio applicazione, inserire il nome di dominio di primo livello in cui l'applicazione dispone dell'autorità amministrativa. Deve essere in un formato di dominio URL.

     Scegli Includi sottodomini per fare in modo che il monitor dell'app raccolga anche dati da tutti i sottodomini del dominio di primo livello.
5. Per Configurazione della raccolta dei dati, specificare se si desidera che il monitor dell'app raccolga ciascuno dei seguenti elementi:

   - Telemetria delle prestazioni— Raccoglie informazioni sul caricamento della pagina e sui tempi di caricamento delle risorse
   - JavaScript errori: raccoglie informazioni sugli JavaScript errori non gestiti generati dall'applicazione
   - Errori HTTP— Raccoglie informazioni sugli errori HTTP generati dall'applicazione

La selezione di queste opzioni fornisce ulteriori informazioni sull'applicazione, ma genera anche più eventi CloudWatch RUM e quindi comporta un aumento dei costi.

Se non si seleziona nessuno di questi, il monitor dell'app raccoglie ancora gli eventi di avvio della sessione e gli ID di pagina in modo da poter vedere quanti utenti utilizzano l'applicazione, inclusi i guasti per tipo e versione del sistema operativo, tipo e versione del browser, tipo di dispositivo e posizione.

6. Seleziona Seleziona questa opzione per consentire al CloudWatch RUM Web Client di impostare i cookie se desideri essere in grado di raccogliere ID utente e ID di sessione da sessioni utente campionate. Gli ID utente vengono generati casualmente da RUM. Per ulteriori informazioni, consulta CloudWatch Cookie del client web RUM (o tecnologie simili).
7. Per Esempi di sessione, inserire la percentuale di sessioni utente che verranno utilizzate per raccogliere i dati RUM. Il valore di default è 100%. La riduzione di questo numero consente di ottenere meno dati, ma riduce i costi. Per ulteriori informazioni sui prezzi di RUM, consulta Prezzi di RUM.
8. I dati degli utenti finali raccolti per CloudWatch RUM vengono conservati per 30 giorni e poi eliminati. Se desideri conservare copie degli eventi RUM nei CloudWatch registri e configurare per quanto tempo conservare tali copie, scegli Seleziona questa opzione per memorizzare i dati di telemetria dell'applicazione nel tuo account CloudWatch Logs in Archiviazione dati. Per impostazione predefinita, il gruppo di CloudWatch log Logs conserva i dati per 30 giorni. È possibile modificare il periodo di conservazione nella console CloudWatch Logs.
9. Per Autorizzazione, specificare se utilizzare un pool di identità Amazon Cognito nuovo o esistente o utilizzare un provider di identità diverso. La creazione di un nuovo pool di identità è l'opzione più semplice che non richiede altri passaggi di configurazione. Per ulteriori informazioni, consulta Passaggio 1: autorizza l'applicazione a inviare dati a AWS.

   La creazione di un nuovo pool di identità Amazon Cognito richiede autorizzazioni amministrative. Per ulteriori informazioni, consulta Politiche IAM per l'utilizzo di CloudWatch RUM.
10. (Facoltativo) Per impostazione predefinita, quando aggiungete lo snippet di codice RUM all'applicazione, il client Web inserisce il JavaScript tag per monitorare l'utilizzo nel codice HTML di tutte le pagine dell'applicazione. Per modificarlo, scegli Configura le pagine e poi scegli tra Includi solo queste pagine o Escludi queste pagine. Quindi, specificare le pagine da includere o escludere. Per specificare una pagina da includere o escludere, inserire gli URL completi. Per specificare pagine aggiuntive, scegli Aggiungi URL.

11. Per abilitare il AWS X-Ray tracciamento delle sessioni utente campionate dall'app monitor, scegli Tracciamento attivo e seleziona Trace my service with. AWS X-Ray

    Se si seleziona questo,`XMLHttpRequest` e `fetch` vengono tracciate le richieste effettuate durante le sessioni utente campionate dal monitor dell'app. È quindi possibile visualizzare tracce e segmenti di queste sessioni utente nel pannello di controllo RUM, dalla mappa di tracciamento X-Ray e dalle pagine dei dettagli della traccia. Queste sessioni utente verranno visualizzate anche come pagine client in Application Signals dopo che l'avrai abilitata per la tua applicazione.

    Apportando ulteriori modifiche alla configurazione del client Web CloudWatch RUM, è possibile aggiungere un'intestazione di traccia X-Ray alle richieste HTTP per consentire il end-to-end tracciamento delle sessioni utente fino ai servizi gestiti a valle. AWS Per ulteriori informazioni, consulta Abilitazione del tracciamento X-Ray end-to-end .
12. (Facoltativo) Per aggiungere tag al monitor dell'app, seleziona Tag, Aggiungi nuovo tag.

    Poi, per Chiave, inserire un nome per il tag. È possibile aggiungere un valore facoltativo al tag in Value (Valore).

    Per aggiungere un altro tag, scegli nuovamente Add tag (Aggiungi tag).

    Per ulteriori informazioni, consulta Tagging Resources. AWS
13. Scegli Aggiungere un monitor app.
14. Nella sezione Sample code (Codice di esempio), puoi copiare lo snippet di codice da aggiungere alla tua applicazione. Ti consigliamo di scegliere JavaScripto TypeScriptutilizzare NPM per installare il client web CloudWatch RUM come JavaScript modulo.

    In alternativa, puoi scegliere HTML per utilizzare una rete di distribuzione dei contenuti (CDN) per installare il client web CloudWatch RUM. Lo svantaggio dell'utilizzo di una CDN è che il client Web viene spesso bloccato dalle estensioni di blocco degli annunci pubblicitari.
15. Scegli Copy (Copia) o Download (Scarica), quindi scegli Done (Fatto).

(Opzionale) Fase 3: Modifica manualmente lo snippet di codice per configurare il CloudWatch client web RUM

È possibile modificare lo snippet di codice prima di inserirlo nell'applicazione, per attivare o disattivare diverse opzioni. Per ulteriori informazioni, consulta la documentazione del client web CloudWatch RUM.

Ci sono tre opzioni di configurazione di cui si deve assolutamente essere a conoscenza, come discusso in queste sezioni.

Impedire la raccolta di URL delle risorse che potrebbero contenere informazioni personali

Per impostazione predefinita, il client web CloudWatch RUM è configurato per registrare gli URL delle risorse scaricate dall'applicazione. Queste risorse includono file HTML, immagini, file CSS, JavaScript file e così via. Per alcune applicazioni, gli URL possono contenere informazioni personali di identificazione (PII).

Se è così per la vostra applicazione, consigliamo di disattivare la raccolta degli URL delle risorse impostando `recordResourceUrl: false` nella configurazione dello snippet di codice, prima di inserirlo nell'applicazione.

Registrazione manuale delle visualizzazioni di pagina

Per impostazione predefinita, il client Web registra le visualizzazioni di pagina quando la pagina viene caricata per la prima volta e quando viene chiamata l'API della cronologia del browser. L'ID di pagina predefinito è `window.location.pathname`. Tuttavia, in alcuni casi potresti voler ignorare questo comportamento e utilizzare l'applicazione per registrare le visualizzazioni di pagina a livello di codice. In questo modo potrai controllare l'ID della pagina e quando viene registrata. Ad esempio, si consideri un'applicazione Web che dispone di un URI con un identificatore variabile, ad esempio `/entity/123` o`/entity/456`. Per impostazione predefinita, CloudWatch RUM genera un evento di visualizzazione della pagina per ogni URI con un ID di pagina distinto che corrisponde al percorso, ma potresti invece volerli raggruppare in base allo stesso ID di pagina. A tale scopo, disabilita l'automazione della visualizzazione delle pagine del client Web utilizzando la configurazione `disableAutoPageView` e utilizza il comando `recordPageView` per impostare l'ID di pagina desiderato. Per ulteriori informazioni, vedere Configurazioni specifiche dell'applicazione su. GitHub

Esempio di script incorporato:

```
cwr('recordPageView', { pageId: 'entityPageId' });
```

JavaScript esempio di modulo:

```
awsRum.recordPageView({ pageId: 'entityPageId' });
```

Abilitazione del tracciamento X-Ray end-to-end

Quando si crea il monitor dell'app, selezionando Traccia il mio servizio con AWS X-Ray consente il tracciamento di richieste `XMLHttpRequest` e `fetch` effettuate durante le sessioni utente che vengono campionate dal monitor dell'app. È quindi possibile visualizzare le tracce di queste richieste HTTP nella dashboard CloudWatch RUM e nelle pagine X-Ray Trace Map e Trace details.

Per impostazione predefinita, queste tracce lato client non sono collegate a tracce lato server a valle. Per connettere le tracce lato client alle tracce lato server e abilitare la end-to-end traccia, imposta l'`addXRayTraceIdHeader`opzione su Nel client Web. `true` Questo fa sì che il client web CloudWatch RUM aggiunga un'intestazione di traccia X-Ray alle richieste HTTP.

Il seguente blocco di codice mostra un esempio di aggiunta di tracce lato client. Alcune opzioni di configurazione vengono omesse da questo esempio per la leggibilità.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            enableXRay: true,
            telemetries: [
                'errors',
                'performance',
                [ 'http', { addXRayTraceIdHeader: true } ]
            ]
        }
    );
</script>
```

> **Warning**
>
> La configurazione del client web CloudWatch RUM per aggiungere un'intestazione di traccia X-Ray alle richieste HTTP può causare il fallimento della condivisione delle risorse tra le origini (CORS) o invalidare la firma della richiesta se la richiesta è firmata con SigV4. Per ulteriori informazioni, consulta la documentazione del client web RUM. CloudWatch Si

consiglia vivamente di testare l'applicazione prima di aggiungere un'intestazione di traccia X-Ray lato client in un ambiente di produzione.

Per ulteriori informazioni, consulta la documentazione del client web CloudWatch RUM

Passaggio 4: inserire lo snippet di codice nella propria applicazione

Successivamente, inserire lo snippet di codice creato nella sezione precedente nell'applicazione.

> **Warning**
>
> Il client web, scaricato e configurato dallo snippet di codice, utilizza cookie (o tecnologie simili) per aiutare a raccogliere i dati dell'utente finale. Prima di inserire lo snippet di codice, vedere Filtraggio in base agli attributi dei metadati nella console.

Se non si dispone dello snippet di codice generato in precedenza, è possibile trovarlo seguendo le istruzioni in Come faccio a trovare uno snippet di codice che ho già generato?.

Per inserire lo snippet di codice CloudWatch RUM nell'applicazione

1. Inserire lo snippet di codice copiato o scaricato nella sezione precedente all'interno dell'elemento `<head>` della propria applicazione. Inserirlo prima dell'elemento `<body>` o qualsiasi altro tag `<script>`.

   Il seguente è un esempio di uno snippet di codice generato:

```
<script>
(function (n, i, v, r, s, c, x, z) {
    x = window.AwsRumClient = {q: [], n: n, i: i, v: v, r: r, c: c};
    window[n] = function (c, p) {
        x.q.push({c: c, p: p});
    };
    z = document.createElement('script');
    z.async = true;
    z.src = s;
    document.head.insertBefore(z, document.getElementsByTagName('script')[0]);
})('cwr',
    '194a1c89-87d8-41a3-9d1b-5c5cd3dafbd0',
    '1.0.0',
```

```
    'us-east-2',
    'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
    {
        sessionSampleRate: 1,
        identityPoolId: "us-east-2:c90ef0ac-e3b8-4d1a-b313-7e73cfd21443",
        endpoint: "https://dataplane.rum.us-east-2.amazonaws.com",
        telemetries: ["performance", "errors", "http"],
        allowCookies: true,
        enableXRay: false
    });
</script>
```

2. Se l'applicazione è un'applicazione Web multipagina, è necessario ripetere il passaggio 1 per ogni pagina HTML che si desidera includere nella raccolta dati.

Passaggio 5: verificare la configurazione del monitor dell'app generando eventi utente

Dopo aver inserito lo snippet di codice e quando l'applicazione aggiornata è in esecuzione, è possibile testarlo generando manualmente eventi utente. Per testarlo, consigliamo anche di fare quanto segue. Questo test prevede tariffe RUM standard CloudWatch .

- Spostarsi tra le pagine dell'applicazione Web.
- Creare più sessioni utente, utilizzando browser e dispositivi diversi.
- Eseguire richieste.
- Causare JavaScript errori.

Dopo aver generato alcuni eventi, visualizzali nella dashboard CloudWatch RUM. Per ulteriori informazioni, consulta Visualizzazione del pannello di controllo CloudWatch RUM.

I dati delle sessioni utente potrebbero richiedere fino a 15 minuti per essere visualizzati nel pannello di controllo.

Se i dati non vengono visualizzati 15 minuti dopo aver generato eventi nell'applicazione, vedere Risoluzione dei problemi RUM CloudWatch .

Configurazione del client web CloudWatch RUM

Le tue applicazioni possono utilizzare uno dei frammenti di codice generati da CloudWatch RUM per installare il client web CloudWatch RUM. Gli snippet generati supportano due metodi di installazione:

come JavaScript modulo tramite NPM o da una rete di distribuzione dei contenuti (CDN). Per ottenere prestazioni migliori, ti consigliamo di utilizzare il metodo di installazione tramite NPM. Per ulteriori informazioni sull'utilizzo di questo metodo, vedete Installazione come modulo. JavaScript

Se utilizzi l'opzione di installazione CDN, gli ad blocker potrebbero bloccare il CDN predefinito fornito da RUM. CloudWatch Questo disabilita il monitoraggio delle applicazioni per gli utenti che hanno installato estensioni di blocco degli annunci pubblicitari. Per questo motivo, si consiglia di utilizzare il CDN predefinito solo per l'onboarding iniziale con RUM. CloudWatch Per ulteriori informazioni sulle modalità di risoluzione di questo problema, consulta la sezione Analisi dell'applicazione.

Lo snippet di codice si trova nel tag <head> di un file HTML e installa il client Web scaricando il client Web e quindi configurandolo per l'applicazione che sta monitorando. Lo snippet è una funzione autoesecuzione simile alla seguente. In questo esempio, il corpo della funzione dello snippet è stato omesso per la leggibilità.

```
<script>
(function(n,i,v,r,s,c,u,x,z){...})(
'cwr',
'00000000-0000-0000-0000-000000000000',
'1.0.0',
'us-west-2',
'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
{ /* Configuration Options Here */ }
);
<script>
```

Argomenti

Lo snippet di codice accetta sei argomenti:

- Uno spazio dei nomi per l'esecuzione di comandi sul client Web, ad esempio `'cwr'`
- L'ID del monitor dell'app, ad esempio `'00000000-0000-0000-0000-000000000000'`
- Versione dell'applicazione, come `'1.0.0'`
- La AWS regione del monitor dell'app, ad esempio `'us-west-2'`
- L'URL del client Web, come `'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js'`
- Opzioni di configurazione specifiche dell'applicazione. Per ulteriori informazioni, consulta la sezione seguente.

Ignorare errori

Il client web CloudWatch RUM ascolta tutti i tipi di errori che si verificano nelle applicazioni. Se l'applicazione emette JavaScript errori che non desiderate visualizzare nella dashboard CloudWatch RUM, potete configurare il client web CloudWatch RUM per filtrare questi errori in modo da visualizzare solo gli eventi di errore pertinenti sulla dashboard CloudWatch RUM. Ad esempio, potreste scegliere di non visualizzare alcuni JavaScript errori nella dashboard perché avete già identificato una soluzione e il volume di questi errori maschera altri errori. È possibile anche ignorare gli errori che non si possono correggere perché appartenenti a una libreria di proprietà di terze parti.

Per ulteriori informazioni su come utilizzare il client Web per filtrare JavaScript errori specifici, consulta l'esempio in Errori nella documentazione Github del client web.

Opzioni di configurazione

Per informazioni sulle opzioni di configurazione disponibili per il client web CloudWatch RUM, consultate la documentazione del client web CloudWatch RUM

Regionalizzazione

Questa sezione illustra le strategie per l'utilizzo di CloudWatch RUM con applicazioni in diverse regioni.

La mia applicazione web è distribuita in più regioni AWS

Se la tua applicazione web è distribuita in più AWS regioni, hai tre opzioni:

- Implementare un monitor di app in un'unica Eegione, in un unico account, che funziona in tutte le Regioni.
- Implementare monitor di app separati per ogni Regione, in account unici.
- Implementare monitor di app separati per ogni Regione, tutti in un account.

Il vantaggio dell'utilizzo di un unico app monitor è che tutti i dati saranno centralizzati in un'unica visualizzazione e tutti i log verranno scritti nello stesso gruppo di log in Logs. CloudWatch Con un unico monitor di app c'è una piccola latenza aggiuntiva per le richieste e un unico punto di errore.

L'utilizzo di più monitor per app rimuove il singolo punto di errore, ma impedisce che tutti i dati vengano combinati in un'unica visualizzazione.

CloudWatch RUM non è stato avviato in alcune regioni in cui è distribuita la mia applicazione

CloudWatch RUM viene lanciato in molte regioni e ha un'ampia copertura geografica. Configurando il CloudWatch RUM nelle regioni in cui è disponibile, è possibile ottenere i vantaggi. Gli utenti finali possono essere ovunque e avere comunque le sessioni incluse se hai configurato un monitor di app nella regione a cui si connettono.

Tuttavia, il CloudWatch RUM non è ancora stato lanciato in AWS GovCloud (Stati Uniti orientali), AWS GovCloud (Stati Uniti occidentali) o in alcuna regione della Cina. Non è possibile inviare dati a CloudWatch RUM da queste regioni.

Utilizzo dei gruppi di pagine

Utilizzare i gruppi di pagine per associare tra loro diverse pagine dell'applicazione in modo da poter visualizzare analisi aggregate per gruppi di pagine. Ad esempio, è possibile visualizzare i tempi di caricamento delle pagine aggregati di tutte le pagine di destinazione.

Puoi inserire le pagine in gruppi di pagine aggiungendo uno o più tag agli eventi di visualizzazione delle pagine nel client web CloudWatch RUM. Negli esempi seguenti la pagina `/home` viene inserita nei gruppi di pagine denominati `en` e `landing`.

Esempio di script incorporato

```
cwr('recordPageView', { pageId: '/home', pageTags: ['en', 'landing']});
```

JavaScript esempio di modulo

```
awsRum.recordPageView({ pageId: '/home', pageTags: ['en', 'landing']});
```

> **Note**
>
> I gruppi di pagine hanno lo scopo di facilitare l'aggregazione dell'analisi su pagine diverse. Per informazioni su come definire e manipolare gli `pageIds` per l'applicazione, consulta Registrazione manuale delle visualizzazioni di pagina in (Opzionale) Fase 3: Modifica manualmente lo snippet di codice per configurare il CloudWatch client web RUM.

Specifica di metadati personalizzati

CloudWatch RUM allega dati aggiuntivi a ciascun evento come metadati. I metadati degli eventi sono costituiti da attributi sotto forma di coppie chiave-valore. È possibile utilizzare questi attributi per cercare o filtrare gli eventi nella console CloudWatch RUM. Per impostazione predefinita, CloudWatch RUM crea alcuni metadati per voi. Per ulteriori informazioni sui metadati predefiniti, consulta Metadati degli eventi RUM.

È inoltre possibile utilizzare il client web CloudWatch RUM per aggiungere metadati personalizzati agli eventi CloudWatch RUM. I metadati personalizzati possono includere attributi di sessione e attributi di pagina.

Per aggiungere metadati personalizzati, è necessario utilizzare la versione 1.10.0 o successiva del CloudWatch client web RUM.

Requisiti e sintassi

Ogni evento può includere fino a 10 attributi personalizzati nei metadati. I requisiti di sintassi per gli attributi personalizzati sono i seguenti:

- Chiavi
  - Massimo 128 caratteri
  - Può includere caratteri alfanumerici, due punti (:) e caratteri di sottolineatura (_)
  - Non può iniziare con `aws:`.
  - Non può essere composto interamente da nessuna delle parole chiave riservate elencate nella sezione seguente. È possibile utilizzare queste parole chiave come parte di un nome chiave più lungo.
- Valori
  - Massimo 256 caratteri
  - Deve essere costituito da stringhe, numeri o valori booleani

Parole chiave riservate

Non puoi utilizzare le seguenti parole chiave riservate come nomi chiave completi. Puoi utilizzare le seguenti parole chiave come parte di un nome chiave più lungo, ad esempio `applicationVersion`.

- `browserLanguage`

- `browserName`
- `browserVersion`
- `countryCode`
- `deviceType`
- `domain`
- `interaction`
- `osName`
- `osVersion`
- `pageId`
- `pageTags`
- `pageTitle`
- `pageUrl`
- `parentPageId`
- `platformType`
- `referrerUrl`
- `subdivisionCode`
- `title`
- `url`
- `version`

> **Note**
>
> CloudWatch RUM rimuove gli attributi personalizzati dagli eventi RUM se un attributo include una chiave o un valore non valido o se è già stato raggiunto il limite di 10 attributi personalizzati per evento.

Aggiunta di attributi di sessione

Quando configuri degli attributi di sessione personalizzati, questi vengono aggiunti a tutti gli eventi di una sessione. È possibile configurare gli attributi di sessione durante l'inizializzazione del client Web CloudWatch RUM o in fase di esecuzione utilizzando il `addSessionAttributes` comando.

Ad esempio, puoi aggiungere la versione dell'applicazione come attributo di sessione. Quindi, nella console CloudWatch RUM, puoi filtrare gli errori per versione per scoprire se un tasso di errore maggiore è associato a una particolare versione dell'applicazione.

Aggiunta di un attributo di sessione all'inizializzazione (esempio NPM)

La sezione del codice in grassetto aggiunge l'attributo di sessione.

```
import { AwsRum, AwsRumConfig } from 'aws-rum-web';

try {
  const config: AwsRumConfig = {
    allowCookies: true,
    endpoint: "https://dataplane.rum.us-west-2.amazonaws.com",
    guestRoleArn: "arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth",
    identityPoolId: "us-west-2:00000000-0000-0000-0000-000000000000",
    sessionSampleRate: 1,
    telemetries: ['errors', 'performance'],
    sessionAttributes: {
        applicationVersion: "1.3.8"
    }
  };

  const APPLICATION_ID: string = '00000000-0000-0000-0000-000000000000';
  const APPLICATION_VERSION: string = '1.0.0';
  const APPLICATION_REGION: string = 'us-west-2';

  const awsRum: AwsRum = new AwsRum(
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    config
  );
} catch (error) {
  // Ignore errors thrown during CloudWatch RUM web client initialization
}
```

Aggiunta di un attributo di sessione in fase di runtime (esempio NPM)

```
awsRum.addSessionAttributes({
    applicationVersion: "1.3.8"
```

```
})
```

Aggiunta di un attributo di sessione in fase di inizializzazione (esempio di script incorporato)

La sezione del codice in grassetto aggiunge l'attributo di sessione.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            sessionSampleRate:1,
            guestRoleArn:'arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth',
            identityPoolId:'us-west-2:00000000-0000-0000-0000-000000000000',
            endpoint:'https://dataplane.rum.us-west-2.amazonaws.com',
            telemetries:['errors','http','performance'],
            allowCookies:true,
            sessionAttributes: {
                applicationVersion: "1.3.8"
            }
        }
    );
</script>
```

Aggiunta di un attributo di sessione in fase di runtime (esempio di script incorporato)

```
<script>
    function addSessionAttribute() {
        cwr('addSessionAttributes', {
            applicationVersion: "1.3.8"
        })
    }

</script>
```

Aggiunta di attributi di pagina

Quando configuri gli attributi di pagina personalizzati, questi vengono aggiunti a tutti gli eventi della pagina corrente. È possibile configurare gli attributi di pagina durante l'inizializzazione del client Web CloudWatch RUM o in fase di esecuzione utilizzando il `recordPageView` comando.

Ad esempio, puoi aggiungere il modello di pagina come attributo di pagina. Quindi, nella console CloudWatch RUM, è possibile filtrare gli errori in base ai modelli di pagina per scoprire se un tasso di errore maggiore è associato a un particolare modello di pagina dell'applicazione.

Aggiunta di un attributo di pagina in fase di inizializzazione (esempio NPM)

La sezione del codice in grassetto aggiunge l'attributo di pagina.

```
const awsRum: AwsRum = new AwsRum(
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    { disableAutoPageView:  true // optional }
);
awsRum.recordPageView({
    pageId:'/home',
    pageAttributes: {
      template: 'artStudio'
    }
});
const credentialProvider = new CustomCredentialProvider();
if(awsCreds) awsRum.setAwsCredentials(credentialProvider);
```

Aggiunta di un attributo di pagina in fase di runtime (esempio NPM)

```
awsRum.recordPageView({
    pageId: '/home',
    pageAttributes: {
        template: 'artStudio'
    }
});
```

Aggiunta di un attributo di pagina in fase di inizializzazione (esempio di script incorporato)

La sezione del codice in grassetto aggiunge l'attributo di pagina.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            disableAutoPageView: true //optional
        }
    );
    cwr('recordPageView', {
       pageId: '/home',
       pageAttributes: {
           template: 'artStudio'
       }
    });
    const awsCreds = localStorage.getItem('customAwsCreds');
    if(awsCreds) cwr('setAwsCredentials', awsCreds)
</script>
```

Aggiunta di un attributo di pagina in fase di runtime (esempio di script incorporato)

```
<script>
    function recordPageView() {
        cwr('recordPageView', {
            pageId: '/home',
            pageAttributes: {
                template: 'artStudio'
            }
        });
    }
</script>
```

Filtraggio in base agli attributi dei metadati nella console

Per filtrare le visualizzazioni nella console CloudWatch RUM con qualsiasi attributo di metadati integrato o personalizzato, utilizza la barra di ricerca. Nella barra di ricerca, puoi specificare fino a 20 termini di filtro sotto forma di chiave=valore da applicare alle visualizzazioni. Ad esempio, per filtrare i dati solo per il browser Chrome, puoi aggiungere il termine di ricerca browserName=Chrome.

Per impostazione predefinita, la console CloudWatch RUM recupera i 100 attributi, chiavi e valori più comuni da visualizzare nel menu a discesa nella barra di ricerca. Per aggiungere altri attributi di metadati come termini di filtro, inserisci la chiave e il valore completi dell'attributo nella barra di ricerca.

Un filtro può includere fino a 20 termini e puoi salvare fino a 20 filtri per monitorare l'app. Quando salvi un filtro, questo viene salvato nel menu a discesa Saved filters (Filtri salvati). Puoi anche eliminare i filtri salvati.

Invio di eventi personalizzati

CloudWatch RUM registra e inserisce gli eventi elencati inInformazioni raccolte dal client web RUM CloudWatch . Se si utilizza la versione 1.12.0 o successiva del client web CloudWatch RUM, è possibile definire, registrare e inviare eventi personalizzati aggiuntivi. Puoi stabilire il nome del tipo di evento e i dati da inviare per ogni tipo di evento che definisci. Ogni payload di eventi personalizzato può contenere fino a 6 KB.

Gli eventi personalizzati vengono inseriti solo se sono stati abilitati nel monitoraggio dell'app. Per aggiornare le impostazioni di configurazione dell'app monitor, utilizza la console CloudWatch RUM o l'UpdateAppMonitorAPI.

Dopo aver abilitato gli eventi personalizzati e aver definito e inviato gli eventi personalizzati, puoi effettuare una ricerca. Per cercarle, usa la scheda Eventi nella console CloudWatch RUM. Effettua la ricerca utilizzando il tipo di evento.

Requisiti e sintassi

Gli eventi personalizzati sono costituiti da un tipo di evento e dai relativi dettagli. I parametri sono i seguenti:

- Tipo di evento
  - Questo valore può essere il tipo o il nome dell'evento. Ad esempio, il tipo di evento integrato CloudWatch RUM chiamato JsErrorha un tipo di evento di`com.amazon.rum.js_error_event`.
  - Deve contenere da 1 a 256 caratteri.
  - Può essere una combinazione di caratteri alfanumerici, caratteri di sottolineatura, trattini e punti.
- Dettagli dell'evento
  - Contiene i dati effettivi che si desidera registrare in CloudWatch RUM.

- Deve essere un oggetto composto da campi e valori.

Esempi di registrazione di eventi personalizzati

Esistono due modi per registrare eventi personalizzati nel client web CloudWatch RUM.

- Utilizza l'`recordEvent`API del client web CloudWatch RUM.
- Utilizza un plug-in personalizzato.

Invio di un evento personalizzato tramite l'API **`recordEvent`** (esempio NPM)

```
awsRum.recordEvent('my_custom_event', {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
)
```

Invio di un evento personalizzato tramite l'API **`recordEvent`** (esempio di script incorporato)

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Esempio di invio di un evento personalizzato tramite un plug-in personalizzato

```
// Example of a plugin that listens to a scroll event, and
```

```
// records a 'custom_scroll_event' that contains the timestamp of the event.
class MyCustomPlugin implements Plugin {
    // Initialize MyCustomPlugin.
    constructor() {
        this.enabled;
        this.context;
        this.id = 'custom_event_plugin';
    }
    // Load MyCustomPlugin.
    load(context) {
        this.context = context;
        this.enable();
    }
    // Turn on MyCustomPlugin.
    enable() {
        this.enabled = true;
        this.addEventHandler();
    }
    // Turn off MyCustomPlugin.
    disable() {
        this.enabled = false;
        this.removeEventHandler();
    }
    // Return MyCustomPlugin Id.
    getPluginId() {
        return this.id;
    }
    // Record custom event.
    record(data) {
        this.context.record('custom_scroll_event', data);
    }
    // EventHandler.
    private eventHandler = (scrollEvent: Event) => {
        this.record({timestamp: Date.now()})
    }
    // Attach an eventHandler to scroll event.
    private addEventHandler(): void {
        window.addEventListener('scroll', this.eventHandler);
    }
    // Detach eventHandler from scroll event.
    private removeEventHandler(): void {
        window.removeEventListender('scroll', this.eventHandler);
    }
```

```
}
```

Visualizzazione del pannello di controllo CloudWatch RUM

CloudWatch RUM consente di raccogliere dati dalle sessioni utente sulle prestazioni dell'applicazione, inclusi i tempi di caricamento delle pagine, il punteggio Apdex, i browser e i dispositivi utilizzati, la geolocalizzazione delle sessioni utente e le sessioni con errori. Tutte queste informazioni vengono visualizzate in un pannello di controllo.

Per visualizzare il pannello di controllo di RUM

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, RUM.

   La scheda Panoramica visualizza le informazioni raccolte da uno dei monitor dell'app che creati.

   La riga superiore di riquadri mostra le seguenti informazioni per questo monitor dell'app:

   - Numero di caricamenti delle pagine
   - Velocità media di caricamento della pagina
   - Punteggio Apdex
   - Stato di tutti gli allarmi associati al monitor dell'app

   L'indice di prestazioni dell'applicazione (Apdex) indica il livello di soddisfazione degli utenti finali. I punteggi variano da 0 (meno soddisfatti) a 1 (i più soddisfatti). I punteggi si basano solo sulle prestazioni dell'applicazione. Agli utenti non viene chiesto di valutare l'applicazione. Per ulteriori informazioni sui punteggi Apdex, consulta In che modo CloudWatch RUM imposta i punteggi di Apdex.

   Molti di questi riquadri includono collegamenti che è possibile utilizzare per esaminare ulteriormente i dati. La scelta di uno di questi link mostra una vista dettagliata con le schede Prestazioni, Errori, Richieste HTTP, Sessioni, Browser e dispositivi degli eventi e Percorso utente nella parte superiore dello schermo.
3. Per concentrarti ulteriormente, scegli la scheda List view (Visualizzazione elenco) quindi scegli il nome del monitor dell'app su cui vuoi concentrarti. Visualizza le seguenti schede per il monitor dell'app scelto.

- La scheda Performance (Prestazioni) visualizza le informazioni sulle prestazioni della pagina, inclusi tempi di caricamento, informazioni sulla sessione, informazioni sulle richieste, parametri vitali web e i dati sul caricamento della pagina nel tempo. Questa vista include controlli per alternare la visualizzazione tra Caricamento di pagine, Richieste, e Posizione.
- La scheda Errori visualizza le informazioni sugli errori Javascript, incluso il messaggio di errore più frequentemente visualizzato da utenti, dispositivi e browser con il maggior numero di errori. Questa visualizzazione include un istogramma degli errori e una visualizzazione a elenco degli errori. È possibile filtrare l'elenco degli errori per dettagli dell'utente e dettagli dell'evento. Scegli un messaggio di errore per visualizzare maggiori dettagli.
- La scheda Richieste HTTP mostra le informazioni sulla richiesta HTTP, tra cui l'URL della richiesta con il maggior numero di errori e i dispositivi e i browser con il maggior numero di errori. Questa scheda include un istogramma delle richieste, una vista a elenco delle richieste e una vista a elenco degli errori di rete. È possibile filtrare gli elenchi per dettagli dell'utente e dettagli dell'evento. Scegli un codice di risposta o un messaggio di errore per visualizzare maggiori dettagli rispettivamente sulla richiesta o sull'errore di rete.
- La scheda Sessioni mostra i parametri della sessione. Questa scheda include un istogramma degli eventi di inizio della sessione e una vista a elenco delle sessioni. È possibile filtrare l'elenco delle sessioni per tipo di evento, dettagli utente e dettagli dell'evento. Scegli un sessionId per visualizzare maggiori dettagli su una sessione.
- La scheda Eventi mostra un istogramma degli eventi RUM e una vista a elenco degli eventi. È possibile filtrare l'elenco delle sessioni per tipo di evento, dettagli dell'utente e dettagli dell'evento. Scegli un evento RUM per visualizzare l'evento non elaborato.
- La scheda Browser e dispositivi visualizza informazioni come le prestazioni e l'utilizzo di diversi browser e dispositivi per accedere all'applicazione. Questa vista include controlli per alternare la visualizzazione tra Browser e Dispositivi.

  Se si restringe l'ambito a un singolo browser, vengono visualizzati i dati suddivisi per versione del browser.
- La schedaPercorso dell'utente visualizza i percorsi utilizzati dai clienti per navigare nell'applicazione. È possibile vedere dove i clienti entrano nell'applicazione e da quale pagina escono dall'applicazione. È possibile anche vedere i percorsi che seguono e la percentuale di clienti che seguono tali percorsi. È possibile fermarsi su un nodo per ottenere ulteriori dettagli su quella pagina. È possibile scegliere un singolo percorso per evidenziare le connessioni per facilitare la visualizzazione.

4. (Facoltativo) In una qualsiasi delle prime sei schede, è possibile scegliere il pulsante Pagine e selezionare una pagina o un gruppo di pagine dall'elenco. Ciò riduce i dati visualizzati a una singola pagina o un gruppo di pagine dell'applicazione. È inoltre possibile contrassegnare le pagine a gruppi di pagine dell'elenco come preferite.

In che modo CloudWatch RUM imposta i punteggi di Apdex

Apdex (Indice delle prestazioni dell'applicazione) è uno standard aperto che definisce un metodo per il report, il benchmark e il tempo di risposta delle applicazioni. Un punteggio Apdex aiuta a comprendere e identificare l'impatto sulle prestazioni delle applicazioni nel tempo.

Il punteggio Apdex indica il livello di soddisfazione degli utenti finali che va da 0 (meno soddisfatti) a 1 (più soddisfatti). I punteggi si basano solo sulle prestazioni dell'applicazione. Agli utenti non viene chiesto di valutare l'applicazione.

Ogni singolo punteggio Apdex rientra in una delle tre soglie. In base alla soglia Apdex e al tempo di risposta effettivo dell'applicazione, esistono tre tipi di prestazioni, come segue:

- Soddisfatti— Il tempo di risposta effettivo dell'applicazione è inferiore o uguale alla soglia Apdex. Per CloudWatch RUM, questa soglia è pari o inferiore a 2000 ms.
- Tollerabile— Il tempo di risposta effettivo dell'applicazione è maggiore della soglia Apdex, ma inferiore o uguale a quattro volte la soglia Apdex. Per CloudWatch RUM, questo intervallo è compreso tra 2000 e 8000 ms.
- Frustrante— Il tempo di risposta effettivo dell'applicazione è superiore a quattro volte la soglia Apdex. Per CloudWatch RUM, questo intervallo è superiore a 8000 ms.

Il punteggio totale di 0-1 Apdex viene calcolato utilizzando la seguente formula:

```
(positive scores + tolerable scores/2)/total scores * 100
```

CloudWatch metriche che puoi raccogliere con RUM CloudWatch

La tabella in questa sezione elenca le metriche raccolte automaticamente con CloudWatch RUM. Puoi visualizzare queste metriche nella CloudWatch console. Per ulteriori informazioni, consulta Visualizzazione di parametri disponibili.

Facoltativamente, puoi anche inviare metriche estese a CloudWatch o Evidently. CloudWatch Per ulteriori informazioni, consulta Parametri estesi.

Tali parametri sono pubblicati nel parametro spazio dei nomi denominato `AWS/RUM`. Tutti i parametri seguenti sono pubblicati con una dimensione `application_name`. Il valore di questa dimensione è il nome del monitor dell'app. Alcuni parametri vengono pubblicati anche con dimensioni aggiuntive, come indicato nella tabella.

Parametro	Unità	Descrizione
`HttpStatusCodeCount`	Conteggio	Il conteggio delle risposte HTTP nell'applicazione, in base al codice di stato della risposta. Dimensioni aggiuntive: • `event_details.response.status` è il codice di stato della risposta, come 200, 400, 404 e così via. • `event_type` Tipo di evento. Attualmente l'unico valore possibile per questa dimensione è `http`.
`Http4xxCount`	Conteggio	Il numero di risposte HTTP nell'applicazione, con codice di stato della risposta 4xx. Questi vengono calcolati in base agli

Parametro	Unità	Descrizione
		eventi `http_event` RUM che generano codici 4xx.
`Http5xxCount`	Conteggio	Il numero di risposte HTTP nell'appl icazione, con codice di stato della risposta 5xx. Questi vengono calcolati in base agli eventi `http_event` RUM che generano codici 5xx.
`JsErrorCount`	Conteggio	Il numero di eventi di JavaScript errore ingeriti.
`NavigationFrustratedCount`	Conteggio	Il numero di eventi di navigazione con una `duration` superiore alla soglia critica, che è di 8000 ms. La durata degli eventi di navigazio ne è tracciata nel parametro `Performan ceNavigat ionDuration` .

Parametro	Unità	Descrizione
`NavigationSatisfiedCount`	Conteggio	Il numero di eventi di navigazione con una `duration` inferiore all'obiettivo Apdex, che è di 2000 ms. La durata degli eventi di navigazione è tracciata nel parametro `PerformanceNavigationDuration`.
`NavigationToleratedCount`	Conteggio	Il numero di eventi di navigazione con una `duration` tra 2000 e 8000 ms. La durata degli eventi di navigazione è tracciata nel parametro `PerformanceNavigationDuration`.
`PageViewCount`	Conteggio	Il conteggio degli eventi di visualizzazione della pagina acquisiti dal monitor dell'app. Questo viene calcolato contando gli eventi RUM. `page_view_event`

Parametro	Unità	Descrizione
`PerformanceResourceDuration`	Millisecondi	La `duration` di un evento della risorsa. Dimensioni aggiuntive: • `event_details.file.type` è il tipo di file dell'evento della risorsa, ad esempio un foglio di stile, un documento, un'immagine, uno script o un carattere. • `event_type` Tipo di evento. Attualmente l'unico valore possibile per questa dimensione è `resource`.
`PerformanceNavigationDuration`	Millisecondi	La `duration` di un evento di navigazione.

Parametro	Unità	Descrizione
`RumEventPayloadSize`	Byte	La dimensione di ogni evento importato da RUM. CloudWatch È possibile utilizzar e anche la statistic a `SampleCount` per questo parametro per monitorare il numero di eventi che un monitor app sta acquisendo.
`SessionCount`	Conteggio	Il numero di eventi di avvio della sessione acquisiti dal monitor dell'app. In altre parole,il numero di nuove sessioni avviate.
`WebVitalsCumulativeLayoutShift`	Nessuno	Tiene traccia del valore degli eventi cumulativi di spostamento del layout.
`WebVitalsFirstInputDelay`	Millisecondi	Tiene traccia del valore dei primi eventi di ritardo di input.
`WebVitalsLargestContentfulPaint`	Millisecondi	Monitora il valore degli eventi Largest Contentful Paint.

Metriche personalizzate e metriche estese che puoi inviare a ed Evidently CloudWatch CloudWatch

Per impostazione predefinita, i monitor delle app RUM inviano le metriche a. CloudWatch Queste metriche e dimensioni predefinite sono elencate nelle CloudWatch metriche che puoi raccogliere con RUM. CloudWatch

Puoi anche configurare un monitor dell'app per esportare le metriche. L'app monitor può inviare metriche estese, metriche personalizzate o entrambe. Può inviarli a CloudWatch o a CloudWatch Evidently o a entrambi.

- Parametri personalizzati: i parametri personalizzati sono parametri da te definiti. Con i parametri personalizzati, puoi utilizzare qualsiasi nome e spazio dei nomi del parametro. Per derivare i parametri, puoi utilizzare qualsiasi evento personalizzato, evento integrato, attributo personalizzato o attributo predefinito.

  Puoi inviare metriche personalizzate sia a Evidently che CloudWatch a Evidently. CloudWatch
- Metriche estese: consente di inviare le metriche CloudWatch RUM predefinite a Evidently per utilizzarle CloudWatch negli esperimenti Evidently. Puoi anche inviare qualsiasi metrica CloudWatch RUM predefinita a con dimensioni aggiuntive. CloudWatch In questo modo, questi parametri possono offrirti una visione più dettagliata.

Argomenti

Parametri personalizzati

Per inviare metriche personalizzate, devi utilizzare le AWS API o al AWS CLI posto della console. Per ulteriori informazioni sull'utilizzo delle AWS API, consulta e. PutRumMetricsDestinationBatchCreateRumMetricDefinitions

Il numero massimo di definizioni di parametri personalizzati e parametri estesi che una singola destinazione può contenere è 2.000. Per ogni parametro personalizzato o esteso inviato a ciascuna destinazione, ogni combinazione di nome e valore della dimensione conta per tale limite. Questo vale anche come metrica CloudWatch personalizzata per i prezzi.

L'esempio seguente mostra come creare un parametro personalizzato derivato da un evento personalizzato. Ecco l'esempio di evento personalizzato utilizzato:

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Dato questo evento personalizzato, puoi creare un parametro personalizzato che conta il numero di visite all'URL di `amazonaws.com` dai browser Chrome. La seguente definizione crea un parametro denominato `AmazonVisitsCount` nel tuo account, nello spazio dei nomi `RUM/CustomMetrics/PageVisits`.

```
{
    "AppMonitorName":"customer-appMonitor-name",
    "Destination":"CloudWatch",
    "MetricDefinitions":[
        {
            "Name":"AmazonVisitsCount",
            "Namespace":"PageVisit",
            "ValueKey":"event_details.visit_count",
            "UnitLabel":"Count",
            "DimensionKeys":{
                "event_details.current_url": "URL"
            },
            "EventPattern":"{\"metadata\":{\"browserName\":[\"Chrome\"]},\"event_type
\":[\"my_custom_event\"],\"event_details\": {\"current_url\": [\"amazonaws.com\"]}}"
        }
    ]
}
```

Parametri estesi

Se imposti le metriche estese, puoi eseguire una o entrambe le operazioni seguenti:

- Invia le metriche CloudWatch RUM predefinite a Evidently per utilizzarle negli esperimenti CloudWatch Evidently. Solo le WebVitalsLargestContentfulPaintmetriche PerformanceNavigationDuration, PerformanceResourceDuration, WebVitalsCumulativeLayoutShiftWebVitalsFirstInputDelay, e possono essere inviate a Evidently.
- Invia una qualsiasi delle metriche CloudWatch RUM predefinite a CloudWatch con dimensioni aggiuntive in modo che le metriche ti offrano una visione più dettagliata. Ad esempio, puoi visualizzare i parametri specifici di un determinato browser utilizzato dagli utenti o le metriche degli utenti in una geolocalizzazione specifica.

Per ulteriori informazioni sulle metriche RUM predefinite, consulta CloudWatch . CloudWatch metriche che puoi raccogliere con RUM CloudWatch

Il numero massimo di definizioni di parametri personalizzati e parametri estesi che una singola destinazione può contenere è 2.000. Per ogni parametro esteso o parametro personalizzato inviato a ciascuna destinazione, ogni combinazione di nome e valore della dimensione conta come parametro esteso per tale limite. Questo vale anche come metrica CloudWatch personalizzata per la determinazione dei prezzi.

Quando invii metriche estese a CloudWatch, puoi utilizzare la console CloudWatch RUM per creare CloudWatch allarmi su di esse.

Le metriche estese vengono addebitate come CloudWatch metriche personalizzate. Per ulteriori informazioni, consulta la sezione Prezzi di Amazon CloudWatch.

Le metriche estese per tutti i nomi delle metriche che è possibile inviare tramite i monitoraggi dell'app supportano le dimensioni seguenti. Questi nomi del parametro sono elencati in CloudWatch metriche che puoi raccogliere con RUM CloudWatch .

- `BrowserName`

  Valori delle dimensioni di esempio: `Chrome`, `Firefox`, `Chrome Headless`
- `CountryCode`: utilizza il formato ISO-3166 con codici a due lettere.

  Valori delle dimensioni di esempio: `US`, `JP`, `DE`
- `DeviceType`

  Valori delle dimensioni di esempio: `desktop`, `mobile`, `tablet`, `embedded`
- `FileType`

Valori delle dimensioni di esempio: `Image`, `Stylesheet`

- `OSName`

  Valori delle dimensioni di esempio: `Linux`, `Windows, iOS`, `Android`

- `PageId`

Configurazione di parametri estesi mediante la console

Per utilizzare la console a cui inviare metriche estese CloudWatch, procedi nel seguente modo.

Per inviare metriche estese a CloudWatch Evidently, devi utilizzare le AWS API o al AWS CLI posto della console. Per informazioni sull'utilizzo delle AWS API per inviare metriche estese a uno o all'altro CloudWatch o a Evidently, consulta e. PutRumMetricsDestinationBatchCreateRumMetricDefinitions

Per utilizzare la console per configurare un monitor di app e inviare metriche estese RUM a CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, RUM.
3. Scegli List view (Visualizzazione elenco), quindi scegli il nome del monitoraggio dell'app che deve inviare le metriche.
4. Scegli la scheda Configuration (Configurazione) e infine RUM extended metrics (Metriche RUM estese).
5. Scegli Send metrics (Invia parametri).
6. Seleziona uno o più nomi di parametri da inviare con dimensioni aggiuntive.
7. Seleziona uno o più fattori da utilizzare come dimensioni per queste metriche. Man mano che effettui le tue scelte, il numero di metriche estese creati viene visualizzato in Number of extended metrics (Numero di parametri estesi).

   Questo numero viene calcolato moltiplicando il numero di nomi dei parametri scelti per il numero delle diverse dimensioni create. Questo numero rappresenta il numero di metriche personalizzate che ti vengono addebitate. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatchdei prezzi di Amazon.

   a. Per inviare un parametro con l'ID pagina come dimensione, scegli Browse for page ID (Individua l'ID pagina), quindi seleziona gli ID pagina da utilizzare.

b. Per inviare un parametro con il tipo di dispositivo come dimensione, scegli Desktop devices (Dispositivi desktop) o Mobile and tablets (Dispositivi mobili e tablet).

c. Per inviare un parametro con il sistema operativo come dimensione, seleziona uno o più sistemi operativi in Operating system (Sistema operativo).

d. Per inviare un parametro con il tipo di browser come dimensione, seleziona uno o più browser in Browsers (Browser).

e. Per inviare un parametro con la geolocalizzazione come dimensione, seleziona una o più posizioni in Locations (Posizioni).

   Nell'elenco verranno visualizzate solo le posizioni da cui questo monitoraggio dell'app ha riportato le metriche.

8. Quando hai finito di selezionare le opzioni, scegli Send metrics (Invia metriche).
9. (Facoltativo) Nell'elenco Extended metrics (Metriche estese), crea un allarme che controlli uno dei parametri scegliendo Create alarm (Crea allarme) nella riga relativa al parametro.

   Per informazioni generali sugli CloudWatch allarmi, consulta Utilizzo degli CloudWatch allarmi Amazon. Per un tutorial sull'impostazione di un allarme in base a una metrica estesa CloudWatch RUM, vedi. Tutorial: creazione di un parametro esteso e del relativo allarme

Interruzione dell'invio di metriche estese

Per utilizzare la console al fine di interrompere l'invio di metriche estese

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, RUM.
3. Scegli List view (Visualizzazione elenco), quindi scegli il nome del monitoraggio dell'app che deve inviare le metriche.
4. Scegli la scheda Configuration (Configurazione) e infine RUM extended metrics (Metriche RUM estese).
5. Seleziona una o più combinazioni di nome e dimensione dei parametri per interrompere l'invio. Quindi scegli Actions (Operazioni), Delete (Elimina).

Tutorial: creazione di un parametro esteso e del relativo allarme

Questo tutorial dimostra come impostare una metrica estesa a cui inviare CloudWatch e quindi come impostare un allarme su quella metrica. In questo tutorial, crei una metrica che tiene traccia JavaScript degli errori nel browser Chrome.

Per configurare il parametro esteso e impostare un allarme su di esso

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, RUM.
3. Scegli List view ((Visualizzazione elenco), quindi scegli il nome del monitoraggio dell'app che deve inviare il parametro.
4. Scegli la scheda Configuration (Configurazione) e infine RUM extended metrics (Metriche RUM estese).
5. Scegli Send metrics (Invia parametri).
6. Seleziona JS ErrorCount.
7. In Browsers (Browser), seleziona Chrome.

   Questa combinazione di JS ErrorCount e Chrome invierà una metrica estesa a CloudWatch. La metrica conta JavaScript gli errori solo per le sessioni utente che utilizzano il browser Chrome. Il nome della metrica sarà JsErrorCounte il nome della dimensione sarà Browser.
8. Scegli Send metrics (Invia parametri).
9. Nell'elenco delle metriche estese, scegli Crea allarme nella riga visualizzata JsErrorCountsotto Nome e in cui viene visualizzato Chrome. BrowserName
10. In Specificare metrica e condizioni, conferma che il nome e i BrowserNamecampi della metrica siano precompilati con i valori corretti.
11. In Statistic (Statistica), seleziona la statistica che desideri utilizzare per l'allarme. Il valore Average (Media) è ideale per questo tipo di parametro di conteggio.
12. Per Period (Periodo), selezionare 5 minutes (5 minuti).
13. In Condizioni, effettuare le seguenti operazioni:

    - Scegli Static (Statico).
    - Scegli Greater (Maggiore) per specificare che l'allarme deve passare allo stato ALARM quando il numero di errori è superiore alla soglia specificata.

- In than... (di...), inserisci il numero per la soglia di allarme. L'allarme entra nello stato ALARM quando il numero di errori in un periodo di 5 minuti supera questo numero.

14. (Facoltativo) Per impostazione predefinita, l'allarme passa allo stato ALARM non appena il numero di errori supera la soglia impostata durante un periodo di 5 minuti. Facoltativamente, è possibile modificare questa impostazione in modo che l'allarme entri nello stato ALARM solo se questo numero viene superato per un periodo superiore a 5 minuti.

    Per eseguire questa operazione, scegli Additional configuration (Configurazione aggiuntiva), quindi in Datapoints to alarm (Data point per allarme) specifica quanti periodi di 5 minuti devono avere un numero di errori superiore alla soglia per attivare l'allarme. Ad esempio, puoi selezionare 2 su 2 per attivare l'allarme solo quando due periodi consecutivi di 5 minuti superano la soglia, oppure 2 su 3 attivarlo se due dei tre periodi consecutivi di 5 minuti superano la soglia.

    Per ulteriori informazioni in merito a questo tipo di valutazione degli allarmi, consulta Valutazione di un allarme.
15. Seleziona Successivo.
16. In Configure actions (Configura operazioni), specifica cosa deve accadere quando l'allarme si attiva. Per ricevere una notifica con Amazon SNS, procedi come segue:

    - Scegliere Add notification (Aggiungi notifica).
    - Scegli In allarme.
    - Seleziona un argomento SNS esistente o crearne uno nuovo. Se ne crei uno nuovo, specifica un nome e aggiungi almeno un indirizzo e-mail.
17. Seleziona Successivo.
18. Inserisci un nome e, facoltativamente, una descrizione per l'allarme, quindi scegli Next (Successivo).
19. Verifica i dettagli e scegli Create alarm (Crea allarme).

Protezione e riservatezza dei dati con RUM CloudWatch

Il modello di responsabilità AWS condivisa si applica alla protezione e alla privacy dei dati in Amazon CloudWatch RUM. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutto il AWS cloud. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa,

consulta il post sul blog The AWS Shared Responsibility Model e sul GDPR sul AWS Security Blog. Per ulteriori risorse sulla conformità ai requisiti GDPR, consulta la sezione Centro generale sulla protezione dei dati (GDPR).

Amazon CloudWatch RUM genera un frammento di codice da incorporare nel codice del sito Web o dell'applicazione Web, in base all'immissione dei dati dell'utente finale che desideri raccogliere. Il client web, scaricato e configurato dallo snippet di codice, utilizza cookie (o tecnologie simili) per aiutare a raccogliere i dati dell'utente finale. L'uso di cookie (o tecnologie simili) è soggetto alle norme sulla privacy dei dati in alcune giurisdizioni. Prima di utilizzare Amazon CloudWatch RUM, ti consigliamo vivamente di valutare i tuoi obblighi di conformità ai sensi della legge applicabile, inclusi eventuali requisiti legali applicabili per fornire avvisi sulla privacy legalmente adeguati e ottenere i consensi necessari per l'uso dei cookie e il trattamento (inclusa la raccolta) dei dati degli utenti finali. Per ulteriori informazioni su come il client web utilizza i cookie (o tecnologie simili) e sui dati degli utenti finali raccolti dal client web, consulta e. Informazioni raccolte dal client web RUM CloudWatch CloudWatch Cookie del client web RUM (o tecnologie simili)

Consigliamo vivamente di non inserire mai informazioni identificative sensibili, ad esempio i numeri di account dei clienti, indirizzi email o altre informazioni personali in campi a formato libero. Tutti i dati che inserisci in Amazon CloudWatch RUM o in altri servizi potrebbero essere inclusi nei log di diagnostica.

CloudWatch Cookie del client web RUM (o tecnologie simili)

Per impostazione predefinita, il client web CloudWatch RUM raccoglie determinati dati sulle sessioni utente. È possibile scegliere di abilitare i cookie in modo che il client Web raccolga un ID utente e un ID di sessione che persistono durante i caricamenti delle pagine. L'ID utente viene generato in modo casuale da RUM.

Se questi cookie sono abilitati, RUM è in grado di visualizzare i seguenti tipi di dati quando si visualizza il pannello di controllo RUM per questo monitor dell'app.

- Dati aggregati basati su ID utente, come il numero di utenti univoci e il numero di utenti diversi che hanno riscontrato un errore.
- Dati aggregati basati su ID di sessione, come il numero di sessioni e il numero di sessioni che hanno riscontrato un errore.
- Il percorso dell'utente, ovvero la sequenza di pagine che include ogni sessione utente campionata.

> **Important**
>
> Se non abiliti questi cookie (o tecnologie simili), il client web registra comunque alcune informazioni sulle sessioni dell'utente finale, come il tipo/versione del browser, il tipo/versione del sistema operativo, il tipo di dispositivo e così via. Questi sono raccolti per fornire informazioni aggregate specifiche della pagina, come i parametri vitali web, le visualizzazioni di pagina e le pagine che hanno riscontrato errori. Per ulteriori informazioni sui tipi di dati registrati, consulta Informazioni raccolte dal client web RUM CloudWatch .

Informazioni raccolte dal client web RUM CloudWatch

Questa sezione documenta lo PutRumEventsschema, che definisce la struttura dei dati che è possibile raccogliere dalle sessioni utente utilizzando CloudWatch RUM.

Una PutRumEventsrichiesta invia a CloudWatch RUM una struttura di dati con i seguenti campi.

- L'ID di questo batch di eventi RUM
- Dettagli del monitor dell'app, che includono i seguenti:
  - ID dell'app del monitor
  - Versione dell'applicazione monitorata
- Dettagli dell'utente, che comprendono i seguenti. Questo viene raccolto solo se sul monitor dell'app sono abilitati i cookie.
  - ID utente generato dal client web
  - ID sessione
- La gamma di Eventi RUM in questo batch.

Schema eventi RUM

La struttura di ciascun evento RUM include i campi riportati di seguito.

- L'ID dell'evento
- Un Timestamp
- Il tipo di evento
- L'agente utente

- Metadati
- Dettagli evento RUM

Metadati degli eventi RUM

I metadati includono metadati di pagina, metadati dell'agente utente, metadati di geolocalizzazione e metadati di dominio.

Metadati della pagina

I metadati della pagina includono quanto segue:

- ID pagina
- Titolo pagina
- ID pagina principale. - Questo viene raccolto solo se sul monitor di app sono abilitati i cookie.
- Profondità di interazione - Questo viene raccolto solo se sul monitor di app sono abilitati i cookie.
- Tag di pagina: puoi aggiungere tag agli eventi delle pagine per raggruppare le pagine. Per ulteriori informazioni, consulta Utilizzo dei gruppi di pagine.

Metadati dell'agente utente

I metadati dell'agente utente includono quanto segue:

- Lingua browser
- Nome browser
- Versione browser
- Nome del sistema operativo
- Versione del sistema operativo
- Tipo dispositivo
- Tipo di piattaforma

Metadati di geolocalizzazione

I metadati di geolocalizzazione includono quanto segue:

- Codice del paese

- Codice di suddivisione

Metadati del dominio

I metadati del dominio includono il dominio URL.

Dettagli evento RUM

I dettagli di un evento seguono uno dei seguenti tipi di schemi, a seconda del tipo di evento.

Evento di avvio della sessione

Questo evento non contiene campi. Questo viene raccolto solo se sul monitor dell'app sono abilitati i cookie.

Schema di visualizzazione pagina

Un'evento di Visualizzazione pagina contiene le seguenti proprietà. È possibile disattivare la raccolta di viste pagina configurando il client Web. Per ulteriori informazioni, consulta la documentazione del client web CloudWatch RUM.

Nome	Type	Descrizione
ID pagina	Stringa	ID che rappresenta in modo univoco questa pagina all'interno dell'applicazione. Per impostazione predefinita, questo è il percorso URL.
ID pagina principale	Stringa	L'ID della pagina in cui si trovava l'utente quando si è spostato alla pagina corrente. Questo viene raccolto solo se sul monitor dell'app sono abilitati i cookie.
Profondità di interazione	Stringa	Questo viene raccolto solo se sul monitor dell'app sono abilitati i cookie.

JavaScript schema di errore

JavaScript gli eventi di errore generati dall'agente contengono le seguenti proprietà. Il client web raccoglie questi eventi solo se si è scelto di raccogliere gli errori di telemetria.

Nome	Type	Descrizione
Tipi di errore	Stringa	Nome dell'errore, se esiste. Per ulteriori informazioni, consulta Error.prototype.name. Alcuni browser potrebbero non supportare i tipi di errore.
Messaggio di errore	Stringa	Messaggio di errore. Per ulteriori informazioni, consulta Error.prototype.message. Se il campo di errore non esiste, questo è il messaggio dell'evento di errore. Per ulteriori informazioni, vedere ErrorEvent. I messaggi di errore potrebbero non essere coerenti su diversi browser.
Traccia della pila	Stringa	La traccia della pila dell'errore, se esistente, è troncata a 150 caratteri. Per ulteriori informazioni, consulta Error.prototype.stack. Alcuni browser potrebbero non supportare le tracce della pila.

Schema eventi DOM

Gli eventi DOM (Document Object Model) generati dall'agente contengono le seguenti proprietà. Questi eventi non vengono raccolti per impostazione predefinita. Vengono raccolti solo se si attiva la telemetria delle interazioni. Per ulteriori informazioni, consulta la documentazione del client web CloudWatch RUM.

Nome	Type	Descrizione
Evento	Stringa	Il tipo di evento DOM, come click, scorrimento o passaggio del mouse. Per ulteriori informazioni, consulta Riferimento evento.
Elemento	Stringa	Il tipo di elemento DOM

Nome	Type	Descrizione
Elemento ID	Stringa	Se l'elemento che ha generato l'evento ha un ID, questa proprietà memorizza tale ID. Per ulteriori informazioni, consulta Element.id.
CSSLocator	Stringa	Il localizzatore CSS utilizzato per identificare l'elemento DOM.
InteractionId	Stringa	Un ID univoco per l'interazione tra l'utente e l'interfaccia utente.

Schema eventi di navigazione

Gli eventi di navigazione vengono raccolti solo se il monitor dell'app ha attivato la telemetria delle prestazioni.

Gli eventi di navigazione usano API di Temporizzazione di navigazione Livello 1 e Tempistica di navigazione Livello 2. Le API di livello 2 non sono supportate su tutti i browser, quindi questi campi più recenti sono facoltativi.

> **Note**
>
> Le metriche del timestamp si basano sul DOM. HighResTimestamp Con le API di livello 2, tutte le temporizzazioni sono per impostazione predefinita relative al `startTime`. Ma per il livello 1, il parametro `navigationStart` viene sottratto dai parametri del timestamp per ottenere valori relativi. Tutti i valori del timestamp sono espressi in millisecondi.

Gli eventi di navigazione contengono le seguenti proprietà.

Nome	Type	Descrizione	Note
Tipo di iniziatore	Stringa	Rappresenta il tipo di risorsa che ha avviato l'evento delle prestazioni.	Value (Valore): «navigazione»

Nome	Type	Descrizione	Note
			Livello 1: «navigazione» Livello 2: entryData.initiatorType

Nome	Type	Descrizione	Note
Tipo di navigazione	Stringa	Rappresenta il tipo di navigazione. Questo attributo non è obbligatorio.	Valore Il valore deve essere uno dei seguenti: • `navigate` è una navigazione avviata scegliendo un collegamento, inserendo un URL nella barra degli indirizzi di un browser, l'invio di un modulo o l'inizializzazione tramite un'operazione di script diversa da `reload` o `back_forward` . • `reload` è una navigazio

Nome	Type	Descrizione	Note
			ne attraverso l'operazione di ricaricamento del browser o `location.reload()`. • `back_forward` è una navigazione attraverso l'operazione di attraversamento della cronologia del browser. • `prerender` è una navigazione avviata da un suggerimento prerender. Per ulteriori informazioni,

Nome	Type	Descrizione	Note
			consulta Prerender.
startTime	Numero	Indica quando viene attivato l'evento.	Value (Valore): 0 Livello 1:entrydata .navigati onStart - entryData .Navigati onStart Livello 2: entryData .startTime

Nome	Type	Descrizione	Note
unloadEventStart	Numero	Indica l'ora in cui il documento precedente nella finestra ha iniziato a scaricarsi dopo che l'evento `unload` è stato lanciato.	Value (Valore): Se non esiste un documento precedente o se il documento precedente o uno dei reindirizzamenti necessari non hanno la stessa origine, il valore restituito è 0. Livello 1: `entryData.unloadEventStart > 0 ? entryData.unloadEventStart - entryData.navigationStart : 0` Livello 2: EntryData.

Nome	Type	Descrizione	Note
			unloadEvent Start

Nome	Type	Descrizione	Note
promptForUnload	Numero	Il tempo necessario per scaricare il documento. In altre parole, il tempo tra `unloadEventStart` e `unloadEventEnd`. `UnloadEventEnd` rappresenta il momento in millisecondi al termine del gestore eventi di scarico.	Value (Valore): Se non esiste un documento precedente o se il documento precedente o uno dei reindirizzamenti necessari non hanno la stessa origine, il valore restituito è 0. Livello 1: EntryData.unloadEventEnd - Dati di ingresso.unloadEventStart Livello 2: EntryData.unloadEventEnd - Dati di ingresso.unloadEventStart

Nome	Type	Descrizione	Note
Redirect Count	Numero	Un numero che rappresenta il numero di reindirizzamenti dall'ultima navigazione non reindirizzata nel contesto di navigazione corrente. Questo attributo non è obbligatorio.	Value (Valore): Se non è presente alcun reindirizzamento o se è presente un reindirizzamento non della stessa origine del documento di destinazione, il valore restituito è 0. Livello 1: Non disponibile Livello 2: entryData.redirectCount

Nome	Type	Descrizione	Note
RedirectStart	Numero	L'ora in cui inizia il primo reindirizzamento HTTP.	Value (Valore): Se non è presente alcun reindirizzamento o se è presente un reindirizzamento non della stessa origine del documento di destinazione, il valore restituito è 0. Livello 1: `entryData.redirectStart > 0 ? entryData.redirectStart - entryData.navigationStart : 0` Livello 2: entryData.redirectStart

Nome	Type	Descrizione	Note
Tempo di reindirizzamento	Numero	Il tempo impiegato per il reindirizzamento HTTP. Questa è la differenza tra `redirectStart` e `redirectEnd` .	Livello 1:: entryData .redirectEnd - entryData .redirectStart Livello 2:: entryData .redirectEnd - entryData .redirectStart

Nome	Type	Descrizione	Note
workerStart	Numero	Questa è una proprietà dell'interfaccia `PerformanceResourceTiming` . Segna l'inizio dell'operazione del thread di lavoro. Questo attributo non è obbligatorio.	Value (Valore): Se un thread Service Worker è già in esecuzione o immediata mente prima di avviare il thread Service Worker, questa proprietà restituis ce l'ora immediata mente prima dell'invio di `FetchEven t` . Restituis ce 0 se la risorsa non viene intercettata da un Service Worker. Livello 1: Non disponibile Livello 2: entryData .workerStart

Nome	Type	Descrizione	Note
workerTime	Numero	Se la risorsa viene intercettata da un Service Worker, restituisce il tempo necessario per l'operazione del thread di lavoro. Questo attributo non è obbligatorio.	Livello 1: Non disponibile Livello 2: `entryData.workerStart > 0 ? entryData.fetchStart - entryData.workerStart : 0`
fetchStart	Numero	Il momento in cui il browser è pronto a recuperare il documento utilizzando una richiesta HTTP. Questo è prima di controllare qualsiasi cache dell'applicazione.	Livello 1: `: entryData.fetchStart > 0 ? entryData.fetchStart - entryData.navigationStart : 0` Livello 2: entrydata .fetchStart

Nome	Type	Descrizione	Note
domainLookupStart	Numero	Ora di avvio della ricerca del dominio.	Value (Valore): Se viene utilizzata una connessione persistente o se le informazioni sono memorizzate in una cache o in una risorsa locale, il valore sarà lo stesso `fetchStart` . Livello 1: `entryData.domainLookupStart > 0 ? entryData.domainLookupStart - entryData.navigationStart : 0` Livello 2: EntryData.

Nome	Type	Descrizione	Note
			domainLookupStart
dns	Numero	Il tempo necessario per la ricerca del dominio.	Value (Valore): Se le risorse e i record DNS sono memorizzati nella cache, il valore previsto è 0. Livello 1: EntryData.domainLookupEnd - Dati di ingresso.domainLookupStart Livello 2: EntryData.domainLookupEnd - Dati di ingresso.domainLookupStart
nextHopProtocol	Stringa	Stringa che rappresenta il protocollo di rete utilizzato per recuperare la risorsa. Questo attributo non è obbligatorio.	Livello 1: Non disponibile Livello 2: EntryData.nextHopProtocol

Nome	Type	Descrizione	Note
connectStart	Numero	Il tempo immediatamente prima che l'agente utente inizi a stabilire la connessione al server per recuperare il documento.	Value (Valore): Se viene utilizzata una connessione persistente RFC2616 o se il documento corrente viene recuperato dalle cache delle applicazioni o dalle risorse locali pertinenti, questo attributo restituisce il valore di `domainLookupEnd` . Livello 1: `entryData.connectStart > 0 ? entryData.connectStart - entryData`

Nome	Type	Descrizione	Note
			`.navigationStart : 0` Livello 2: entryData .connectStart
connect	Numero	Misura il tempo necessario per stabilire le connessioni di trasporto o per eseguire l'autenticazione SSL. Include anche il tempo bloccato impiegato quando sono presenti troppe richieste simultanee emesse dal browser.	Livello 1: entryData .connectEnd - EntryData .connectStart Livello 2: entryData .connectEnd - EntryData .connectStart
secureConnectionStart	Numero	Se lo schema URL della pagina corrente è «https», questo attributo restituisce il tempo immediatamente prima che l'agente del'utente avvii il processo di handshake per proteggere la connessione corrente. Restituisce 0 se non viene utilizzato HTTPS. Per ulteriori informazioni sugli schemi URL, consulta Rappresentazione dell'URL.	Formula: EntryData. secureConnectionStart

Nome	Type	Descrizione	Note
Tempo tls	Numero	Il tempo necessario per completare una handshake SSL.	Livello 1: `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0` Livello 2: `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0`

Nome	Type	Descrizione	Note
Inizio richiesta	Numero	Il tempo immediatamente prima che l'agente utente inizi a richiedere la risorsa dal server, dalle cache delle applicazioni pertinenti o dalle risorse locali.	Livello 1: `: entryData.requestStart > 0 ? entryData.requestStart - entryData.navigationStart : 0` Livello 2: entryData .requestStart
timeToFirstByte	Numero	Il tempo necessario per ricevere il primo byte di informazioni dopo la richiesta. Questa tempo è relativo al `startTime` .	Livello 1: entryData .responseStart - entryData .requestStart Livello 2: entryData .responseStart - entryData .requestStart

Nome	Type	Descrizione	Note
Avvio risposta	Numero	Il tempo immediatamente dopo che il parser HTTP dell'agente utente riceve il primo byte della risposta dalle cache dell'applicazione pertinenti, dalle risorse locali o dal server.	Livello 1: `entryData.responseStart > 0 ? entryData.responseStart - entryData.navigationStart : 0` Livello 2: entryData .response Start

Nome	Type	Descrizione	Note
responseTime	Stringa	Il tempo necessario per ricevere una risposta completa sotto forma di byte dalle cache delle applicazioni pertinenti, dalle risorse locali o dal server.	Livello 1: `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0` Livello 2: `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0`

Nome	Type	Descrizione	Note
domIntera ctive	Numero	Il momento in cui il parser ha terminato il suo lavoro sul documento principale e viene costruito il DOM HTML. In questo momento, il `Document.readyState` cambia in «interattivo» e l'evento `readystatechange` corrispondente viene generato.	Livello 1: `entryData .domInter active > 0 ? entryData .domInter active - entryData .navigati onStart : 0` Livello 2:Entrydata .com Interacti ve

Nome	Type	Descrizione	Note
domContentLoadedEventStart	Numero	Rappresenta il valore temporale uguale al tempo immediatamente precedente all'attivazione dell'ContentLoaded evento DOM da parte dell'agente utente nel documento corrente. L'ContentLoaded evento DOM si attiva quando il documento HTML iniziale è stato completamente caricato e analizzato. A questo punto, il documento HTML principale ha terminato l'analisi, il browser inizia a costruire l'albero di rendering e le sottofonti devono ancora essere caricate. Questo non attende che i fogli di stile, le immagini e i fotogrammi secondari finiscano il caricamento.	Livello 1: `entryData.domContentLoadedEventStart > 0 ? entryData.domContentLoadedEventStart - entryData.navigationStart : 0` Livello 2: EntryData.domContentLoadedEventStart

Nome	Type	Descrizione	Note
domContentLoaded	Numero	Questa ora di inizio e di fine della costruzione dell'albero di rendering è contrassegnata dalla `domContentLoadedEventStart` e `domContentLoadedEventEnd`. Consente a CloudWatch RUM di tracciare l'esecuzione. Questa proprietà è la differenza tra `domContentLoadedStart` e `domContentLoadedEnd`. Durante questo periodo, DOM e CSSOM sono pronti. Questa proprietà attende l'esecuzione dello script, ad eccezione degli script asincroni e creati dinamicamente. Se gli script dipendono dai fogli di stile, `domContentLoaded` aspetta anche sui fogli di stile. Non aspetta le immagini. **Note** I valori effettivi di `domContentLoadedStart` e `domContentLoadedEnd` approssimati a `domContentLoaded` nel pannello di rete di Google Chrome. Indica il tempo di costruzione dell'albero di rendering HTML DOM + CSSOM dall'inizio del processo di caricamento della pagina. Nel caso dei parametri di navigazione, il valore di `domContentLoaded` rappresenta la differenza tra i valori iniziale e finale, ovvero il tempo necessario per scaricare le sottofonti e la costruzione dell'albero di rendering.	Livello 2: EntryData.domContentLoadedEventEnd - Dati di ingresso.domContentLoadedEventStart Livello 2: EntryData.domContentLoadedEventEnd - Dati di ingresso.domContentLoadedEventStart

Nome	Type	Descrizione	Note
DOM completo	Numero	Il tempo immediatamente prima che il browser imposti la disponibilità corrente del documento corrente per il completamento. A questo punto, il caricamento di sottofonti, come le immagini, è completo. Ciò include il tempo impiegato per scaricare contenuti bloccanti come CSS e sincroni. JavaScript Questo si avvicina a `loadTime` nel pannello di Rete di Google Chrome.	Livello 1: `entryData .domCompl ete > 0 ? entryData .domCompl ete - entryData .navigati onStart : 0` Livello 2: Entrydata .com completo
domProces singTime	Numero	Il tempo totale tra la risposta e l'avvio dell'even to di caricamento.	Livello 1: EntryData . loadEvent Start - EntryData .responseEnd Livello 2: EntryData . loadEvent Start - EntryData .responseEnd

Nome	Type	Descrizione	Note
loadEvent Start	Numero	Il tempo immediatamente prima dell'evento `load` del documento corrente viene attivato.	Livello 1: `entryData .loadEventStart > 0 ? entryData .loadEventStart - entryData .navigationStart : 0` Livello 2: EntryData . loadEvent Start
loadEvent Time	Numero	La differenza tra `loadEventStart` e `loadEventEnd` . Durante questo periodo verranno attivate funzioni o logiche aggiuntive in attesa di questo evento di caricamento.	Livello 1: EntryData . loadEvent End - Dati di ingresso. loadEvent Start Livello 2: EntryData . loadEvent End - Dati di ingresso. loadEvent Start

Nome	Type	Descrizione	Note
durata	Stringa	La durata è il tempo di caricamento totale della pagina. Registra i tempi per il download della pagina principale e di tutte le sue sottofonti sincrone e anche per il rendering della pagina. Le risorse asincrone come gli script continuano a essere scaricate in un secondo momento. Questa è la differenza tra le proprietà `loadEventEnd` e `startTime` .	Livello 1: EntryData . loadEvent End - EntryData .navigati onStart Livello 2: entryData .duration
Dimensioni intestazione	Numero	Restituisce la differenza tra `transferSize` e `encodedBodySize` . Questo attributo non è obbligatorio.	Livello 1: Non disponibile Livello 2: entryData .transferSize - EntryData . encodedBo dySize Livello 2: entryData .transferSize - EntryData . encodedBo dySize

Nome	Type	Descrizione	Note
Rapporto di compressione	Numero	Il rapporto di `encodedBodySize` e `decodedBodySize`. Il valore di `encodedBodySize` è la dimensione compressa della risorsa escludendo le intestazioni HTTP. Il valore di `decodedBodySize` è la dimensione decompressa della risorsa escludendo le intestazioni HTTP. Questo attributo non è obbligatorio.	Livello 1: Non disponibile. Livello 2: `entryData.encodedBodySize > 0 ? entryData.decodedBodySize / entryData.encodedBodySize : 0`
navigationTimingLevel	Numero	La versione dell'API di tempistica di navigazione.	Value (Valore): 1 o 2

Schemi di eventi delle risorse

Gli eventi delle risorse vengono raccolti solo se il monitor dell'app ha attivato la telemetria delle prestazioni.

Le metriche del timestamp si basano su The DOM typedef. HighResTimeStamp Con le API di livello 2, per impostazione predefinita, tutte le tempistiche sono relative al `startTime`. Ma per le API di livello 1, il parametro `navigationStart` viene sottratto dai parametri del timestamp per ottenere valori relativi. Tutti i valori del timestamp sono espressi in millisecondi.

Gli eventi delle risorse generati dall'agente contengono le seguenti proprietà.

Nome	Type	Descrizione	Note
targetUrl	Stringa	Restituisce l'URL della risorsa.	Formula: EntryData .name
Tipo di iniziatore	Stringa	Rappresenta il tipo di risorsa che ha avviato l'evento della risorsa per le prestazioni.	Value (Valore): «risorsa» Formula: entryData .initiatorType
durata	Stringa	Restituisce la differenza tra proprietà `responseEnd` e `startTime` . Questo attributo non è obbligatorio.	Formula: entryData .duration
Dimensioni di trasferimento	Numero	Restituisce la dimensione (in ottetti) della risorsa recuperata, inclusi i campi di intestazi one della risposta e il corpo del payload della risposta. Questo attributo non è obbligatorio.	Formula: entryData .transferSize
TipoFile	Stringa	Estensioni derivate dal pattern URL di destinazi one.	

Il più grande schema di eventi di pittura contentful

I più grandi eventi di pittura contentful contengono le seguenti proprietà.

Questi eventi vengono raccolti solo se il monitor dell'app ha attivato la telemetria delle prestazioni.

Nome	Descrizione		
Valore	Per ulteriori informazioni,		

Nome	Descrizione		
	consulta Web Vitals (Informazioni sulla salute Web).		

Primo evento di ritardo di input

I primi eventi di ritardo di input contengono le seguenti proprietà.

Questi eventi vengono raccolti solo se il monitor dell'app ha attivato la telemetria delle prestazioni.

Nome	Descrizione		
Valore	Per ulteriori informazioni, consulta Web Vitals (Informazioni sulla salute Web).		

Evento di spostamento cumulativo del layout

Gli eventi cumulativi di spostamento di layout contengono le seguenti proprietà.

Questi eventi vengono raccolti solo se il monitor dell'app ha attivato la telemetria delle prestazioni.

Nome	Descrizione		
Valore	Per ulteriori informazioni, consulta Web Vitals (Informazioni sulla salute Web).		

Evento HTTP

Gli eventi HTTP possono contenere le seguenti proprietà. Conterrà un campo `Response` o campo `Error`, ma non entrambi.

Questi eventi vengono raccolti solo se il monitor dell'app ha attivato la telemetria HTTP.

Nome	Descrizione
Richiesta	Il campo della richiesta include quanto segue: • Il campo `Method`, che può avere valori come GET, POST e così via. • L'URL
Risposta	La risposta include quanto segue: • Stato, come 2xx, 4xx o 5xx • Testo di stato
Errore	La sezione di errore include quanto segue: • Type • Messaggio • Nome file • Numero riga • Numero colonna • Traccia della pila

Schemi di eventi X-Ray

Questi eventi vengono raccolti solo se il monitor dell'app ha attivato il tracciamento X-Ray.

Per informazioni sugli schemi degli eventi X-Ray trace, consulta documenti di un segmento AWS X-Ray.

Tempi di cambio percorso per applicazioni a pagina singola

In un'applicazione tradizionale a più pagine, quando un utente richiede il caricamento di nuovi contenuti, l'utente sta effettivamente richiedendo una nuova pagina HTML dal server. Di conseguenza, il client web CloudWatch RUM acquisisce i tempi di caricamento utilizzando le normali metriche dell'API prestazionale.

Tuttavia, le applicazioni Web a pagina singola utilizzano JavaScript Ajax per aggiornare l'interfaccia senza caricare una nuova pagina dal server. Gli aggiornamenti a pagina singola non vengono registrati dall'API di sincronizzazione del browser, ma utilizzano invece i tempi di modifica del percorso.

CloudWatch RUM supporta il monitoraggio sia del caricamento di pagine complete dal server sia degli aggiornamenti a pagina singola, con le seguenti differenze:

- Per la tempistica del cambio del percorso, non ci sono parametri forniti dal browser come `tlsTime`, `timeToFirstByte` e così via.
- Per la tempistica del cambio del percorso, il campo `initiatorType` sarà `route_change`.

Il client web CloudWatch RUM ascolta le interazioni dell'utente che possono portare a una modifica del percorso e, quando tale interazione utente viene registrata, il client Web registra un timestamp. Quindi la tempistica del cambio del percorso inizierà se entrambe le condizioni seguenti sono vere:

- Un'API della cronologia del browser (eccetto i pulsanti avanti e indietro del browser) è stata utilizzata per eseguire il cambio di percorso.
- La differenza tra il tempo di rilevamento del cambio del percorso e il timestamp dell'ultima interazione dell'utente è inferiore a 1000 ms. In questo modo si evita l'asimmetria dei dati.

Quindi, una volta che inizia la sincronizzazione del cambio del percorso, tale tempistica viene completata se non ci sono richieste AJAX e mutazioni DOM in corso. Quindi il timestamp dell'ultima attività completata verrà utilizzato come timestamp di completamento.

La tempistica per il cambio de percorso scade se ci sono richieste AJAX o mutazioni DOM in corso per più di 10 secondi (per impostazione predefinita). In questo caso, il client web CloudWatch RUM non registrerà più la tempistica di questa modifica del percorso.

Di conseguenza, la durata di un evento di cambio del percorso viene calcolata come segue:

```
(time of latest completed activity) - (latest user interaction timestamp)
```

Gestisci le tue applicazioni che utilizzano RUM CloudWatch

Segui i passaggi descritti in queste sezioni per gestire l'uso di RUM da parte delle tue applicazioni. CloudWatch

Come faccio a trovare uno snippet di codice che ho già generato?

Per trovare un frammento di codice CloudWatch RUM che hai già generato per un'applicazione, segui questi passaggi.

Per trovare uno snippet di codice già generato

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, RUM.
3. Scegli Visualizzazione elenco.
4. Accanto al nome del monitor dell'app, scegli Visualizza JavaScript.
5. Nel riquadro JavaScript Snippet, scegli Copia negli appunti.

Modificare l'applicazione

Per modificare le impostazioni di un monitor app, seguire i seguenti passaggi. È possibile modificare qualsiasi impostazione tranne il nome del monitor dell'app.

Per modificare il modo in cui l'applicazione utilizza RUM CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, RUM.
3. Scegli Visualizzazione elenco.
4. Scegli il pulsante accanto al nome dell'applicazione, quindi scegli Operazioni,Modificare.
5. Modificare tutte le impostazioni tranne il nome dell'applicazione. Per ulteriori informazioni in merito all'impostazione , consulta Fase 2: creazione di un monitor dell'app.
6. Al termine, scegli Save (Salva).

   La modifica delle impostazioni cambia lo snippet di codice. Ora è necessario incollare lo snippet di codice aggiornato nella propria applicazione.

7. Dopo aver creato il frammento di JavaScript codice, scegli Copia negli appunti o Scarica, quindi scegli Fine.

   Per avviare il monitoraggio con le nuove impostazioni, inserire lo snippet di codice nell'applicazione. Inserire lo snippet di codice all'interno dell'elemento `<head>` della prorpia applicazione, prima dell'elemento `<body>` o qualsiasi altro tag `<script>`.

Smetti di usare CloudWatch RUM o elimina il monitor di un'app

Per smettere di usare CloudWatch RUM con un'applicazione, rimuovi il frammento di codice generato da RUM dal codice dell'applicazione.

Per eliminare un monitor dell'app RUM, seguire i seguenti passaggi.

Per eliminare un monitor dell'app

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Application Signals, RUM.
3. Scegli Visualizzazione elenco.
4. Scegli il pulsante accanto al nome dell'applicazione, quindi scegli Operazioni, Elimina.
5. Nel campo di conferma immetti **`Delete`** e quindi scegli Delete (Elimina).
6. Se non l'hai già fatto, elimina lo snippet di codice CloudWatch RUM dal codice dell'applicazione.

CloudWatch Quote RUM

CloudWatch Il RUM ha le seguenti quote.

Risorsa	Quota predefinita
Monitor app	20 per account È possibile richiedere un aumento della quota.
Frequenza di acquisizione RUM	50 PutRumEventsrichieste al secondo (TPS). È possibile richiedere un aumento della quota.

Risoluzione dei problemi RUM CloudWatch

Questa sezione contiene suggerimenti per aiutarti a risolvere CloudWatch i problemi RUM.

Non ci sono dati per la mia applicazione

In primo luogo, accertarsi che lo snippet di codice sia stato inserito correttamente nell'applicazione. Per ulteriori informazioni, consulta Passaggio 4: inserire lo snippet di codice nella propria applicazione.

Se questo non è il problema, forse non c'è ancora stato traffico verso la propria applicazione. Generare traffico accedendo all'applicazione nello stesso modo in cui farebbe un utente.

I dati hanno smesso di essere registrati per la mia applicazione

L'applicazione potrebbe essere stata aggiornata e ora non contiene più un frammento di codice CloudWatch RUM. Controllare il codice dell'applicazione.

Un'altra possibilità è che qualcuno abbia aggiornato lo snippet di codice ma non abbia inserito lo snippet aggiornato nell'applicazione. Trovare lo snippet di codice corretto corrente seguendo le istruzioni in Come faccio a trovare uno snippet di codice che ho già generato? e confrontarlo con lo snippet di codice incollato nella propria applicazione.

Monitoraggio della rete

Gli argomenti di questa sezione descrivono le funzionalità di monitoraggio della CloudWatch rete e di Internet fornite da Amazon CloudWatch Internet Monitor e Amazon CloudWatch Network Monitor. Questi servizi ti aiutano a ottenere visibilità operativa sulla rete e sulle prestazioni di Internet e sulla disponibilità delle applicazioni ospitate su AWS.

- Internet Monitor utilizza i dati di connettività AWS acquisiti dalla sua impronta di rete globale per calcolare una base di prestazioni e disponibilità per il traffico connesso a Internet. È possibile visualizzare una visione globale dei modelli di traffico e degli eventi sanitari e approfondire facilmente le informazioni sugli eventi. Puoi anche ricevere avvisi per eventi sanitari su Internet che influiscono sui client delle tue applicazioni. Inoltre, puoi utilizzare le informazioni fornite da Internet Monitor per esplorare potenziali miglioramenti all'esperienza dei tuoi clienti, utilizzando Amazon CloudFront o effettuando il routing attraverso diversi Regioni AWS canali.
- Network Monitor utilizza un approccio basato su agenti completamente gestito per consentirti di tracciare e visualizzare la latenza e la perdita di pacchetti per le connessioni di rete ibride. Per raccogliere misurazioni e consentire a Network Monitor di creare avvisi relativi agli eventi sanitari per l'applicazione, è necessario creare delle sonde che vengono inviate dalle risorse ospitate su indirizzi IP di destinazione locali. AWS Non è necessario installare agenti aggiuntivi per monitorare le prestazioni della rete. Come con Internet Monitor, è possibile impostare avvisi e soglie, ottenere informazioni per risolvere rapidamente i problemi e quindi intervenire per migliorare l'esperienza dell'utente finale.

Argomenti

Utilizzo di Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor offre visibilità su come i problemi di Internet influiscono sulle prestazioni e sulla disponibilità tra le applicazioni ospitate AWS e gli utenti finali. Può ridurre il tempo necessario per diagnosticare i problemi di Internet da giorni a minuti. Internet Monitor utilizza i dati di connettività AWS acquisiti dalla sua impronta di rete globale per calcolare una base di prestazioni e disponibilità per il traffico connesso a Internet. Si tratta degli stessi dati AWS utilizzati per monitorare l'operatività e la disponibilità di Internet. Con queste misurazioni come base, Monitor Internet ti aiuta

a comprendere la presenza di problemi significativi per gli utenti finali (client) nelle diverse aree geografiche in cui viene eseguita l'applicazione.

Nella CloudWatch console Amazon, puoi visualizzare una visione globale dei modelli di traffico e degli eventi sanitari e approfondire facilmente le informazioni sugli eventi, con diverse granularità geografiche (località). È possibile visualizzare chiaramente l'impatto e individuare le ubicazioni e le reti dei client (ASN, in genere provider di servizi Internet o ISP) interessati. Se Internet Monitor determina che un problema di disponibilità o prestazioni di Internet è causato da un ASN specifico o dalla AWS rete, fornisce tali informazioni.

Funzionalità principali di Monitor Internet

- Monitor Internet suggerisce approfondimenti e consigli che possono aiutarti a migliorare l'esperienza degli utenti finali. Puoi scoprire, quasi in tempo reale, come migliorare la latenza prevista della tua applicazione passando a utilizzare altri servizi o reindirizzando il traffico verso il tuo carico di lavoro tramite diversi. Regioni AWS
- Con Monitor Internet, puoi identificare rapidamente cosa influisce sulle prestazioni e sulla disponibilità dell'applicazione, in modo da poter individuare e risolvere i problemi.
- Internet Monitor pubblica le misurazioni di Internet su CloudWatch Logs and CloudWatch Metrics, per supportare l'utilizzo di CloudWatch strumenti con informazioni sanitarie per località e ASN (provider di servizi Internet) specifici per l'applicazione. Facoltativamente, puoi anche pubblicare le misurazioni via Internet su Amazon S3.
- Internet Monitor invia eventi sanitari ad Amazon EventBridge in modo che tu possa configurare le notifiche. Se un problema è causato dalla AWS rete, riceverai automaticamente anche una AWS Health Dashboard notifica con le misure adottate AWS per mitigare il problema.

Come usare Monitor Internet

Per utilizzare Internet Monitor, è necessario creare un monitor e associarvi le risorse dell'applicazione (VPC, Network Load Balancer, CloudFront distribuzioni o WorkSpaces directory) per consentire a Internet Monitor di sapere dove si trova il traffico rivolto a Internet dell'applicazione. Internet Monitor pubblica quindi le misurazioni Internet relative alle reti urbane, AWS ovvero alle ubicazioni dei client e agli ASN (in genere provider di servizi Internet o ISP), da cui i client accedono all'applicazione. Per ulteriori informazioni, consulta Come funziona Amazon CloudWatch Internet Monitor. Per iniziare a utilizzare Monitor Internet, consulta Guida introduttiva ad Amazon CloudWatch Internet Monitor tramite la console.

Indice

Supportato Regioni AWS per Amazon CloudWatch Internet Monitor

I Regioni AWS paesi in cui è supportato Amazon CloudWatch Internet Monitor sono elencati in questa sezione. Per un elenco aggiornato delle regioni in cui è supportato Internet Monitor, incluse le regioni con attivazione, consulta gli endpoint e le quote di Amazon CloudWatch Internet Monitor nell'Amazon Web Services General Reference.

Tieni presente che Internet Monitor archivia i dati di un monitor solo nella Regione AWS zona in cui lo crei, sebbene un monitor possa includere risorse in più regioni.

Nome della regione (supporto Opt-in)	Regione
Africa (Città del Capo)	`af-south-1`

Nome della regione (supporto Opt-in)	Regione
Asia Pacifico (Hong Kong)	`ap-east-1`
Asia Pacific (Hyderabad)	`ap-south-2`
Asia Pacifico (Giacarta)	`ap-southeast-3`
Asia Pacifico (Melbourne)	`ap-southeast-4`
Europa (Milano)	`eu-south-1`
Europa (Spagna)	`eu-south-2`
Europa (Zurigo)	`eu-central-2`
Medio Oriente (Bahrein)	`me-south-1`
Medio Oriente (Emirati Arabi Uniti)	`me-central-1`

Nome della regione (supporto predefinito)	Regione
Stati Uniti orientali (Ohio)	`us-east-2`
Stati Uniti orientali (Virginia settentrionale)	`us-east-1`
Stati Uniti occidentali (California settentrionale)	`us-west-1`
Stati Uniti occidentali (Oregon)	`us-west-2`
Asia Pacifico (Mumbai)	`ap-south-1`
Asia Pacific (Osaka)	`ap-northeast-3`
Asia Pacific (Seul)	`ap-northeast-2`
Asia Pacifico (Singapore)	`ap-southeast-1`
Asia Pacifico (Sydney)	`ap-southeast-2`

Nome della regione (supporto predefinito)	Regione
Asia Pacifico (Tokyo)	`ap-northeast-1`
Canada (Centrale)	`ca-central-1`
Europa (Francoforte)	`eu-central-1`
Europa (Irlanda)	`eu-west-1`
Europe (London)	`eu-west-2`
Europe (Paris)	`eu-west-3`
Europa (Stoccolma)	`eu-north-1`
Sud America (San Paolo)	`sa-east-1`

Prezzi di Amazon CloudWatch Internet Monitor

Con Amazon CloudWatch Internet Monitor, non ci sono costi iniziali o impegni a lungo termine. Il prezzo di Monitor Internet prevede due componenti: una tariffa per risorsa monitorata e una tariffa per rete urbana. Una rete urbana è la posizione da cui i client accedono alle risorse dell'applicazione e la rete (ASN, ad esempio un provider di servizi Internet o ISP) tramite la quale i client accedono alle risorse. Tieni presente che ti vengono addebitati anche CloudWatch prezzi standard per i log e qualsiasi metrica, dashboard, allarmi o approfondimenti aggiuntivi che crei.

Quando crei un monitor, scegli una percentuale di traffico da monitorare. Per tenere sotto controllo la fattura, puoi anche impostare un limite per il numero massimo di reti urbane da monitorare. Puoi aggiornare la percentuale di traffico da monitorare o il limite massimo per le reti urbane in qualsiasi momento modificando il monitor. Sono incluse le prime 100 reti urbane (su tutti i monitor per account). Dopodiché, pagherai solo per il numero effettivo aggiuntivo di reti urbane monitorate, fino al numero massimo.

Paghi solo il numero effettivo aggiuntivo di reti urbane monitorate, fino al numero massimo, senza alcun costo per le prime 100 reti urbane (su tutti i monitor per account). Dalla fattura mensile viene detratto un importo fisso equivalente al costo di 100 reti urbane.

Ad esempio, una grande azienda globale potrebbe scegliere di monitorare il 100% del traffico connesso a Internet e impostare un massimo di 50.000 reti urbane, per un monitor con una sola risorsa. Supponendo che il traffico raggiunga le 50.000 reti urbane, quella parte della fattura ammonterebbe a circa 2.700 USD al mese. Per un'altra azienda, con meno aree geografiche, con un monitor con una sola risorsa e 200 reti urbane, questa parte della bolletta si aggirerebbe intorno ai 13 USD al mese. Per ulteriori informazioni, consulta Scelta del limite massimo per le reti urbane.

Puoi provare diverse opzioni con il calcolatore dei prezzi. Per scoprire le opzioni di prezzo, nella CloudWatch pagina Calcolatore prezzi, scorri verso il basso fino a Internet Monitor.

Per ulteriori informazioni su Internet Monitor e CloudWatch sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Componenti e termini per Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor utilizza o fa riferimento a quanto segue.

Monitoraggio

Un monitor include le risorse per una singola applicazione per la quale desideri visualizzare le misurazioni delle prestazioni e della disponibilità di Internet e sulla quale desideri ricevere avvisi relativi agli eventi di integrità. Quando si crea un monitor per un'applicazione, si aggiungono risorse per l'applicazione per definire le città (posizioni) che Monitor Internet deve monitorare. Monitor Internet utilizza i modelli di traffico delle risorse dell'applicazione che aggiungi in modo da poter pubblicare le misurazioni delle prestazioni e della disponibilità di Internet specifiche solo per le posizioni e gli ASN (in genere, i provider di servizi Internet o ISP) che comunicano con l'applicazione. In altre parole, le risorse che aggiungi creano un ambito delle reti urbane che desideri che Monitor Internet monitori e per le quali pubblichi le misurazioni.

Risorsa aggiunta al monitoraggio («risorsa monitorata»)

Una risorsa aggiunta a un monitor è una «risorsa monitorata» in Internet Monitor. In altre parole:

- Ogni VPC aggiunto in una Regione è una risorsa monitorata. Quando aggiungi un VPC, Internet Monitor monitora il traffico di qualsiasi applicazione connessa a Internet nel VPC, ad esempio un'applicazione ospitata su un'istanza Amazon EC2, dietro un Network Load Balancer o un container. AWS Fargate
- Ogni Network Load Balancer aggiunto in una regione è una risorsa monitorata.
- Ogni WorkSpaces directory che aggiungi in una regione è una risorsa monitorata.
- Ogni CloudFront distribuzione aggiunta è una risorsa monitorata.

Numeri di sistema autonomi (ASN)

In Monitor Internet, l'ASN si riferisce in genere a un provider di servizi Internet (ISP), come Verizon o Comcast. Un ASN è un provider di rete utilizzato da un client per accedere all'applicazione Internet. Un sistema autonomo (AS) è un insieme di prefissi IP (Internet Routable Internet Protocol) che appartengono a una rete o a un insieme di reti gestite, controllate e supervisionate da un'unica organizzazione.

Rete urbana (posizione e ASN)

Una rete urbana è la posizione (ad esempio una città) da cui i client accedono alle risorse dell'applicazione e l'ASN, in genere un provider di servizi Internet (ISP), tramite il quale i client accedono alle risorse. Per facilitare il controllo della bolletta, è possibile impostare un limite per il numero massimo di reti urbane che Internet Monitor può monitorare per ogni monitor. Pagherai solo per il numero effettivo di reti urbane monitorate, fino al numero massimo. Per ulteriori informazioni, consulta Scelta di un limite massimo di reti urbane.

Misurazioni Internet

Internet Monitor pubblica le misurazioni di Internet nei file di registro nella sezione CloudWatch Logs ogni cinque minuti per le 500 principali reti urbane (sedi dei clienti e ASN, in genere provider di servizi Internet o ISP) del tuo account. Queste misurazioni quantificano il punteggio delle prestazioni, il punteggio di disponibilità, i byte trasferiti (byte in entrata e in uscita) e il tempo di andata e ritorno delle reti urbane dell'applicazione. Si tratta di misurazioni per le reti urbane specifiche dei tuoi VPC, Network Load Balancer, distribuzioni o elenchi. CloudFront WorkSpaces Facoltativamente, puoi scegliere di pubblicare misurazioni ed eventi Internet per tutte le reti urbane monitorate (fino al limite di servizio di 500.000 reti urbane) in un bucket Amazon S3.

Metriche

Internet Monitor genera metriche aggregate per le metriche, per il traffico globale verso l' CloudWatch applicazione e il traffico globale verso ciascuna di esse. Regione AWS Per ulteriori informazioni, consulta Utilizzo di CloudWatch metriche con Amazon CloudWatch Internet Monitor.

Evento di integrità

Monitor Internet crea automaticamente eventi di integrità per avvisare l'utente in caso di problemi specifici che riguardano l'applicazione. Monitor Internet rileva i problemi relativi a Internet, come l'aumento della latenza di rete, in tutto il mondo. Quindi utilizza le misurazioni storiche di Internet relative all'impronta dell'infrastruttura AWS globale per calcolare l'impatto dei problemi attuali sull'applicazione e crea eventi sanitari. Monitor Internet, per impostazione predefinita, crea

eventi di integrità in base alle soglie di impatto globale e locale. Per ulteriori informazioni sulla configurazione delle soglie, vedere Modifica delle soglie degli eventi di integrità.

Ogni evento di integrità include informazioni sulle reti urbane interessate. È possibile visualizzare gli eventi sanitari nella CloudWatch console o utilizzando l' AWS SDK o AWS CLI con le azioni dell'API Internet Monitor. Internet Monitor invia anche EventBridge notifiche ad Amazon per eventi sanitari. Per ulteriori informazioni, consulta Quando Monitor Internet crea e risolve eventi di integrità.

Evento su Internet

Internet Monitor visualizza informazioni sui recenti eventi sanitari globali, denominati eventi Internet, su una mappa meteorologica Internet disponibile per tutti i AWS clienti. Non è necessario creare un monitor in Internet Monitor per visualizzare la mappa meteorologica di Internet. A differenza degli eventi relativi alla salute, gli eventi su Internet non sono specifici per i singoli clienti o per il traffico delle loro applicazioni. Per ulteriori informazioni, consulta Mappa meteorologica globale di Internet in Amazon CloudWatch Internet Monitor.

Soglie

Monitor Internet crea eventi di integrità in base alle soglie di impatto globale e locale. È possibile modificare le soglie predefinite e configurare altre opzioni, come la disattivazione delle soglie locali. Per ulteriori informazioni sulla configurazione delle soglie, vedere Modifica delle soglie degli eventi di integrità.

Punteggi di prestazioni e disponibilità

Analizzando i dati AWS raccolti, Internet Monitor è in grado di rilevare quando le prestazioni e la disponibilità dell'applicazione sono diminuite rispetto alle baseline stimate calcolate da Internet Monitor. Per facilitare la visualizzazione di questi cali, Monitor Internet riporta le informazioni sotto forma di punteggi. Un punteggio delle prestazioni rappresenta la percentuale stimata di traffico che non registra un calo delle prestazioni. Analogamente, un punteggio di disponibilità rappresenta la percentuale stimata di traffico che non registra un calo di disponibilità. Per ulteriori informazioni, vedere Come AWS calcola i punteggi di prestazioni e disponibilità.

Byte trasferiti e byte monitorati trasferiti

I byte trasferiti sono il numero totale di byte di traffico in ingresso e in uscita tra un'applicazione e la rete urbana (ovvero la posizione AWS e l'ASN, in genere il provider di servizi Internet) in cui i client accedono a un'applicazione. I byte monitorati trasferiti sono un parametro simile, ma includono solo i byte per il traffico monitorato.

Tempo di andata e ritorno

Il tempo di andata e ritorno (RTT) è il tempo impiegato da una richiesta dell'utente del client per restituire una risposta all'utente. Quando l'RTT (round-trip time) viene aggregato tra le posizioni dei clienti (città o altre aree geografiche), il valore viene ponderato in base alla quantità di traffico delle applicazioni generata da ciascuna posizione del cliente.

Mappa meteorologica globale di Internet in Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor mostra una mappa meteorologica globale di Internet disponibile per tutti i AWS clienti. Per visualizzare la mappa, nella CloudWatch console Amazon, accedi a Internet Monitor.

La mappa evidenzia gli eventi Internet («interruzioni») in tutto il mondo che riguardano AWS i clienti, con particolare riferimento alle città e alle reti specifiche (ASN, in genere provider di servizi Internet) in cui si verificano problemi di prestazioni o disponibilità. La mappa meteorologica di Internet include gli eventi Internet delle ultime 24 ore.

Non è necessario creare un monitor in Internet Monitor per visualizzare la mappa meteorologica di Internet. A differenza degli eventi relativi alla salute in Internet Monitor, gli eventi su Internet non sono specifici per i singoli clienti o per il traffico delle loro applicazioni.

Sulla mappa meteorologica di Internet, puoi scegliere un evento su Internet per conoscerne i dettagli. Per un evento online, puoi visualizzare l'ora di inizio, l'ora di fine (se l'evento è terminato), lo stato attuale (Attivo o Risolto) e il tipo di problema di interruzione (Disponibilità o Prestazioni). Per ulteriori informazioni su come viene creata la mappa meteorologica di Internet e su cosa è inclusa, consulta le domande frequenti sulla mappa meteorologica globale di Internet.

Per visualizzare e utilizzare informazioni dettagliate specifiche sul traffico delle applicazioni e sulle ubicazioni dei client, è possibile configurare facilmente un monitor in Internet Monitor per la propria applicazione. Potrai quindi visualizzare i modelli e gli eventi relativi a prestazioni e disponibilità, attuali e storici, oltre a ricevere avvisi relativi agli eventi sanitari, personalizzati in base all'impatto dell'applicazione e ai clienti. La mappa meteorologica di Internet offre una visione d'insieme, mentre un monitor specifico filtra le informazioni solo in base alle misurazioni e ai dettagli pertinenti all'applicazione. Con un monitor, puoi anche esplorare le metriche storiche e ottenere consigli per migliorare l'esperienza del cliente con la tua applicazione. Per ulteriori informazioni, consulta Guida introduttiva ad Amazon CloudWatch Internet Monitor tramite la console.

Come funziona Amazon CloudWatch Internet Monitor

Questa sezione fornisce informazioni su come funziona Amazon CloudWatch Internet Monitor. Ciò include le descrizioni di come AWS raccoglie i dati che utilizza per aiutare a rilevare problemi di connettività su Internet e di come vengono calcolati i punteggi di prestazioni e disponibilità.

Indice

In che modo Internet Monitor si concentra solo sull'impronta del traffico delle applicazioni

Internet Monitor concentra il monitoraggio solo sul sottoinsieme di Internet a cui accedono gli utenti delle tue AWS risorse, anziché monitorare ampiamente il tuo sito Web da ogni regione del mondo come fanno altri strumenti. Si tratta di una soluzione economica e accessibile per aziende di piccole e grandi dimensioni.

Internet Monitor utilizza le stesse potenti sonde e gli stessi algoritmi di rilevamento dei problemi di cui AWS si avvale internamente e avvisa l'utente dei problemi di connettività che influiscono sull'applicazione creando eventi sanitari in Internet Monitor. Monitor Internet ti consente quindi di accedere alla mappa risultante delle prestazioni e della disponibilità, sovrapponendo il profilo di traffico creato dai visualizzatori attivi, in base alle risorse dell'applicazione.

Utilizzando queste informazioni, Monitor Internet mostra solo gli eventi pertinenti (ovvero gli eventi che si verificano in luoghi con visualizzatori attivi) e solo l'impatto di tali eventi sul volume complessivo di visualizzatori. Pertanto, l'impatto di un evento, in termini percentuali, si basa sul traffico totale a livello mondiale.

Internet Monitor pubblica su CloudWatch Logs le misurazioni Internet ogni cinque minuti per le 500 principali reti cittadine (sedi dei clienti e ASN, in genere provider di servizi Internet o ISP) che

inviano traffico a ciascun monitor. Facoltativamente, puoi decidere di pubblicare misurazioni ed eventi Internet per tutte le reti urbane monitorate (fino al limite di servizio di 500.000 reti urbane) in un bucket Amazon S3. Per ulteriori informazioni, consulta Pubblicazione di misurazioni Internet su Amazon S3 in Amazon CloudWatch Internet Monitor.

I vantaggi di Monitor Internet includono quanto segue:

- L'utilizzo di Monitor Internet non comporta carichi o costi aggiuntivi sull'applicazione ospitata in AWS.
- Non è necessario includere il codice di misurazione delle prestazioni nelle risorse lato client o nell'applicazione.
- È possibile visualizzare le prestazioni e la disponibilità di Internet per la porzione a cui è connessa l'applicazione, comprese le informazioni relative al cosiddetto "ultimo miglio".

Tieni presente che, poiché Internet Monitor crea misurazioni in base alle tue AWS risorse, Internet Monitor crea solo eventi specifici per il traffico delle tue applicazioni. I problemi globali di Internet in generale non vengono segnalati. Inoltre, quando la sede del servizio è una Regione AWS, le misurazioni e gli eventi emessi sono progettati per rappresentare la connettività a livello regionale e non rappresentano accuratamente la connettività tra la posizione dell'utente finale e una zona di disponibilità.

Come AWS misura i problemi di connettività e calcola le misurazioni

Amazon CloudWatch Internet Monitor utilizza i dati di connettività Internet tra diversi CloudFront punti di presenza (PoP) Regioni AWS e Amazon in diverse sedi dei clienti tramite numeri di sistema autonomi (ASN), in genere provider di servizi Internet (ISP). Si tratta dei dati di connettività che vengono utilizzati internamente dagli AWS operatori, su base giornaliera, per rilevare in modo proattivo i problemi di connettività su Internet globale.

Per tutti Regione AWS, sappiamo quali parti di Internet comunicano con la Regione e facciamo quanto segue:

- Monitoriamo attivamente tali porzioni di Internet, con una finestra continua di 30 giorni.
- Utilizziamo sonde di rete e di protocollo di livello superiore, inclusi probing in entrata e in uscita.

AWS dispone di sonde attive e passive che misurano la latenza (prestazioni) al 90° percentile e la raggiungibilità (disponibilità) da ogni punto del CloudFront servizio all' Regione AWS intera rete Internet. I modelli anomali di connettività tra un servizio e la sede del cliente vengono monitorati e quindi segnalati come avvisi al cliente.

Calcolo della disponibilità e RTT

Il tempo di andata e ritorno (RTT) è il tempo impiegato da una richiesta dell'utente per restituire una risposta. Quando il tempo di andata e ritorno viene aggregato tra le posizioni degli utenti finali, il valore viene ponderato in base alla quantità di traffico generato da ciascuna posizione dell'utente finale.

Ad esempio, con due posizioni degli utenti finali, una che serve il 90% del traffico con un RTT di 5 ms e l'altra che serve il 10% del traffico con un RTT di 10 ms, il risultato è un RTT aggregato di 5,5 ms (che deriva da 5 ms * 0,9 + 10 ms * 0,1).

Tieni presente che esistono differenze tra le risorse sulla misurazione della latenza dell'ultimo miglio. Per le misurazioni della latenza di Internet Monitor, i VPC, i Network Load Balancer e le WorkSpaces directory non includono la latenza dell'ultimo miglio.

Calcolo dei punteggi di prestazioni e disponibilità

AWS dispone di dati storici sostanziali sulle prestazioni e sulla disponibilità di Internet tra AWS i servizi e le diverse reti cittadine (sedi e ASN). Applicando l'analisi statistica ai dati, Monitor Internet può individuare il momento in cui le prestazioni e la disponibilità dell'applicazione sono diminuite grazie a un confronto con una baseline stimata calcolata precedentemente. Per facilitare la visualizzazione di questi cali, le informazioni vengono riportate sotto forma di punteggi di integrità, vale a dire con un punteggio delle prestazioni e un punteggio di disponibilità.

I punteggi di integrità vengono calcolati con diverse granularità. Con la massima granularità, calcoliamo il punteggio di integrità per un'area geografica, come una città o un'area metropolitana, e un ASN (una rete urbana). Inoltre, sommiamo i punteggi di integrità individuali ai punteggi di integrità globali per un'applicazione in un monitor. Se visualizzi i punteggi delle prestazioni o della disponibilità senza filtrare per aree geografiche o provider di servizi specifici, Monitor Internet ti mostra i punteggi di integrità globali.

I punteggi di integrità globali si riferiscono all'intera applicazione per il periodo di tempo specificato. Quando il punteggio di prestazioni o disponibilità delle coppie città-rete dell'applicazione raggiunge o scende al di sotto della soglia di integrità corrispondente in termini di prestazioni o disponibilità, Monitor Internet attiva un evento di integrità. Per impostazione predefinita, la soglia è del 95% sia per le prestazioni complessive che per la disponibilità. Monitor Internet crea anche eventi di integrità in base a soglie locali, se l'opzione è abilitata, come di default, in base ai valori configurati dall'utente. Per ulteriori informazioni

sulla configurazione delle soglie degli eventi di integrità, consulta Modifica delle soglie degli eventi di integrità.

Quando si esaminano le informazioni nel monitor e nei file di log per esaminare i problemi e saperne di più, puoi filtrare per città (posizione), reti (ASN o provider di servizi Internet) o entrambe. Quindi, puoi utilizzare i filtri per visualizzare i punteggi di integrità per diverse città, ASN o coppie città-rete, a seconda dei filtri scelti.

- Un punteggio di disponibilità rappresenta la percentuale stimata di traffico che non registra un calo di disponibilità. Monitor Internet stima la percentuale di traffico che subisce un calo rispetto al traffico totale visualizzato e alle misurazioni delle metriche di disponibilità. Ad esempio, un punteggio di disponibilità del 99% per una coppia di posizione dell'utente finale e del servizio equivale all'1% del traffico che ha subito un calo di disponibilità per tale coppia.
- Un punteggio delle prestazioni rappresenta la percentuale di traffico che non registra un calo delle prestazioni. Ad esempio, un punteggio delle prestazioni del 99% per una coppia di posizione dell'utente finale e del servizio equivale all'1% del traffico che ha subito un calo di prestazioni per tale coppia.

Calcolo di TTFB e RTT (latenza)

Il time to first byte (TTFB) si riferisce al tempo che intercorre tra il momento in cui un client effettua una richiesta e il momento in cui riceve il primo byte di informazioni dal server. AWS i calcoli per TTFB misurano il tempo trascorso da Amazon EC2 o Amazon CloudFront al nodo di misurazione di Internet Monitor (incluso l'ultimo miglio del nodo). In altre parole, Internet Monitor misura il tempo trascorso dall'utente alla regione Amazon EC2 per TTFB per EC2 e dall'utente a TTFB per. CloudFront CloudFront

Per quanto riguarda il tempo di andata e ritorno (RTT), Monitor Internet include il tempo dalla rete urbana (ovvero la posizione del client e l'ASN, in genere un provider di servizi Internet), come mappato dall'indirizzo IP pubblico, alla Regione AWS. Ciò significa che Monitor Internet non offre visibilità all'ultimo miglio per gli utenti che accedono a Internet da un gateway o da una VPN.

Tieni presente che esistono differenze tra le risorse sulla misurazione della latenza dell'ultimo miglio. Per le misurazioni della latenza di Internet Monitor, i VPC, i Network Load Balancer e le directory non includono la latenza dell'ultimo miglio. WorkSpaces

Internet Monitor include informazioni TTFB medie nella sezione Suggerimenti per l'ottimizzazione del traffico della scheda Traffic Insights della CloudWatch dashboard, per

aiutarti a valutare le opzioni per le diverse configurazioni dell'applicazione che possono migliorare le prestazioni.

Misurazioni e aggregazioni regionali e delle zone di disponibilità

Sebbene Internet Monitor aggreghi le misurazioni e condivida l'impatto a livello regionale, calcola l'impatto a livello di zona di disponibilità (AZ). Ciò significa che, se, per un evento, è interessata solo una zona di zona e la maggior parte del traffico attraversa quella zona, l'impatto sul traffico è evidente. Tuttavia, per lo stesso evento, se il traffico dell'applicazione non fluisce attraverso una zona di zona interessata, non si riscontra alcun impatto.

Tieni presente che questo vale solo per le risorse che non sono WorkSpaces directory. WorkSpaces le directory vengono misurate solo a livello regionale.

Precisione della geolocalizzazione in Monitor Internet

Per informazioni sulla posizione, Internet Monitor utilizza i dati di geolocalizzazione IP forniti da. MaxMind L'accuratezza delle informazioni sulla posizione nelle misurazioni di Internet Monitor dipende dalla precisione dei dati. MaxMind

Tieni presente che le misurazioni `Metro` del livello potrebbero non essere accurate per le località al di fuori degli Stati Uniti.

Quando Monitor Internet crea e risolve eventi di integrità

Monitor Internet crea e chiude gli eventi di integrità per il traffico dell'applicazione monitorato in base alle soglie correnti impostate. Monitor Internet ha una configurazione di soglia predefinita ma è possibile impostare una configurazione personalizzata per le soglie. Monitor Internet determina l'impatto complessivo dei problemi di connettività sull'applicazione e l'impatto sulle aree locali in cui l'applicazione dispone di client e crea eventi di integrità quando vengono superate le soglie.

Internet Monitor calcola l'impatto dei problemi di connettività sulla posizione di un client in base ai dati storici sulle prestazioni e sulla disponibilità di Internet per il traffico di rete tramite AWS il servizio. Applica le informazioni pertinenti all'applicazione, in base alle posizioni geografiche degli ASN e dei servizi in cui i client utilizzano l'applicazione: le coppie città-rete interessate. Le posizioni sono determinate dalle risorse che si aggiungono al monitor. Monitor Internet utilizza quindi l'analisi statistica per rilevare quando le prestazioni e la disponibilità sono diminuite, con ripercussioni sull'esperienza del client dell'applicazione.

I cali di prestazioni e disponibilità calcolati da Monitor Internet sono rappresentati come la percentuale di traffico che non registra alcun calo. L'impatto è l'opposto: è una rappresentazione

di quanto un problema sia complesso per gli utenti finali di un cliente. Quindi, se ad esempio si verifica un calo di disponibilità globale del 93%, l'impatto corrispondente sarebbe del 7%.

Quando il punteggio di prestazioni o disponibilità delle coppie città-rete dell'applicazione raggiunge o scende al di sotto della soglia di integrità corrispondente in termini di prestazioni o disponibilità, Monitor Internet attiva un evento di integrità. Per impostazione predefinita, la soglia è del 95% sia per le prestazioni che per la disponibilità. I valori per raggiungere o scendere al di sotto della soglia sono cumulativi, quindi è possibile che diversi eventi più piccoli si combinino per raggiungere la percentuale di soglia o che un singolo evento raggiunga o scenda al di sotto del livello di soglia.

Finché i punteggi relativi alle prestazioni o alla disponibilità che hanno innescato l'evento sono pari o inferiori alla percentuale di soglia dell'evento di integrità corrispondente in termini di impatto complessivo, l'evento di integrità rimane attivo. Quando il punteggio o i punteggi combinati che hanno innescato l'evento superano la soglia, Monitor Internet risolve l'evento di integrità.

Monitor Internet crea eventi di integrità anche in base alle soglie locali e alla percentuale di traffico complessivo su cui un problema ha un impatto. È possibile configurare le opzioni per le soglie locali o disattivare del tutto le soglie locali.

Per ulteriori informazioni sulla configurazione delle soglie degli eventi di integrità, consulta Modifica delle soglie degli eventi di integrità.

Tempistica del rapporto sugli eventi di integrità

Monitor Internet raccoglie tutti i segnali relativi ai problemi di Internet tramite un aggregatore, al fine di creare in pochi minuti eventi di stato nei monitoraggi.

Quando possibile, Internet Monitor analizza l'origine di un evento sanitario per determinare se è stato causato da AWS o da un ASN. L'analisi degli eventi di integrità continua dopo la risoluzione di un evento. Monitor Internet può aggiornare gli eventi con nuove informazioni per un massimo di un'ora.

Come funziona Monitor Internet con il traffico IPv4 e IPv6

Monitor Internet misura l'integrità di una rete solo su IPv4 e mostra gli eventi relativi allo stato di integrità e i parametri di disponibilità e prestazioni, se si indirizza il traffico verso quella rete su qualsiasi famiglia di IP (IPv4 o IPv6). Se servi traffico proveniente da una risorsa dual-stack, ad esempio una CloudFront distribuzione dual-stack, Internet Monitor segnala un evento di integrità e mostra un calo del punteggio di prestazioni o di disponibilità solo se il traffico IPv4 presenta gli stessi problemi per la risorsa del traffico IPv6.

Tieni presente che i parametri di Monitor Internet per i byte complessivi in entrata e in uscita riflettono accuratamente tutto il traffico Internet (IPv4 e IPv6).

In che modo Internet Monitor seleziona il sottoinsieme di reti urbane da includere

Quando si imposta un limite massimo per il numero di reti urbane monitorate dal monitor o si sceglie una percentuale di traffico da monitorare, Internet Monitor sceglie le reti urbane da includere (monitorare) in base al volume di traffico recente più elevato.

Ad esempio, se imposti un limite massimo di 100 reti urbane, Internet Monitor monitora (fino a) 100 reti urbane in base al traffico dell'applicazione durante un periodo recente di un'ora. In particolare, Internet Monitor monitora le 100 principali reti urbane che hanno registrato il maggior traffico nell'ultima finestra di un'ora prima dell'ultima finestra di un'ora.

Per illustrare ciò, supponiamo che l'ora corrente sia le 14:30. In questo scenario, il traffico visualizzato sul monitor è stato acquisito tra le 13:00 e le 14:00 e la misurazione del volume di traffico utilizzata da Internet Monitor per determinare le 100 principali reti urbane è stata acquisita tra le 12:00 e le 13:00.

Come viene creata la mappa meteorologica globale di Internet (Domande frequenti)

La mappa meteorologica CloudWatch Internet di Amazon Internet Monitor è disponibile sulla console Internet Monitor per tutti i AWS clienti autenticati. Questa sezione include dettagli su come viene creata la mappa meteorologica di Internet e su come utilizzarla.

Cos'è la mappa meteorologica di Internet Monitor?

La mappa meteorologica di Internet fornisce una rappresentazione visiva dei problemi di Internet in tutto il mondo. Evidenzia le sedi dei clienti interessati, ovvero le città più l'ASN (in genere i provider di servizi Internet). La mappa mostra una combinazione di problemi di disponibilità e prestazioni che hanno recentemente influito sull'esperienza Internet dei clienti nelle principali sedi e servizi dei clienti a livello globale. AWS

Da dove provengono i dati per la mappa?

I dati si basano su una combinazione di sondaggi attivi e passivi su Internet. Per saperne di più su come Internet Monitor misura i dati, puoi leggere la sezione Come AWS misura i problemi di connettività.

Con che frequenza viene aggiornata la mappa?

La mappa meteorologica di Internet viene aggiornata ogni 15 minuti.

Quali reti vengono monitorate per rilevare eventuali interruzioni?

AWS tiene traccia delle reti di tutto il mondo che rappresentano importanti prefissi IP utilizzati dai clienti per effettuare connessioni Internet. AWS Monitoriamo le interruzioni nelle sedi dei clienti che sono le principali interlocutori per volume di traffico inviato e ricevuto dalla rete. AWS

Cosa determina se un evento Internet è incluso nella mappa?

Ecco alcuni criteri di alto livello che utilizziamo per determinare se un evento Internet è incluso nella mappa meteorologica di Internet:

- AWS rileva la presenza di un evento di disponibilità o di performance.
- Se l'evento è di breve durata, ad esempio, dura meno di 5 minuti, lo ignoriamo.
- Quindi, se l'evento si svolge in una sede del cliente classificata come top talker, viene considerato un'interruzione.

Quali soglie vengono utilizzate per la mappa meteorologica di Internet?

Le soglie per determinare le interruzioni non sono statiche per la mappa meteorologica di Internet. Internet Monitor determina cosa costituisce un evento in base al rilevamento di una deviazione dai valori previsti. Puoi saperne di più su come funziona esaminando in che modo Internet Monitor determina quando creare eventi sanitari per i monitor creati con il servizio. Quando crei un monitor, Internet Monitor genera misurazioni dello stato del traffico Internet specifiche per il traffico delle tue applicazioni. Internet Monitor avvisa inoltre l'utente in caso di problemi relativi al traffico Internet dell'applicazione in caso di problemi che influiscono sul traffico Internet dell'applicazione.

Cosa posso fare con questi dati?

La mappa meteorologica di Internet fornisce un breve riepilogo dei principali eventi Internet accaduti in tutto il mondo nelle ultime 24 ore. Ti aiuta a farti un'idea dell'esperienza di monitoraggio di Internet, senza dover registrare il tuo traffico Internet su Internet Monitor. Per sfruttare appieno il potenziale delle funzionalità di monitoraggio di Internet AWS e personalizzarle per le applicazioni e i servizi ospitati su Internet AWS, puoi creare un monitor in Internet Monitor.

Quando create un monitor, consentite a Internet Monitor di identificare i percorsi Internet specifici che influiscono sui client delle vostre applicazioni e ottenete l'accesso a caratteristiche e funzionalità che possono aiutarvi a migliorare l'esperienza del cliente. Riceverai inoltre notifiche proattive in caso di nuovi problemi relativi a Internet che hanno un impatto specifico sul traffico e sui client delle applicazioni.

Come posso ottenere maggiori dettagli sugli eventi?

Fai clic su un'interruzione sulla mappa per visualizzare i dettagli che includono l'inizio e la fine di un evento, la città e l'ASN interessati e il tipo di problema (ad esempio, un problema di prestazioni o un problema di disponibilità).

Per ottenere informazioni più dettagliate sugli eventi e per ottenere misurazioni personalizzate del traffico delle applicazioni, crea un monitor in Internet Monitor.

Esempi di casi d'uso di Amazon CloudWatch Internet Monitor

In questa sezione, descriviamo diversi esempi specifici, con collegamenti a post di blog con maggiori dettagli. Questi esempi mostrano come puoi utilizzare le funzionalità di Amazon CloudWatch Internet Monitor per aiutarti a monitorare la tua applicazione e migliorare l'esperienza degli utenti.

Impostazione di avvisi e decisione delle azioni da intraprendere

Puoi utilizzare Monitor Internet per ottenere informazioni sui parametri medi delle prestazioni di Internet nel tempo e sugli eventi di integrità per rete urbana (posizione del cliente e ASN, in genere un provider di servizi Internet). Utilizzando Internet Monitor, puoi identificare gli eventi che influiscono sull'esperienza dell'utente finale per le applicazioni ospitate su Amazon Virtual Private Clouds (VPC), Network Load Balancers WorkSpaces, Amazon o Amazon. CloudFront

Dopo aver creato un monitor, hai diverse opzioni per ricevere avvisi sugli eventi di integrità di Monitor Internet. Queste includono notifiche basate su CloudWatch allarmi che utilizzano metriche degli eventi o EventBridge regole Amazon per filtrare gli eventi sanitari. Puoi scegliere diverse opzioni per le notifiche o le azioni basate sugli allarmi, tra cui, ad esempio, AWS SMS notifiche o aggiornamenti a un CloudWatch gruppo di log.

Per vedere un esempio con linee guida dettagliate, consulta il seguente post di blog: Introduzione ad Amazon CloudWatch Internet Monitor.

Identificazione dei problemi di latenza e migliora il TTFB per migliorare l'esperienza di gioco multigiocatore

Usa Monitor Internet per identificare rapidamente dove i giocatori nelle app di cloud gaming globali riscontrano problemi di latenza a livello globale e per fornire informazioni su come migliorare le prestazioni. Identificando dove la maggior parte dei giocatori ha attualmente il time to first byte (TTFB) più lento, sai come migliorare la latenza per rendere più felice la tua base di giocatori più numerosa.

Ora, quando sei pronto per implementare il prossimo server EC2 per il tuo gioco, scegli Regione AWS quello che, secondo Internet Monitor, ridurrà il TTFB nell'area con l'alta latenza e il grande gruppo di giocatori.

Per informazioni dettagliate sulla configurazione e l'utilizzo di Internet Monitor per questo caso d'uso, consulta il seguente post di blog: Using Amazon CloudWatch Internet Monitor for a Better Gaming Experience.

Osservabilità tra account diversi di Internet Monitor

Con l'osservabilità tra più account di Internet Monitor, puoi monitorare le tue applicazioni che si estendono su più AWS account all'interno di un unico account. Regione AWS

Puoi utilizzare Amazon CloudWatch Observability Access Manager per configurare uno o più AWS account come account di monitoraggio. Fornirai all'account di monitoraggio la possibilità di visualizzare i dati nel tuo account di origine creando un sink nel tuo account di monitoraggio. Un sink è una risorsa che rappresenta un punto di attacco in un account di monitoraggio. Per Internet Monitor, il punto di attacco delle risorse è un monitor. Il sink viene utilizzato per creare un collegamento dal tuo account di origine al tuo account di monitoraggio. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Risorse obbligatorie

Per una corretta funzionalità dell'osservabilità tra più account di CloudWatch Application Insights, assicurati che i seguenti tipi di telemetria siano condivisi tramite Observability Access Manager. CloudWatch

- Monitor in Internet Monitor
- Metriche in Amazon CloudWatch
- Gruppi di log in Amazon CloudWatch Logs

Guida introduttiva ad Amazon CloudWatch Internet Monitor tramite la console

Per iniziare a usare Amazon CloudWatch Internet Monitor, devi creare un monitor in Internet Monitor per la tua applicazione aggiungendo AWS le risorse che utilizza e impostando diverse opzioni di configurazione. Questo capitolo fornisce la procedura per aggiungere un monitor nella console. Include anche una sezione con maggiori dettagli sulle risorse di Monitor Internet e altre sezioni con

descrizioni e limitazioni relative alle diverse opzioni che è possibile o necessario configurare per il monitor.

Indice

Creazione di un monitor in Amazon CloudWatch Internet Monitor utilizzando la console

Puoi creare un monitor in Amazon CloudWatch Internet Monitor per la tua applicazione aggiungendo AWS le risorse che utilizza e quindi impostando diverse opzioni di configurazione. Le risorse che aggiungi, Amazon Virtual Private Clouds (VPC), Network Load Balancer (NLB), CloudFront distribuzioni o WorkSpaces directory, forniscono le informazioni per Internet Monitor per mappare le informazioni sul traffico Internet per la tua applicazione. Dopo aver creato il monitor, attendi 15-30 minuti per generare il profilo di traffico specifico per il luogo in cui viene utilizzata l'applicazione. Quindi, puoi utilizzare il monitor Internet Monitor o altri strumenti per visualizzare ed esplorare le prestazioni e la disponibilità relative all'utilizzo del client. Questi strumenti forniscono informazioni dettagliate utilizzando le misurazioni del traffico delle applicazioni, raccolte e pubblicate dal monitor, ad esempio su CloudWatch Logs.

In genere, è più semplice creare un monitor in Monitor Internet per un'applicazione. All'interno dello stesso monitor, è possibile cercare e ordinare le misurazioni e i parametri nei file di log di Monitor Internet in base a diverse posizioni e ASN (in genere provider di servizi Internet) o altre informazioni. Ad esempio, non è necessario creare monitor separati per applicazioni in aree diverse.

La procedura descritta in questa sezione illustra la configurazione del monitor tramite la console. Per vedere esempi di utilizzo delle azioni AWS Command Line Interface con l'API Internet Monitor, per creare un monitor, visualizzare eventi e così via, consultaEsempi di utilizzo della CLI con Amazon Internet Monitor CloudWatch .

Creazione di un monitor utilizzando la console

1. Aprire la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, in Monitoraggio della rete, scegli Internet Monitor.
3. Scegli Crea monitor.
4. In Monitor name (Nome del monitoraggio), inserisci il nome che desideri utilizzare per questo monitoraggio in Internet Monitor (Monitoraggio Internet).
5. Scegli Add resources (Aggiungi risorse), quindi seleziona le risorse per impostare i limiti di monitoraggio da utilizzare con Monitor Internet per questo monitoraggio.

   > **Note**
   >
   > Ricorda quanto segue:
   >
   > - Per generare un output significativo con Internet Monitor, i VPC aggiunti devono essere connessi a Internet mediante la configurazione di un Internet Gateway.
   > - È possibile aggiungere una combinazione di VPC e CloudFront distribuzioni, aggiungere WorkSpaces directory o aggiungere Network Load Balancer. Non è possibile aggiungere Network Load Balancer o WorkSpaces directory insieme ad altri tipi di risorse.

6. Scegli una percentuale del traffico Internet da monitorare.
7. Facoltativamente, specifica opzioni aggiuntive in Impostazioni avanzate.

   - In Limite massimo per reti urbane, puoi selezionare un limite per il numero di reti urbane (posizioni e ASN o provider di servizi Internet) per cui Monitor Internet monitorerà il traffico. È possibile modificare questa impostazione in qualsiasi momento modificando il monitor. Per informazioni, consulta Scelta del limite massimo per le reti urbane.

     Per ripristinare le impostazioni predefinite, inserisci `500000`.

     Se imposti un limite massimo per le reti urbane, imposta un limite per il numero di reti urbane monitorate da Monitor Internet per la tua applicazione, indipendentemente dalla percentuale di traffico che scegli di monitorare.
   - Facoltativamente, puoi specificare un nome per il bucket Amazon S3 e un prefisso personalizzato per pubblicare misurazioni Internet su Amazon S3 per tutte le reti urbane monitorate.

Internet Monitor pubblica su CloudWatch Logs ogni cinque minuti le prime 500 misurazioni Internet (in base al volume di traffico) relative alla tua applicazione. Se scegli di pubblicare le misurazioni su S3, le misurazioni vengono comunque pubblicate su Logs. CloudWatch Per ulteriori informazioni, consulta Pubblicazione di misurazioni Internet su Amazon S3 in Amazon CloudWatch Internet Monitor.

- Facoltativamente, puoi aggiungere un tag al monitoraggio.

8. Scegli Crea monitor.

Dopo aver creato un monitor, potrai modificarlo in qualsiasi momento, ad esempio per modificare la percentuale di traffico dell'applicazione, aggiornare il limite massimo delle reti urbane o aggiungere o rimuovere risorse. Puoi anche eliminare il monitor. Per eseguire queste attività in Monitor Internet, seleziona un monitor, quindi scegli un'opzione nel menu Operazioni. Non puoi modificare il nome di un monitor.

Per visualizzare il pannello di controllo di Monitor Internet

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Monitoraggio della rete, quindi Internet Monitor.

   La scheda Monitors (Monitoraggi) mostra un elenco dei monitoraggi che hai creato.

Per visualizzare ulteriori informazioni su un monitoraggio specifico, selezionalo.

Aggiunta di risorse al monitor

Quando crei un monitor, devi associare le risorse dell'applicazione ad esso: Amazon Virtual Private Clouds (VPC), Network Load Balancers, CloudFront distribuzioni Amazon, Network Load Balancers (NLB) o directory Amazon. WorkSpaces Quindi, Internet Monitor sa dove si trovano il traffico e i client connessi a Internet dell'applicazione e può creare e mantenere un profilo di traffico che determina le misurazioni pertinenti da pubblicare per il monitor.

È possibile aggiungere i seguenti tipi di risorse a un monitor in Internet Monitor come «risorse monitorate». Tieni presente che Internet Monitor non supporta l'aggiunta di diversi tipi di risorse in un unico monitor.

- VPC: ogni VPC aggiunto in una Regione è una risorsa monitorata. Quando aggiungi un VPC, Internet Monitor monitora il traffico di qualsiasi applicazione connessa a Internet nel VPC, ad

esempio un'applicazione ospitata su un'istanza Amazon EC2, dietro un Network Load Balancer o in un container. AWS Fargate

- Network Load Balancer: ogni NLB aggiunto è una risorsa monitorata.
- CloudFront distribuzioni: ogni CloudFront distribuzione che aggiungi è una risorsa monitorata.
- WorkSpaces directory: ogni WorkSpaces directory aggiunta in una regione è una risorsa monitorata.

Quando si monitora il traffico per i VPC, viene monitorato il traffico per le applicazioni ospitate sui sistemi di bilanciamento del carico dietro il VPC. Puoi scegliere di monitorare il traffico per singoli sistemi di bilanciamento del carico Network Load Balancer invece di monitorare un VPC con più sistemi di bilanciamento del carico. Ciò può essere utile, ad esempio, se è necessario comprendere e configurare le funzionalità per migliorare le prestazioni o l'efficienza a livello di sistema di bilanciamento del carico. In alternativa, potresti aver bisogno di informazioni sulla conformità a livello di Network Load Balancer.

Quando aggiungi risorse a un monitor in Monitor Internet, tieni presente quanto riportato di seguito:

- Per generare un output significativo con Internet Monitor, i VPC aggiunti devono essere connessi a Internet mediante la configurazione di un Internet Gateway.
- Internet Monitor non supporta l'aggiunta di diversi tipi di risorse in un unico monitor.

Quando aggiungi VPC o NLB come risorse, ci sono delle differenze regionali tra le regioni che scelgono di aderire. Per ulteriori informazioni, consulta Supportato Regioni AWS per Amazon CloudWatch Internet Monitor.

Tieni presente che esistono differenze tra le risorse sulla misurazione della latenza dell'ultimo miglio. Per le misurazioni della latenza di Internet Monitor, VPC, NLB e directory non includono la latenza dell'ultimo miglio. WorkSpaces

Scelta della percentuale di traffico delle applicazioni da monitorare

La copertura scelta per la percentuale di traffico delle applicazioni da monitorare determina quante reti urbane (posizioni dei client e ASN, in genere provider di servizi Internet) per l'applicazione vengono monitorate, fino a un limite massimo opzionale per le reti urbane che è possibile impostare.

Se scegli di monitorare meno del 100% del traffico delle applicazioni, potrebbe esistere una lacuna di osservabilità con il monitor. Questo perché se Amazon CloudWatch Internet Monitor crea eventi

sanitari che non monitorano il traffico, non sarai a conoscenza di tali problemi. Potresti anche avere una copertura inferiore per le informazioni sui punteggi di prestazioni e disponibilità relativi all'accesso dei client alla tua applicazione.

Le sezioni seguenti descrivono le opzioni per esplorare le impostazioni e la copertura della percentuale di traffico e per farsi un'idea dell'impatto dell'aumento o della diminuzione della copertura.

- Esplorazione della modifica della percentuale di traffico delle applicazioni
- Visualizzazione del numero di reti urbane monitorate con diverse impostazioni di percentuale di traffico

Modifica della percentuale di traffico delle applicazioni

Puoi esplorare i valori in base ai quali potresti voler modificare la percentuale di traffico delle tue applicazioni visualizzando il numero di reti urbane monitorate quando modifichi la percentuale. La procedura in questa sezione fornisce step-by-step informazioni.

Nella console Monitor Internet, puoi provare ad aumentare o diminuire la percentuale di traffico delle applicazioni per il tuo monitor e visualizzare il numero stimato di reti urbane che verrebbero coperte di conseguenza. Con questa opzione, puoi visualizzare rapidamente in che modo la modifica della percentuale di traffico influisce sul numero di monitor urbani monitorati. In questo modo puoi farti un'idea di quale potrebbe essere una buona percentuale di traffico dell'applicazione da scegliere per la tua applicazione.

Esplorazione della copertura del monitoraggio aumentando e diminuendo la percentuale di traffico delle applicazioni

1. Aprire la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, in Monitoraggio della rete, scegli Internet Monitor.
3. Nell'elenco dei monitor, scegli un monitor.
4. Nella scheda Panoramica, nella sezione Traffico monitorato, scegli il grafico percentuale, quindi scegli Aggiorna copertura di monitoraggio.
5. Nella finestra di dialogo Esplora e imposta la copertura di monitoraggio del traffico, fai clic sulle frecce per aumentare o diminuire la percentuale di traffico da monitorare. Scegliendo il 100% del traffico, puoi vedere quante reti urbane sono monitorate con una copertura completa per il monitoraggio della tua applicazione.

6. Per saperne di più su come il numero di reti urbane monitorate (stimato qui) potrebbe influire sui costi, scegli il link al Calcolatore CloudWatch dei prezzi, quindi scorri verso il basso fino a Internet Monitor.
7. Per impostare una nuova percentuale di traffico da monitorare, scegli Aggiorna la copertura del monitor. Oppure, per mantenere il livello di copertura attuale, scegli Annulla.

Visualizza il numero di reti urbane monitorate con diverse impostazioni di percentuale di traffico

Puoi visualizzare il numero di reti urbane che verrebbero monitorate per la tua applicazione con diverse percentuali di traffico dell'applicazione. La procedura riportata in questa sezione fornisce informazioni. step-by-step

Nella console di Monitor Internet, è possibile visualizzare grafici che mostrano come cambierebbe la copertura delle reti urbane a seconda delle percentuali di traffico delle applicazioni, in un intervallo di tempo specificato. Si tratta di un modo rapido per visualizzare e confrontare la copertura di monitoraggio per l'applicazione in base a percentuali di traffico specifiche, il tutto su un unico grafico.

Visualizzazione dei grafici della percentuale di traffico delle applicazioni e della corrispondente copertura delle reti urbane

1. Aprire la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, in Monitoraggio della rete, scegli Internet Monitor.
3. Nell'elenco dei monitor, scegli un monitor.
4. Scegli la scheda Informazioni sul traffico e scorri verso il basso fino ai grafici del traffico Internet.
5. In Confronta le opzioni per la copertura del traffico, nell'elenco a discesa, seleziona una o più percentuali. Puoi scegliere una o più percentuali di traffico delle applicazioni e il grafico delle reti urbane totali monitorate viene aggiornato per mostrare la copertura di monitoraggio fornita da Monitor Internet per quella percentuale di traffico. Scegliendo Reti urbane con il 100% del traffico, puoi vedere quante reti urbane sono monitorate con una copertura completa per il monitoraggio della tua applicazione.

Ricorda:

- La copertura del traffico viene calcolata in base al numero di reti urbane nell'ora precedente del traffico dell'applicazione. Ciò significa che, dopo aver scelto una percentuale specifica di traffico da monitorare, le reti urbane monitorate per l'applicazione potrebbero essere meno numerose rispetto a quelle mostrate nel grafico di confronto della copertura del traffico.

- Per assicurarti che tutto il traffico delle tue applicazioni sia monitorato, imposta `TrafficPercentageToMonitor` su 100 e non impostare `MaxCityNetworksToMonitor`. In alternativa, puoi impostare `MaxCityNetworksToMonitor` su 500.000, il limite massimo di Monitor Internet.
- Se hai impostato un limite massimo per le reti urbane, il numero totale di reti urbane monitorate non supererà mai tale limite, indipendentemente dall'opzione di percentuale di traffico dell'applicazione selezionata.
- Scopri di più su come il numero di reti urbane monitorate potrebbe influire sui costi. Nella CloudWatch pagina Calcolatore dei prezzi, scorri verso il basso fino a Internet Monitor.

Per impostare una nuova percentuale di traffico da monitorare, in Esplora altre opzioni di copertura del traffico, scegli Aggiorna copertura di monitoraggio. Nella finestra di dialogo, scegli una percentuale di traffico, quindi scegli Aggiorna copertura del monitor.

Scelta del limite massimo per le reti urbane

Amazon CloudWatch Internet Monitor può monitorare il traffico delle tue applicazioni per alcune o tutte le località in cui i client accedono alle risorse delle tue applicazioni e tutti gli ASN (in genere provider di servizi Internet) attraverso cui accedono all'applicazione, ovvero le reti urbane per il traffico Internet delle tue applicazioni. Quando crei il monitor, scegli una percentuale del traffico dell'applicazione da monitorare, che puoi aggiornare in qualsiasi momento modificando il monitor.

Oltre a impostare una percentuale di traffico, puoi anche impostare un limite massimo per il numero di reti urbane monitorate. In questa sezione viene descritto come il limite per le reti urbane può aiutarti a gestire i costi di fatturazione e fornisce informazioni e un esempio per aiutarti a determinare un limite da impostare.

Il limite massimo che imposti per il numero di reti urbane aiuta a garantire che la fattura sia prevedibile. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon. Puoi anche scoprire in che modo valori diversi per il numero di reti urbane effettivamente monitorate possono influire sulla bolletta utilizzando il calcolatore dei CloudWatch prezzi. Per esplorare le opzioni, nella CloudWatch pagina Calcolatore dei prezzi, scorri verso il basso fino a Internet Monitor.

Per aggiornare il monitor e modificare il limite massimo per le reti urbane, consulta Modifica o eliminazione di un monitor di Monitor Internet.

Come funziona la fatturazione con i limiti massimi delle reti urbane

L'impostazione di un limite massimo per il numero di reti urbane monitorate può aiutare a prevenire costi imprevisti in fattura. Ciò è utile, ad esempio, se i modelli di traffico variano notevolmente. I costi di fatturazione aumentano per ogni rete urbana monitorata dopo le prime 100 reti urbane che sono incluse (su tutti i monitor per account). Se imposti un limite massimo per le reti urbane, imposta un limite per il numero di reti urbane monitorate da Monitor Internet per la tua applicazione, indipendentemente dalla percentuale di traffico che decidi di monitorare.

Paghi solo per il numero di reti urbane effettivamente monitorate. Il limite massimo per rete urbana che scegli ti consente di impostare un limite sul totale che può essere incluso quando Monitor Internet monitora il traffico con il tuo monitor. È possibile modificare il limite massimo in qualsiasi momento modificando il monitor.

Per esplorare le opzioni, nella CloudWatch pagina Calcolatore dei prezzi scorri verso il basso fino a Internet Monitor. Per ulteriori informazioni sui prezzi di Internet Monitor, consulta la sezione Internet Monitor nella pagina CloudWatch dei prezzi di Amazon.

Come scegliere il limite massimo per le reti urbane

Per aiutarti a decidere il limite massimo per le reti urbane da selezionare, considera la quantità di traffico che desideri monitorare per la tua applicazione. I seguenti parametri di Monitor Internet possono aiutarti ad analizzare l'utilizzo e la copertura del traffico dopo aver creato il monitor: `CityNetworksMonitored`, `TrafficMonitoredPercent` e uno o più dei parametri `CityNetworksFor`*`NN`*`PercentTraffic`, dove *`NN`* è un valore percentuale che corrisponde a uno dei seguenti: 25, 50, 90, 95, 99 o 100. Per esaminare le definizioni di questi parametri e di tutti gli altri parametri di Monitor Internet, consulta Utilizzo di CloudWatch metriche con Amazon CloudWatch Internet Monitor.

Per visualizzare un grafico generale della copertura del traffico Internet, vai alla scheda Traffic Insights sulla CloudWatch dashboard e, nella sezione Grafici del traffico Internet, scegli un'opzione per Confronta le opzioni per la copertura del traffico. Il grafico mostrato nella sezione mostra il numero effettivo di reti urbane monitorate per l'applicazione, oltre alle linee del grafico per le diverse percentuali di traffico delle applicazioni selezionate nell'elenco a discesa. Per ulteriori informazioni, consulta Impostazione della percentuale di traffico delle applicazioni.

Per esplorare le opzioni in modo più dettagliato, puoi utilizzare i parametri di Monitor Internet, come descritto negli esempi seguenti. Questi esempi mostrano come selezionare il limite massimo per le reti urbane più adatto a te, a seconda dell'ampiezza della copertura del traffico Internet

dell'applicazione che desideri. L'utilizzo delle query per le metriche di Internet Monitor in CloudWatch Metrics può aiutarti a comprendere meglio la copertura del traffico Internet della tua applicazione.

Esempio di determinazione del limite massimo delle reti urbane

Ad esempio, supponiamo di aver impostato un limite massimo di monitoraggio di 100 reti urbane e che i client accedano all'applicazione su 2.637 reti urbane. In CloudWatch Metrics, vedrai restituite le seguenti metriche di Internet Monitor:

```
CityNetworksMonitored 100
TrafficMonitoredPercent  12.5
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

Da questo esempio, puoi vedere che attualmente stai monitorando il 12,5% del tuo traffico Internet, con il limite massimo impostato su 100 reti urbane. Se desideri monitorare il 90% del tuo traffico, il parametro successivo fornisce informazioni al riguardo: `CityNetworksFor90PercentTraffic` indica che dovresti monitorare 2.143 reti urbane per una copertura del 90%. A tal fine, è necessario aggiornare il monitor e impostare il limite massimo delle reti urbane su 2.143.

Allo stesso modo, supponiamo che desideri avere un monitoraggio del traffico Internet al 100% per la tua applicazione. Il parametro successivo, `CityNetworksFor100PercentTraffic`, indica che a tale scopo, è necessario aggiornare il monitor per impostare il limite massimo per le reti urbane a 2.637.

Se ora imposti il massimo a 5.000 reti urbane, poiché è maggiore di 2.637, verranno restituiti i seguenti parametri:

```
CityNetworksMonitored 2637
TrafficMonitoredPercent  100
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

Da questi parametri, puoi vedere che con il limite più alto, monitori tutte le 2.637 reti urbane, ovvero il 100% del tuo traffico Internet.

Pubblicazione di misurazioni Internet su Amazon S3 in Amazon CloudWatch Internet Monitor

Puoi scegliere di fare in modo che Amazon CloudWatch Internet Monitor pubblichi le misurazioni Internet su Amazon S3 del traffico connesso a Internet verso le reti urbane monitorate (sedi dei clienti e ASN, in genere provider di servizi Internet) sul monitor, fino al limite di servizio di 500.000 reti urbane. Internet Monitor pubblica automaticamente le misurazioni di Internet su CloudWatch Logs ogni cinque minuti per le prime 500 reti urbane (per volume di traffico) per ogni monitor. Le misurazioni che pubblica su S3 includono le prime 500 pubblicate su Logs. CloudWatch

Puoi scegliere l'opzione di pubblicazione su S3 e specificare il bucket in cui pubblicare le misurazioni quando crei o aggiorni il monitor. Per poterlo specificare in Monitor Internet, il bucket deve essere già stato creato in S3. Esiste un limite di servizio di 500.000 reti urbane per le misurazioni Internet pubblicate su S3. Monitor Internet pubblica le misurazioni di Internet su S3 come eventi, una serie di oggetti di log compressi che vengono memorizzati nel bucket.

Quando crei il bucket S3 in cui Internet Monitor può pubblicare le misurazioni, assicurati di seguire le indicazioni sulle autorizzazioni fornite da Logs. CloudWatch In questo modo, Internet Monitor può pubblicare i log direttamente su S3 e, se necessario, AWS può creare e modificare le politiche delle risorse associate al gruppo di log che riceve i log. Per ulteriori informazioni, consulta Logs sent to CloudWatch Logs nella Amazon CloudWatch Logs User Guide.

I file di log pubblicati sono compressi. Se apri i file di log utilizzando la console Amazon S3, questi vengono decompressi e vengono visualizzati gli eventi di misurazione di Internet. Se i file vengono scaricati, per visualizzare gli eventi devono prima essere decompressi.

Puoi anche eseguire query sulle misurazioni Internet nei file di log utilizzando Amazon Athena. Amazon Athena è un servizio di query interattivo che semplifica l'analisi dei dati in Amazon S3 con SQL standard. Per ulteriori informazioni, consulta Utilizzo di Amazon Athena per eseguire query delle misurazioni Internet nei file di log di Amazon S3.

Utilizzo di un monitor di Monitor Internet

Esistono diversi modi per utilizzare un monitor Amazon CloudWatch Internet Monitor dopo averlo creato: ad esempio, puoi visualizzare le informazioni nella CloudWatch dashboard, ottenere informazioni utilizzando e impostare avvisi sullo stato. AWS Command Line Interface

Il monitor fornisce informazioni sull'applicazione e sulle preferenze di configurazione, cosicché Monitor Internet possa personalizzare misurazioni e parametri da pubblicare negli eventi per tuo

conto. Monitor Internet raccoglie le misurazioni dell'impronta dell'infrastruttura globale per AWS. Queste misurazioni sono un'enorme quantità di informazioni sulle prestazioni e sulla disponibilità della rete, provenienti da tutto il mondo. Utilizzando le informazioni provenienti dalle risorse aggiunte all'applicazione, Monitor Internet pubblica automaticamente misurazioni delle prestazioni e della disponibilità relative alle reti urbane (ovvero posizioni dei client e ASN, in genere provider di servizi Internet o ISP) in cui l'applicazione è attiva. Pertanto, le misurazioni e le metriche nella dashboard di Internet Monitor e nei CloudWatch registri, relative alla disponibilità, alle prestazioni, ai byte monitorati trasferiti e al tempo di andata e ritorno, sono specifiche per le ubicazioni dei clienti e gli ASN.

Monitor Internet determina anche quando ci sono anomalie nelle prestazioni e nella disponibilità. Per impostazione predefinita, Internet Monitor sovrappone il traffico alle misurazioni della disponibilità e delle prestazioni raccolte per ogni coppia sorgente-destinazione nelle AWS sedi dei clienti, per determinare quando si verificano cali notevoli delle prestazioni o della disponibilità. Quando si verifica un deterioramento significativo delle posizioni e dell'ambito dell'applicazione, Monitor Internet genera un evento di integrità e pubblica le informazioni sul problema sul monitor.

Dopo aver configurato un monitor, puoi utilizzarlo per accedere o ricevere avvisi sulle informazioni fornite da Monitor Internet, nei seguenti modi:

- Utilizza la CloudWatch dashboard per visualizzare ed esplorare gli eventi relativi a prestazioni, disponibilità e integrità; esplora i dati storici dell'applicazione e ottieni informazioni su nuovi modi per configurare l'applicazione per migliorare le prestazioni. Per ulteriori informazioni, consulta quanto segue:
  - Monitoraggio delle prestazioni e della disponibilità in tempo reale in Amazon CloudWatch Internet Monitor (scheda Panoramica)
  - Filtraggio e visualizzazione dei dati storici in Amazon CloudWatch Internet Monitor (scheda Historical explorer)
  - Ottenere informazioni per migliorare le prestazioni delle applicazioni in Amazon CloudWatch Internet Monitor (scheda Traffic insights)
- Configura le soglie degli eventi di integrità per modificare ciò che spinge Monitor Internet a creare un evento di integrità per la tua applicazione. È possibile configurare soglie generali e soglie locali (rete urbana). Per ulteriori informazioni, consulta Modifica delle soglie degli eventi di integrità.
- Utilizza AWS CLI i comandi con le azioni dell'API Internet Monitor per visualizzare le informazioni sul profilo di traffico, visualizzare le misurazioni, elencare gli eventi sanitari e così via. Per ulteriori informazioni, consulta Esempi di utilizzo della CLI con Amazon Internet Monitor CloudWatch .
- Utilizza CloudWatch strumenti standard, come CloudWatch Contributor Insights, CloudWatch Metrics explorer e CloudWatch Logs Insights per visualizzare i dati in esso contenuti. CloudWatch

Per ulteriori informazioni, consulta Esplorazione dei dati con CloudWatch gli strumenti e l'interfaccia di interrogazione di Internet Monitor.

- Usa Athena con i log di S3 per accedere e analizzare le misurazioni Internet di Monitor Internet per la tua applicazione se hai attivato la pubblicazione delle misurazioni su S3.
- Crea EventBridge notifiche Amazon per avvisarti quando Internet Monitor rileva la presenza di un evento sanitario. Per ulteriori informazioni, consulta Utilizzo di Amazon CloudWatch Internet Monitor con Amazon EventBridge.
- Ricevi automaticamente una AWS Health Dashboard notifica quando Internet Monitor determina che un problema è causato dalla AWS rete. La notifica include le misure adottate AWS per mitigare il problema.

Modifica o eliminazione di un monitor di Monitor Internet

Utilizzando il menu Azione, puoi modificare o eliminare un monitor in Amazon CloudWatch Internet Monitor dopo averlo creato. Ad esempio, è possibile modificare un monitor per completare le seguenti operazioni:

- Modifica della percentuale di traffico delle applicazioni da monitorare
- Impostazione o aggiornamento del limite massimo delle reti urbane
- Modifica delle soglie degli eventi di integrità per quanto riguarda la disponibilità o i punteggi delle prestazioni
- Aggiunta o rimozione di risorse
- Abilitazione o aggiornamento della pubblicazione di eventi in Amazon S3

Puoi anche eliminare un monitor. Non puoi modificare il nome di un monitor dopo averlo creato.

Per apportare modifiche a un monitor o eliminare un monitor, utilizza una delle seguenti procedure.

Modifica di un monitor

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, in Monitoraggio della rete, scegli Internet Monitor.
3. Scegli il tuo monitor, quindi scegli il menu Operazione.
4. Scegli Aggiorna monitor.

5. Effettua gli aggiornamenti desiderati. Ad esempio, per modificare la percentuale di traffico da monitorare, in Traffico delle applicazioni da monitorare, seleziona o inserisci una percentuale.
6. Scegli Aggiorna.

Per eliminare un monitor

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, in Monitoraggio della rete, scegli Internet Monitor.
3. Scegli il tuo monitor, quindi scegli il menu Operazione.
4. Scegliere Disabilita.
5. Scegli di nuovo il menu Azione, quindi scegli Elimina.

Per ulteriori informazioni sulle opzioni che è possibile aggiornare, consulta quanto segue:

- Per ulteriori informazioni sulle risorse che aggiungi in Monitor Internet, consulta Aggiunta di risorse al monitor.
- Per ulteriori informazioni sulla percentuale di traffico delle applicazioni, consulta Scelta della percentuale di traffico delle applicazioni da monitorare.
- Per ulteriori informazioni sulla modifica delle soglie degli eventi di integrità, consulta Modifica delle soglie degli eventi di integrità.
- Per ulteriori informazioni sul limite massimo delle reti urbane, consulta Scelta del limite massimo per le reti urbane.
- Per ulteriori informazioni sulla scelta di pubblicare eventi su S3, consulta Pubblicazione di misurazioni Internet su Amazon S3 in Amazon CloudWatch Internet Monitor.

Aggiungi o crea un monitor Amazon CloudWatch Internet Monitor con Amazon VPC

Quando crei un Amazon Virtual Private Cloud VPC in AWS Management Console, puoi facoltativamente scegliere di configurarne il monitoraggio anche in Amazon CloudWatch Internet Monitor. È possibile aggiungere il VPC a un monitor esistente oppure creare un nuovo monitor per il VPC nella console Amazon VPC.

Utilizzando Monitor Internet con il VPC, è possibile visualizzare e valutare misurazioni e parametri relativi alla disponibilità, alle prestazioni, ai byte monitorati trasferiti e ai tempi di andata e ritorno specifici per le posizioni dei client e gli ASN dell'applicazione (in genere, provider di servizi Internet).

Monitor Internet determina anche la presenza di anomalie nelle prestazioni e nella disponibilità e crea nel monitor degli eventi di integrità, per i quali è possibile scegliere di ricevere notifiche. Per ulteriori informazioni su come utilizzare un monitor per gestire e migliorare l'esperienza dei clienti con l'applicazione, consulta la pagina Utilizzo di un monitor di Monitor Internet.

> **Important**
>
> Per creare un monitor o aggiungere un VPC a un monitor esistente, è necessario disporre delle autorizzazioni corrette. Per ulteriori informazioni, consulta Identity and Access Management per Amazon CloudWatch Internet Monitor.

Aggiunta di un VPC a un monitor esistente

Puoi scegliere di fare in modo che Amazon CloudWatch Internet Monitor aggiunga un nuovo VPC a un monitor esistente per te quando crei il VPC in. AWS Management Console Dopo aver aggiunto il VPC, attendi qualche minuto, quindi le metriche per il VPC inizieranno a essere visualizzate sulla console di Internet Monitor.

Puoi modificare il monitor in qualsiasi momento, rimuovere il VPC o aggiungere un altro VPC o altre risorse. Puoi anche modificare la percentuale di traffico che stai monitorando o apportare altre modifiche. Se scegli di rimuovere il VPC dal monitor, il traffico proveniente dai client verso quel VPC non verrà più monitorato da Monitor Internet.

Per ulteriori informazioni sull'aggiornamento di un monitor, consulta la pagina Modifica o eliminazione di un monitor di Monitor Internet.

Creazione di un monitor per un VPC

Se scegli di creare un monitor per un VPC, la procedura guidata Crea monitor ti guida attraverso i vari passaggi. Il VPC viene aggiunto come risorsa monitorata quando si crea il monitor. Se lo desideri, puoi anche scegliere una percentuale di traffico client che desideri monitorare per la tua applicazione (l'impostazione predefinita è 100%).

Per saperne di più, consulta le informazioni riportate nella sezione Creazione di un monitor in Amazon CloudWatch Internet Monitor utilizzando la console.

Prezzi

Con Amazon CloudWatch Internet Monitor, paghi solo per ciò che usi. Il prezzo di Monitor Internet prevede due componenti: una tariffa per risorsa monitorata e una tariffa per rete urbana. Una rete

urbana è il luogo da cui i client accedono alle risorse dell'applicazione e la rete (un ASN, come un provider di servizi Internet o un ISP) attraverso cui i client accedono alle risorse.

Per ulteriori informazioni, inclusi esempi di prezzi, consulta Prezzi di Amazon CloudWatch Internet Monitor

Interruzione del monitoraggio di un VPC

Se desideri interrompere il monitoraggio della tua risorsa VPC con Internet Monitor, procedi come segue nella console Internet Monitor:

Rimozione di una risorsa da un monitor

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, in Monitoraggio della rete, scegli Internet Monitor.
3. Scegli il tuo monitor, quindi scegli il menu Operazione.
4. Scegli Aggiorna monitor.
5. In Risorse aggiunte, scegli Rimuovi risorse.
6. Scegli il VPC da rimuovere, quindi scegli Rimuovi.
7. Scegli Aggiorna.

Aggiungi o crea un monitor Amazon CloudWatch Internet Monitor con CloudFront

Nella dashboard delle metriche per una distribuzione nella CloudFront console Amazon, puoi configurare un monitoraggio aggiuntivo per una distribuzione in Amazon CloudWatch Internet Monitor. Puoi aggiungere la distribuzione a un monitor esistente oppure puoi creare un nuovo monitor per la distribuzione.

Utilizzando Internet Monitor con la CloudFront distribuzione, è possibile visualizzare e valutare misurazioni e metriche relative alla disponibilità, alle prestazioni, ai byte monitorati trasferiti e ai tempi di andata e ritorno specifici per le ubicazioni dei client e gli ASN dell'applicazione (in genere i provider di servizi Internet). Monitor Internet determina anche la presenza di anomalie nelle prestazioni e nella disponibilità e crea nel monitor degli eventi di integrità, per i quali è possibile scegliere di ricevere notifiche. Per ulteriori informazioni su come utilizzare un monitor per gestire e migliorare l'esperienza dei clienti con l'applicazione, consulta la pagina Utilizzo di un monitor di Monitor Internet.

> **Important**
>
> Per creare un monitor o aggiungere una distribuzione a un monitor esistente, è necessario disporre delle autorizzazioni corrette. Per ulteriori informazioni, consulta Identity and Access Management per Amazon CloudWatch Internet Monitor.

Aggiungere una distribuzione a un monitor esistente

Puoi scegliere di fare in modo che Internet Monitor aggiunga una distribuzione a un monitor esistente direttamente dalla dashboard CloudFront delle metriche nel AWS Management Console. Dopo aver aggiunto la distribuzione, attendi qualche minuto, quindi le metriche relative alla distribuzione inizieranno a essere visualizzate sulla console di Internet Monitor.

È possibile modificare il monitor in qualsiasi momento, per rimuovere la distribuzione o aggiungere un'altra distribuzione o altre risorse. Puoi anche modificare la percentuale di traffico che stai monitorando o apportare altre modifiche. Se scegli di rimuovere la distribuzione dal monitor, il traffico proveniente dai client verso quella distribuzione non viene più monitorato da Internet Monitor.

Per ulteriori informazioni sull'aggiornamento di un monitor, consulta la pagina Modifica o eliminazione di un monitor di Monitor Internet.

Crea un monitor per una distribuzione

Se scegli di creare un monitor per una distribuzione, la procedura guidata Crea monitor ti guida attraverso i passaggi. La distribuzione viene aggiunta come risorsa monitorata quando si crea il monitor. Se lo desideri, puoi anche scegliere una percentuale di traffico client che desideri monitorare per la tua applicazione (l'impostazione predefinita è 100%).

Per saperne di più, consulta le informazioni riportate nella sezione Creazione di un monitor in Amazon CloudWatch Internet Monitor utilizzando la console.

Prezzi

Con Amazon CloudWatch Internet Monitor, paghi solo per ciò che usi. Il prezzo di Monitor Internet prevede due componenti: una tariffa per risorsa monitorata e una tariffa per rete urbana. Una rete urbana è il luogo da cui i client accedono alle risorse dell'applicazione e la rete (un ASN, come un provider di servizi Internet o un ISP) attraverso cui i client accedono alle risorse.

Per ulteriori informazioni, inclusi esempi di prezzi, consulta Prezzi di Amazon CloudWatch Internet Monitor

Interrompi il monitoraggio di una distribuzione

Se desideri interrompere il monitoraggio delle risorse di distribuzione con Internet Monitor, procedi come segue nella console Internet Monitor:

Rimozione di una risorsa da un monitor

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, in Monitoraggio della rete, scegli Internet Monitor.
3. Scegli il tuo monitor, quindi scegli il menu Operazione.
4. Scegli Aggiorna monitor.
5. In Risorse aggiunte, scegli Rimuovi risorse.
6. Scegli la distribuzione da rimuovere, quindi scegli Rimuovi.
7. Scegli Aggiorna.

Esempi di utilizzo della CLI con Amazon Internet Monitor CloudWatch

Questa sezione include esempi per l'utilizzo delle operazioni AWS Command Line Interface con Amazon CloudWatch Internet Monitor.

Prima di iniziare, assicurati di accedere per utilizzarlo AWS CLI con lo stesso AWS account su cui sono presenti Amazon Virtual Private Clouds (VPC), Network Load Balancers CloudFront , distribuzioni Amazon o WorkSpaces directory Amazon che desideri monitorare. Monitor Internet non supporta l'accesso alle risorse tra account. Per ulteriori informazioni sull'utilizzo di, consulta il Command AWS CLI Reference.AWS CLI Per ulteriori informazioni sull'utilizzo delle azioni API con Amazon CloudWatch Internet Monitor, consulta la Amazon CloudWatch Internet Monitor API Reference Guide.

Argomenti

- Creazione di un monitoraggio
- Visualizzare i dettagli del monitoraggio
- Elenco degli eventi di stato
- Visualizzazione di un evento di stato specifico
- Visualizzazione dell'elenco dei monitoraggi
- Modifica del monitoraggio
- Eliminazione del monitoraggio

Creazione di un monitoraggio

Durante la creazione di un monitoraggio in Monitor Internet, indica un nome e associa le risorse al monitoraggio per mostrare dove si trova il traffico Internet dell'applicazione. Specifica una percentuale di traffico che definisce la quantità di traffico dell'applicazione monitorata. Ciò determina anche il numero di reti urbane, ovvero le posizioni dei client e gli ASN, in genere provider di servizi Internet o ISP, che vengono monitorati. Per tenere sotto controllo la fattura, puoi anche impostare un limite per il numero massimo di reti urbane da monitorare per le risorse dell'applicazione. Per ulteriori informazioni, consulta Scelta del limite massimo per le reti urbane.

Infine, puoi decidere se pubblicare tutte le misurazioni Internet per la tua applicazione su Amazon S3. Le misurazioni Internet per le 500 principali reti urbane (per volume di traffico) vengono pubblicate automaticamente su CloudWatch Logs by Internet Monitor, ma puoi scegliere di pubblicare tutte le misurazioni anche su S3.

Per creare un monitor con AWS CLI, si usa il comando. `create-monitor` Il comando seguente crea un monitor che monitora il 100% del traffico ma imposta un limite massimo di 10.000 reti urbane, aggiunge una risorsa VPC e sceglie di pubblicare misurazioni Internet su Amazon S3.

> **Note**
>
> Internet Monitor pubblica su CloudWatch Logs le misurazioni Internet ogni cinque minuti per le 500 principali reti cittadine (sedi dei clienti e ASN, in genere provider di servizi Internet o ISP) che inviano traffico a ciascun monitor. Facoltativamente, puoi decidere di pubblicare misurazioni ed eventi Internet per tutte le reti urbane monitorate (fino al limite di servizio di 500.000 reti urbane) in un bucket Amazon S3. Per ulteriori informazioni, consulta Pubblicazione di misurazioni Internet su Amazon S3 in Amazon CloudWatch Internet Monitor.

```
aws internetmonitor --create-monitor monitor-name "TestMonitor" \
    --traffic-percentage-to-monitor 100 \
    --max-city-networks-to-monitor 10000 \
    --resources "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --internet-measurements-log-delivery
 S3Config="{BucketName=MyS3Bucket,LogDeliveryStatus=ENABLED}"
```

```
{
"Arn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
 "Status": "ACTIVE"
```

```
}
```

> **Note**
>
> Non puoi modificare il nome di un monitor.

Visualizzare i dettagli del monitoraggio

Per visualizzare le informazioni su un monitor con, si usa il AWS CLI comando. `get-monitor`

```
aws internetmonitor get-monitor --monitor-name "TestMonitor"
```

```
{
    "ClientLocationType": "city",
    "CreatedAt": "2022-09-22T19:27:47Z",
    "ModifiedAt": "2022-09-22T19:28:30Z",
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "MonitorName": "TestMonitor",
    "ProcessingStatus": "OK",
    "ProcessingStatusInfo": "The monitor is actively processing data",
    "Resources": [
        "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889"
    ],
    "MaxCityNetworksToMonitor": 10000,
    "Status": "ACTIVE"
}
```

Elenco degli eventi di stato

Quando le prestazioni del traffico Internet dell'applicazione peggiorano, Monitor Internet crea eventi di stato all'interno del monitoraggio. Per visualizzare un elenco degli eventi sanitari in corso con AWS CLI, utilizzare il `list-health-events` comando

```
aws internetmonitor list-health-events --monitor-name "TestMonitor"
```

```
{
    "HealthEvents": [
        {
            "EventId": "2022-06-20T01-05-05Z/latency",
```

```
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:15:14Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.21,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 60.370000000000005,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                }
            ],
            "StartedAt": "2022-06-20T01:05:05Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-05-05Z/latency"
        },
        {
            "EventId": "2022-06-20T01-17-56Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:30:23Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.29,
            "ClientLocations": [
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 75.32,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
```

```
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 22.91,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
                }
            ],
            "StartedAt": "2022-06-20T01:17:56Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-17-56Z/latency"
        },
        {
            "EventId": "2022-06-20T01-34-20Z/latency",
```

```
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:35:04Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.15,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 39.45,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 29.770000000000003,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
```

```
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
                }
            ],
            "StartedAt": "2022-06-20T01:34:20Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-34-20Z/latency"
        }
    ]
}
```

Visualizzazione di un evento di stato specifico

Per visualizzare informazioni più dettagliate su uno specifico evento di stato con la CLI, esegui il comando `get-health-event` con il nome del monitoraggio e l'ID dell'evento di stato.

```
aws internetmonitor get-monitor --monitor-name "TestMonitor" --event-id "health-event/
TestMonitor/2021-06-03T01:02:03Z/latency"
```

```
{
    "EventId": "2022-06-20T01-34-20Z/latency",
    "Status": "RESOLVED",
    "EndedAt": "2022-06-20T01:35:04Z",
    "ServiceLocations": [
        {
            "Name": "us-east-1"
        }
    ],
    "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor/
health-event/2022-06-20T01-34-20Z/latency",
    "LastUpdatedAt": "2022-06-20T01:35:04Z",
    "ClientLocations": [
        {
            "City": "Lockport",
            "PercentOfClientLocationImpacted": 39.45,
            "PercentOfTotalTraffic": 2.01,
            "Country": "United States",
```

```
            "Longitude": -78.6913,
            "AutonomousSystemNumber": 26101,
            "Latitude": 43.1721,
            "Subdivision": "New York",
            "NetworkName": "YAHOO-BF1"
        },
        {
            "City": "Toronto",
            "PercentOfClientLocationImpacted": 29.770000000000003,
            "PercentOfTotalTraffic": 1.05,
            "Country": "Canada",
            "Longitude": -79.3623,
            "AutonomousSystemNumber": 14061,
            "Latitude": 43.6547,
            "Subdivision": "Ontario",
            "CausedBy": {
                "Status": "ACTIVE",
                "Networks": [
                    {
                        "AutonomousSystemNumber": 16509,
                        "NetworkName": "Amazon.com"
                    }
                ],
                "NetworkEventType": "AWS"
            },
            "NetworkName": "DIGITALOCEAN-ASN"
        },
        {
            "City": "Shenzhen",
            "PercentOfClientLocationImpacted": 4.07,
            "PercentOfTotalTraffic": 0.61,
            "Country": "China",
            "Longitude": 114.0683,
            "AutonomousSystemNumber": 37963,
            "Latitude": 22.5455,
            "Subdivision": "Guangdong",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        },
        {
            "City": "Hangzhou",
            "PercentOfClientLocationImpacted": 2.88,
            "PercentOfTotalTraffic": 0.7799999999999999,
            "Country": "China",
            "Longitude": 120.1612,
```

```
            "AutonomousSystemNumber": 37963,
            "Latitude": 30.2994,
            "Subdivision": "Zhejiang",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        }
    ],
    "StartedAt": "2022-06-20T01:34:20Z",
    "ImpactType": "PERFORMANCE",
    "PercentOfTotalTrafficImpacted": 1.15
}
```

Visualizzazione dell'elenco dei monitoraggi

Per visualizzare un elenco di tutti i monitoraggi del tuo account con la CLI, esegui il comando `list-monitors`.

```
aws internetmonitor list-monitors
```

```
{
    "Monitors": [
        {
            "MonitorName": "TestMonitor",
            "ProcessingStatus": "OK",
            "Status": "ACTIVE"
        }
    ],
    "NextToken": " zase12"
}
```

Modifica del monitoraggio

Per aggiornare le informazioni relative al monitoraggio utilizzando la CLI, usa il comando `update-monitor` e specifica il nome del monitoraggio da aggiornare. È possibile aggiornare la percentuale di traffico da monitorare, il limite del numero massimo di reti urbane da monitorare, aggiungere o rimuovere le risorse che Monitor Internet utilizza per monitorare il traffico e modificare lo stato del monitor da `ACTIVE` a `INACTIVE` o viceversa. Tieni presente che non puoi modificare il nome del monitoraggio.

La risposta a una chiamata `update-monitor` restituisce solo il `MonitorArn` e il `Status`.

L'esempio seguente mostra come utilizzare il comando `update-monitor` per modificare il numero massimo di reti urbane per monitorare in `50000`:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --max-city-networks-to-
monitor 50000
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": " ACTIVE "
}
```

L'esempio seguente mostra come aggiungere e rimuovere risorse:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" \
    --resources-to-add "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --resources-to-remove "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-2222444455556666"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "ACTIVE"
}
```

L'esempio seguente mostra come utilizzare il comando `update-monitor` per modificare lo stato del monitoraggio in `INACTIVE`:

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --status "INACTIVE"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "INACTIVE"
}
```

Eliminazione del monitoraggio

Puoi eliminare un monitoraggio con la CLI tramite il comando `delete-monitor`. Innanzitutto, imposta il monitoraggio come inattivo. Per eseguire questa operazione, usa il comando `update-monitor` per modificare lo stato in `INACTIVE`. Verifica che il monitoraggio sia inattivo controllandone lo stato tramite il comando `get-monitor`.

Quando lo stato del monitoraggio è `INACTIVE`, puoi utilizzare la CLI per eseguire il comando `delete-monitor` ed eliminare così il monitoraggio. La risposta per una chiamata `delete-monitor` riuscita è vuota.

```
aws internetmonitor delete-monitor --monitor-name "TestMonitor"
```

```
{}
```

Monitoraggio e ottimizzazione con il pannello di controllo di Monitor Internet

Le informazioni in questa sezione descrivono come filtrare e visualizzare le informazioni sulla dashboard di Amazon CloudWatch Internet Monitor per visualizzare e ottenere informazioni dettagliate sul traffico Internet e sulla configurazione dell' AWS applicazione.

Dopo aver creato un monitor per monitorare le prestazioni e la disponibilità di Internet dell'applicazione, Amazon CloudWatch Internet Monitor pubblica CloudWatch log contenenti misurazioni Internet per le coppie di localizzazione e rete client (città-rete) e pubblica CloudWatch metriche aggregate per il traffico verso la tua applicazione e verso ciascuna posizione periferica. Regione AWS Puoi filtrare, esplorare e ottenere suggerimenti orientati all'azione da queste informazioni di Monitor Internet in diversi modi.

Per iniziare, sulla console, in Monitoraggio della rete, scegli Internet Monitor CloudWatch .

Questa sezione descrive principalmente come filtrare e visualizzare le metriche di Internet Monitor utilizzando. AWS Management Console In alternativa, è possibile utilizzare le operazioni dell'API di Internet Monitor con AWS CLI o un SDK per lavorare direttamente con gli eventi di Internet Monitor archiviati nei file di CloudWatch registro. Per ulteriori informazioni, consulta Utilizzo del monitor e delle informazioni sulle misurazioni. Per ulteriori informazioni sull'uso delle operazioni API, consulta Esempi di utilizzo della CLI con Amazon Internet Monitor CloudWatch Amazon CloudWatch Internet Monitor API Reference.

Nel pannello di controllo di Monitor Internet sono disponibili tre schede:

- Nella scheda Overview (Panoramica) puoi visualizzare le informazioni correnti e quelle cronologiche sulle prestazioni e sulla disponibilità dell'applicazione, oltre agli eventi di stato che influiscono sulle posizioni dei client.
- Nella scheda successiva, Esploratore cronologico, puoi filtrare per posizione, ASN, data e così via e visualizzare i parametri del traffico Internet nel tempo, utilizzando i grafici.

- Nella scheda Informazioni sul traffico , oltre a visualizzare le informazioni sul traffico monitorato riepilogate in diversi modi personalizzabili, puoi ottenere suggerimenti per configurazioni ottimizzate al fine di migliorare le prestazioni per diverse coppie di posizione e ASN. Internet Monitor prevede il miglioramento delle prestazioni dell'applicazione, in base ai modelli di traffico e alle prestazioni passate, quando si modifica il modo in cui si indirizza il traffico o AWS le risorse utilizzate. Puoi anche visualizzare un grafico per confrontare quante reti urbane sono incluse nella copertura di monitoraggio, in base alla percentuale di traffico dell'applicazione scelta per il monitor.

Inoltre, poiché Internet Monitor genera e pubblica file di registro con le misurazioni del traffico, puoi utilizzare altri CloudWatch strumenti della console per visualizzare ulteriormente i dati pubblicati da Internet Monitor, tra cui CloudWatch Contributor Insights, CloudWatch Metrics e Logs Insights. CloudWatch Per ulteriori informazioni, consulta Esplorazione dei dati con CloudWatch gli strumenti e l'interfaccia di interrogazione di Internet Monitor.

Scopri come usare Monitor Internet per esplorare le misurazioni delle prestazioni e della disponibilità nelle sezioni seguenti.

Argomenti

Monitoraggio delle prestazioni e della disponibilità in tempo reale in Amazon CloudWatch Internet Monitor (scheda Panoramica)

Utilizza la scheda Panoramica nella CloudWatch console, sotto Internet Monitor, per ottenere una visione di alto livello delle prestazioni e della disponibilità del traffico monitorato dal monitor. Viene mostrata inoltre una mappa panoramica del traffico Internet, con cluster di traffico che consentono di visualizzare il traffico globale dell'applicazione e la posizione e l'impatto degli eventi di stato.

Punteggi di integrità

Il grafico Health score mostra le informazioni sulle prestazioni e sulla disponibilità per il traffico globale. AWS dispone di dati storici sostanziali sulle prestazioni e sulla disponibilità di Internet per

il traffico di rete tra località geografiche per diversi ASN e AWS servizi. Internet Monitor utilizza i dati di connettività raccolti dalla sua impronta di rete globale per calcolare una base di prestazioni e disponibilità per il traffico Internet. AWS Si tratta degli stessi dati che utilizziamo AWS per monitorare l'operatività e la disponibilità di Internet.

Con queste misurazioni come baseline, Monitor Internet è in grado di rilevare quando le prestazioni e la disponibilità dell'applicazione diminuiscono a confronto con la baseline. Per facilitare la visualizzazione di questi cali, tali informazioni vengono riportate come un punteggio delle prestazioni e un punteggio di disponibilità. Per ulteriori informazioni, consulta Esplorazione dei dati con CloudWatch gli strumenti e l'interfaccia di interrogazione di Internet Monitor.

Il grafico Punteggi di integrità include gli eventi di integrità che si sono verificati durante un intervallo di tempo scelto. Quando si verifica un evento di stato, viene visualizzato un calo nella riga delle prestazioni o della disponibilità nel grafico. Se selezioni l'evento, puoi ottenere maggiori dettagli e sul grafico vengono visualizzate delle bande, con informazioni su data e ora che mostrano la durata dell'evento.

Puoi anche esaminare queste metriche accedendo direttamente ai file di log per ogni punto dati. Nel menu Azioni, scegli Visualizza CloudWatch registri.

Panoramica del traffico Internet

La mappa Panoramica del traffico Internet mostra il traffico Internet e gli eventi di integrità specifici delle posizioni e gli ASN da cui gli utenti accedono all'applicazione. I paesi in grigio sulla mappa sono quelli che includono il traffico dell'applicazione.

Ogni cerchio sulla mappa indica un evento di integrità in un'area, in un periodo di tempo selezionato. Internet Monitor crea eventi sanitari quando rileva un problema, a una soglia specifica, di connettività tra una delle risorse ospitate AWS e una rete urbana in cui un utente accede all'applicazione. Scegliendo un cerchio sulla mappa vengono visualizzati maggiori dettagli sull'evento di stato relativo a quella località. Inoltre, per i cluster che presentano eventi di integrità, puoi visualizzare informazioni dettagliate nella tabella Health events (Eventi di stato) sotto la mappa.

Monitor Internet crea e risolve gli eventi di integrità in un monitoraggio quando determina che un evento sta avendo un impatto globale significativo sull'applicazione. Se non sono presenti eventi di integrità che superano la soglia per l'impatto sul traffico per le posizioni dei client nel periodo di tempo selezionato, la mappa è vuota. Per ulteriori informazioni, consulta Quando Monitor Internet crea e risolve eventi di integrità.

Modifica delle soglie degli eventi di integrità

È possibile configurare diverse opzioni relative a come e quando Monitor Internet crea eventi di integrità per l'applicazione. Scegli Aggiorna soglie per apportare le modifiche.

Puoi modificare la soglia generale che attiva Monitor Internet per creare un evento di integrità. La soglia predefinita per gli eventi di integrità è del 95% sia per i punteggi delle prestazioni che per i punteggi di disponibilità. Cioè, quando il punteggio complessivo di prestazioni o disponibilità dell'applicazione scende al 95% o inferiore, Monitor Internet crea un evento relativo allo stato di integrità. Per quanto riguarda la soglia generale, l'evento di integrità può essere attivato da un singolo problema di grandi dimensioni o dalla combinazione di più problemi minori.

È inoltre possibile modificare la soglia locale, ovvero la rete urbana, combinandola con una percentuale del livello di impatto complessivo, in modo da innescare un evento di integrità. Impostando una soglia che crei un evento di integrità quando un punteggio scende al di sotto della soglia per una o più reti urbane (posizioni e ASN, in genere ISP), puoi ottenere informazioni dettagliate su quando si verificano problemi in posizioni con traffico inferiore, ad esempio.

Un'opzione di soglia locale aggiuntiva funziona insieme alla soglia locale per i punteggi di disponibilità o prestazioni. Il secondo fattore è la percentuale del traffico complessivo che deve essere influenzata prima che Monitor Internet crei un evento di integrità basato sulla soglia locale.

Configurando le opzioni di soglia per il traffico complessivo e il traffico locale, è possibile ottimizzare la frequenza con cui vengono creati gli eventi di integrità in base all'utilizzo dell'applicazione e alle varie esigenze. Tieni presente che quando imposti una soglia locale su un valore inferiore, in genere vengono creati più eventi di integrità a seconda dell'applicazione e degli altri valori di configurazione della soglia impostati.

In sintesi, è possibile configurare le soglie degli eventi di integrità per i punteggi di prestazioni, i punteggi di disponibilità o entrambi, nei seguenti modi:

- Scegli soglie globali diverse per attivare un evento di integrità.
- Scegli soglie locali diverse per attivare un evento di integrità. Con questa opzione, puoi anche modificare la percentuale di impatto sull'intera applicazione che deve essere superata prima che Monitor Internet crei un evento.
- Scegli di disattivare l'attivazione di un evento di integrità in base alle soglie locali o di abilitare le opzioni di soglia locali.

Puoi anche configurare le opzioni per i punteggi di prestazioni, i punteggi di disponibilità o entrambi. È possibile configurare una combinazione di opzioni o solo una di esse.

Per aggiornare le soglie e altre opzioni di configurazione per i punteggi di prestazioni, i punteggi di disponibilità o entrambi, procedi come segue:

Modifica delle opzioni di configurazione delle soglie

1. Nel AWS Management Console, accedi a CloudWatch, quindi, nel riquadro di navigazione a sinistra, scegli Internet Monitor.
2. Nella scheda Panoramica, nella sezione Cronologia degli eventi di integrità, scegli Aggiorna soglie.
3. Nella pagina di dialogo che viene aperta, scegli i nuovi valori e le opzioni che desideri per le soglie e le altre opzioni che attivano Monitor Internet per creare un evento di integrità. Puoi effettuare le seguenti operazioni:
   - Scegli un nuovo valore per Soglia del punteggio di disponibilità, Soglia del punteggio di prestazione o entrambi i cambi.

     I grafici nelle sezioni relative a ciascuna impostazione mostrano l'impostazione della soglia corrente e i punteggi effettivi degli eventi di integrità recenti relativi alla disponibilità o alle prestazioni, per l'applicazione. Visualizzando i valori tipici, è possibile farsi un'idea dei valori in base ai quali è possibile modificare una soglia.

     Suggerimento: per visualizzare un grafico più grande e modificare l'intervallo di tempo, scegli lo strumento di espansione nell'angolo in alto a destra del grafico.
   - Scegli di attivare o disattivare una soglia locale per la disponibilità o le prestazioni, o entrambe. Quando un'opzione è abilitata, puoi impostare la soglia e il livello di impatto per quando desideri che Monitor Internet crei un evento di integrità.
4. Dopo aver configurato le opzioni di soglia, salva gli aggiornamenti selezionando Aggiorna le soglie degli eventi di integrità.

Per ulteriori informazioni su come funzionano gli eventi di integrità, consulta Quando Monitor Internet crea e risolve gli eventi di integrità.

Tabella degli eventi di integrità

La tabella Eventi di integrità riporta le posizioni dei client interessati dagli eventi di integrità e le informazioni relative a tali eventi. Nella tabella sono incluse le colonne seguenti.

	Descrizione
Client location (Posizione del client)	La posizione degli utenti finali interessati dall'evento che hanno riscontrato una maggiore latenza o una disponibilità ridotta. Per ulteriori informazioni sulla precisione della posizione dei client in Monitor Internet, consulta Informazioni e precisione sulla geolocalizzazione in Monitor Internet.
Traffic impact (Impatto sul traffico)	L'entità dell'impatto causato dall'evento, in termini di maggiore latenza o ridotta disponibilità. Per quanto riguarda la latenza, si tratta della percentuale di aumento della latenza durante l'evento rispetto alle prestazioni tipiche del traffico, da questa posizione del client a questa AWS posizione utilizzando questa rete client.
Client network (Rete del client)	La rete su cui viaggiava il traffico. In genere, è il provider di servizi Internet (ISP) o il numero di sistema autonomo (ASN) per il traffico di rete.
AWS posizione	La AWS posizione per il traffico di rete, che può essere una Regione AWS o una edge location Internet.
Impact type (Tipo di impatto)	Il tipo di impatto dell'evento di stato. Gli eventi di stato sono in genere causati da aumenti di latenza (problemi di prestazioni) o raggiungibilità (problemi di disponibilità). Potresti anche essere in grado di fare clic sul tipo di impatto per vedere la causa della compromissione. Quando possibile, Internet Monitor analizza l'origine di un

	Descrizione
	evento sanitario per determinare se è stato causato da AWS o da un ASN (internet service provider). Si noti che questa analisi continua dopo la risoluzione dell'evento. Monitor Internet può aggiornare gli eventi con nuove informazioni per un massimo di un'ora.

Se scegli una delle posizioni del client nella tabella Eventi di integrità, puoi visualizzare maggiori dettagli sull'evento di integrità in tale posizione. Ad esempio, puoi vedere quando l'evento è iniziato, quando è terminato e l'impatto sul traffico locale.

Visualizzazione del percorso di rete

L'analisi dei problemi completata ha un percorso di rete completo in Visualizzazione del percorso di rete. Il percorso completo mostra ogni nodo lungo il percorso di rete dell'applicazione per l'evento sanitario, tra la AWS posizione e il client, per una coppia client-posizione.

Se Monitor Internet determina la causa di una compromissione, questa viene contrassegnata da un cerchio rosso tratteggiato. I problemi possono essere causati dagli ASN, in genere dai provider di servizi Internet (ISP), oppure la causa può essere AWS. Se le cause della compromissione sono molteplici, vengono cerchiati più nodi.

Filtraggio e visualizzazione dei dati storici in Amazon CloudWatch Internet Monitor (scheda Historical explorer)

Usa la scheda Historical explorer nella CloudWatch console, sotto Internet Monitor, per filtrare e visualizzare i dati della tua applicazione che si trova in CloudWatch Logs. Internet Monitor pubblica CloudWatch nei log le misurazioni specifiche dell'applicazione relative alla disponibilità, alle prestazioni, ai byte monitorati trasferiti (o al numero di connessioni client, solo per WorkSpaces le directory) e al tempo di andata e ritorno per le reti urbane monitorate. Regioni AWS

> **Note**
>
> Internet Monitor pubblica le misurazioni Internet CloudWatch nei registri ogni cinque minuti per le prime 500 reti urbane (in base al volume di traffico) (ovvero posizioni dei client e ASN, in genere provider di servizi Internet o ISP) che inviano traffico a ciascun monitor. Facoltativamente, puoi decidere di pubblicare misurazioni ed eventi Internet per tutte le reti urbane monitorate (fino al limite di servizio di 500.000 reti urbane) in un bucket Amazon S3. Per ulteriori informazioni, consulta Pubblicazione di misurazioni Internet su Amazon S3 in Amazon CloudWatch Internet Monitor.

Per iniziare a esplorare i dati dell'applicazione, seleziona un periodo di tempo. Quindi scegli una posizione geografica specifica, ad esempio una città, e facoltativamente, altri filtri. Monitor Internet applica i filtri ai dati nei log delle misurazioni di Internet che ha pubblicato per le reti delle città per il traffico delle tue applicazioni. Quindi puoi vedere i grafici dei dati che mostrano il punteggio delle prestazioni, il punteggio di disponibilità, i byte monitorati trasferiti (per VPC, Network Load Balancer e CloudFront distribuzioni) o il numero di connessioni client (per WorkSpaces le directory) e il tempo di andata e ritorno (RTT) per la tua applicazione nel tempo.

La tabella All events (Tutti gli eventi) sotto i grafici mostra gli eventi di stato restituiti dal per il traffico dell'applicazione, con informazioni su ciascun evento. Sono incluse le seguenti colonne.

	Descrizione
Event start (Inizio dell'evento)	Ora di inizio dell'evento di stato.
Stato	Se l'evento è ancora attivo o è stato risolto.
Client location (Posizione del client)	La posizione degli utenti finali interessati dall'evento che hanno riscontrato un aumento della latenza o una riduzione delle prestazioni. Per ulteriori informazioni sulla precisione della posizione dei client in Monitor Internet, consulta Informazioni e precisione sulla geolocalizzazione in Monitor Internet.

	Descrizione
Traffic impact (Impatto sul traffico)	L'entità dell'impatto dell'evento sulla posizione dell'evento di integrità. Questo è, ad esempio, l'impatto sulla latenza, rispetto alle prestazio ni tipiche del traffico da una posizione del client alla AWS posizione tramite l'ASN del client, in genere un provider di servizi Internet (ISP). Analogamente, per un evento che influisce sulla disponibilità, si nota l'impatto sulla disponibilità rispetto alla disponibilità tipica della posizione del client per la sede tramite l' AWS ASN del client.
Event duration (Durata dell'evento)	Quanto è durato l'evento. Monitor Internet termina gli eventi di stato quando non interessa no più del 5% in totale delle posizioni dei client dell'applicazione.
Client ISP (ISP del client)	L'ASN, in genere il provider di servizi Internet (ISP), che era il gestore del traffico di rete.
Service location (Posizione del servizio)	La posizione del servizio da cui ha avuto origine il traffico di rete, che può essere una Regione AWS o una edge location Internet.

In alternativa, puoi esaminare le misurazioni dell'applicazione accedendo direttamente ai log per ogni punto dati. Nel menu Azioni, scegli Visualizza CloudWatch registri. Tieni presente che, poiché gli eventi di misurazione vengono pubblicati sul tuo account al momento della creazione, puoi creare anche altri CloudWatch dashboard o allarmi basati su di essi. Per ulteriori informazioni, consulta Ottenere informazioni per migliorare le prestazioni delle applicazioni in Amazon CloudWatch Internet Monitor (scheda Traffic insights) e Creazione di allarmi con Amazon CloudWatch Internet Monitor.

Oltre a esplorare e analizzare i parametri e le misurazioni di Monitor Internet e a creare pannelli di controllo e allarmi basati su di essi, puoi utilizzare Monitor Internet per comprendere come ottimizzare le prestazioni dell'applicazione. La scheda Traffic insights (Informazioni sul traffico) offre diversi modi per aiutarti a esplorare le opzioni. Per ulteriori informazioni, consulta Suggerimenti per

l'ottimizzazione del traffico nella scheda Informazioni sul traffico. Inoltre, puoi vedere gli esempi specifici nel capitolo sui casi d'uso di Monitor Internet.

Ottenere informazioni per migliorare le prestazioni delle applicazioni in Amazon CloudWatch Internet Monitor (scheda Traffic insights)

Usa la scheda Traffic Insights nella CloudWatch console, sotto Internet Monitor, per visualizzare le informazioni di riepilogo sul traffico principale (per volume) per la tua applicazione. È possibile filtrare e ordinare il traffico delle applicazioni in diversi modi. Quindi, scorri verso il basso e seleziona diverse combinazioni di configurazioni per l'applicazione al fine di visualizzare le alternative ottimali suggerite da Monitor Internet per ottenere le migliori prestazioni TTFB (Time to First Byte).

Internet Monitor pubblica su CloudWatch Logs le misurazioni Internet ogni cinque minuti per le prime 500 reti urbane (per volume di traffico) (ovvero posizioni dei clienti e ASN, in genere provider di servizi Internet o ISP) che inviano traffico a ciascun monitor. Facoltativamente, puoi decidere di pubblicare misurazioni ed eventi Internet per tutte le reti urbane monitorate (fino al limite di servizio di 500.000 reti urbane) in un bucket Amazon S3. Per ulteriori informazioni, consulta Pubblicazione di misurazioni Internet su Amazon S3 in Amazon CloudWatch Internet Monitor.

Riepiloghi del traffico principale

Puoi iniziare visualizzando i riepiloghi generali del traffico e delle prestazioni complessive dell'applicazione, in un intervallo di tempo specifico, filtrati per posizione del client. Puoi anche esaminare le prestazioni della tua applicazione per le principali (o ultime) posizioni dei clienti in base al volume di traffico, filtrate e ordinate in diversi modi. Ad esempio, puoi ordinare per granularità (ovvero città, regione, Paese o area metropolitana), per traffico totale, tempo medio del primo byte (TTFB) e altri fattori.

Per ulteriori informazioni sulla precisione della posizione dei client in Monitor Internet, consulta Informazioni e precisione sulla geolocalizzazione in Monitor Internet.

> **Note**
>
> I filtri utilizzati si applicano all'intera pagina, quindi influiscono sulle reti urbane incluse nei grafici di riepilogo e nelle informazioni sul traffico totale, nonché sulle reti urbane incluse nella sezione Suggerimenti per l'ottimizzazione del traffico che segue.

Suggerimenti di ottimizzazione del traffico

La sezione Suggerimenti per l'ottimizzazione del traffico mostra un insieme filtrato di reti urbane monitorate (posizioni e ASN, provider di servizi Internet) per il traffico, insieme al traffico totale dei clienti per ciascuna di esse. Le voci della tabella si basano sui filtri che hai scelto per il traffico dell'applicazione per Informazioni sul traffico nella parte superiore della pagina. L'impostazione predefinita sono le prime 10 città per volume di traffico. In genere nella tabella vengono visualizzate più di 10 righe, poiché esiste una voce per ogni coppia unica città-rete. In altre parole, esiste una riga per ogni combinazione di posizione (città) e ASN (provider di rete) tramite cui i client accedono all'applicazione, ad esempio Dallas, Texas, USA e Comcast.

> **Note**
>
> Per visualizzare i suggerimenti di ottimizzazione del traffico per tutte le reti urbane monitorate, puoi eseguire una query direttamente in Insights. CloudWatch Per una query di esempio che non include il filtro di granularità geografica che limita l'elenco delle reti urbane in questa pagina, consulta Utilizzo di CloudWatch Logs Insights con Amazon CloudWatch Internet Monitor.

In questa sezione, seleziona diverse opzioni: Amazon EC2 o CloudFront entrambe. In questo modo puoi vedere quali sono i valori del tempo medio previsto per il primo byte (TTFB) per i client quando utilizzi la tua applicazione con tali servizi in diverse AWS regioni, rispetto all'attuale TTFB. Per ulteriori informazioni sui calcoli TTFB, consulta Calcoli di AWS per TTFB e latenza.

Selezionando diverse opzioni e visualizzando quindi i risultati nella tabella, puoi iniziare a pianificare configurazioni e implementazioni in grado di migliorare le prestazioni dei tuoi client. Tieni presente che potresti vedere un trattino (-) anziché un valore in una colonna quando i dati non sono disponibili per la visualizzazione. Per esaminare un esempio specifico di come migliorare le prestazioni, consulta Usare Amazon CloudWatch Internet Monitor per una migliore esperienza di gioco.

Ad esempio, per iniziare, per una rete urbana specifica (posizione del client e coppia ASN), prova a selezionare l'EC2 o l' CloudFront opzione o entrambe. Per ogni rete urbana elencata nella tabella, Internet Monitor mostra i potenziali miglioramenti delle prestazioni del TTFB, sulla base di una scelta di routing del traffico (attraverso una specifica opzione Regione AWS) rispetto alla configurazione corrente. (Si noti che, per completezza, la tabella include anche percorsi già ottimizzati.) Ad esempio, potresti vedere un TTFB medio previsto di 50 ms per l'utilizzo del routing

EC2 attraverso `us-east-1` rispetto alla configurazione attuale con un TTFB di 100 ms dove si utilizza il routing EC2 attraverso `us-west-2`. Quindi potresti prendere in considerazione l'idea di instradare tramite `us-west-2`.

Come altro esempio, potresti selezionare EC2 e poi vedere che non comporta una differenza misurabile in termini di prestazioni per una sede client e un ASN, ma poi tieni presente che quando selezioni CloudFront con la stessa regione, riduce leggermente il TTFB. Ciò suggerisce che se aggiungi una CloudFront distribuzione davanti all'applicazione, ciò potrebbe comportare un miglioramento delle prestazioni e potrebbe valere la pena provare, per questa ubicazione del client e per l'ASN.

Esplorazione dei dati con CloudWatch gli strumenti e l'interfaccia di interrogazione di Internet Monitor

Oltre a visualizzare le prestazioni e la disponibilità della tua applicazione con la dashboard di Amazon CloudWatch Internet Monitor, puoi utilizzare diversi metodi per approfondire i dati che Internet Monitor genera per te. Questi metodi includono l'utilizzo di CloudWatch strumenti con dati di Internet Monitor archiviati in file di CloudWatch registro e l'utilizzo dell'interfaccia di interrogazione di Internet Monitor. Gli strumenti che puoi utilizzare includono CloudWatch Logs Insights, CloudWatch Metrics, CloudWatch Contributor Insights e Amazon Athena. È possibile utilizzare alcuni o tutti questi strumenti, oltre al pannello di controllo, per esplorare i dati di Monitor Internet in base alle proprie esigenze.

Internet Monitor aggrega le CloudWatch metriche sul traffico verso la tua applicazione e verso ciascuna di esse e include dati come l'impatto totale sul traffico Regione AWS, la disponibilità e il tempo di andata e ritorno. Questi dati vengono pubblicati in CloudWatch Logs e possono essere utilizzati anche con l'interfaccia di interrogazione di Internet Monitor. I dettagli sulla geogranularità e altri aspetti delle informazioni che è possibile esplorare variano in base al tipo.

Amazon CloudWatch Internet Monitor pubblica i dati per il monitor a intervalli di 5 minuti, quindi li rende disponibili in diversi modi. La tabella seguente elenca gli scenari di accesso ai dati di Monitor Internet e descrive le funzionalità dei dati raccolti per ognuno di essi.

Funzionalità	CloudWatch Registri	Esportazione in S3	Interfaccia di interrogazione	CloudWatch cruscotto
Abilitata per default	Sì	No	Sì	Sì
Numero di reti urbane per le quali vengono raccolti i dati	Prime 500 (vedi nota sotto)	Tutti	Tutti	Tutti
Conservazione dei dati	Controllato dall'utente	Controllato dall'utente	30 giorni	30 giorni
Geogranularità per cui vengono raccolti i dati	Tutti (rete urbana, rete metropolitana, rete regionale, rete nazionale)	Rete urbana	Tutti (rete urbana, rete metropolitana, rete regionale, rete nazionale)	Tutti (rete urbana, rete metropolitana, rete regionale, rete nazionale)
Come interrogare e filtrare i dati	Utilizzo di CloudWatch Logs Insights con Amazon CloudWatch Internet Monitor	Utilizzo di Amazon Athena per eseguire query delle misurazioni Internet nei file di log di Amazon S3	Utilizzo dell'interfaccia di interrogazione Amazon CloudWatch Internet Monitor	Monitoraggio e ottimizzazione con il pannello di controllo di Monitor Internet

Nota: per le reti urbane vengono acquisite le prime 500 misurazioni, per le reti metropolitane le prime 250, per le reti regionali le prime 100 e per le reti nazionali le prime 50.

Questo capitolo descrive come interrogare ed esplorare i dati utilizzando CloudWatch gli strumenti o l'interfaccia di interrogazione di Internet Monitor, insieme ad esempi per ogni metodo.

Indice

- Utilizzo di Contributor Insights con Amazon CloudWatch Internet Monitor
- Utilizzo di CloudWatch metriche con Amazon CloudWatch Internet Monitor
- Utilizzo di Amazon Athena per eseguire query delle misurazioni Internet nei file di log di Amazon S3
- Utilizzo dell'interfaccia di interrogazione Amazon CloudWatch Internet Monitor

Utilizzo di CloudWatch Logs Insights con Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor pubblica misurazioni granulari della disponibilità e del tempo di andata e ritorno su CloudWatch Logs e puoi utilizzare le query CloudWatch Logs Insights per filtrare un sottoinsieme di log per una città o area geografica specifica (posizione del cliente), ASN (ISP) del client e posizione di origine. AWS

Per ulteriori informazioni sulla precisione della posizione dei client in Monitor Internet, consulta Informazioni e precisione sulla geolocalizzazione in Monitor Internet.

Gli esempi in questa sezione possono aiutarti a creare query CloudWatch Logs Insights per saperne di più sulle misurazioni e le metriche del traffico delle tue applicazioni. Se utilizzi questi esempi in CloudWatch Logs Insights, sostituisci *`monitorName con il nome del tuo monitor`*.

Visualizzazione dei suggerimenti di ottimizzazione del traffico

Nella scheda Informazioni sul traffico di Monitor Internet, puoi visualizzare i suggerimenti per l'ottimizzazione del traffico, filtrati per posizione. Per visualizzare le stesse informazioni visualizzate nella sezione Suggerimenti per l'ottimizzazione del traffico di quella scheda, ma senza il filtro di granularità della posizione, puoi utilizzare la seguente CloudWatch query di Logs Insights.

1. In AWS Management Console, vai a Logs Insights. CloudWatch
2. Per Log Group (Gruppo di log), seleziona `/aws/internet-monitor/`*`monitorName`*`/byCity` e `/aws/internet-monitor/`*`monitorName`*`/byCountry`, quindi specifica un intervallo di tempo.
3. Aggiungi la query seguente, quindi eseguila.

```
fields @timestamp,
clientLocation.city as @city, clientLocation.subdivision as @subdivision,
 clientLocation.country as @country,
`trafficInsights.timeToFirstByte.currentExperience.serviceName` as @serviceNameField,
concat(@serviceNameField, ` (`, `serviceLocation`, `)`) as @currentExperienceField,
```

```
concat(`trafficInsights.timeToFirstByte.ec2.serviceName`, ` (`,
 `trafficInsights.timeToFirstByte.ec2.serviceLocation`, `)`) as @ec2Field,
`trafficInsights.timeToFirstByte.cloudfront.serviceName` as @cloudfrontField,
concat(`clientLocation.networkName`, ` (AS`, `clientLocation.asn`, `)`) as @networkName
| filter ispresent(`trafficInsights.timeToFirstByte.currentExperience.value`)
| stats avg(`trafficInsights.timeToFirstByte.currentExperience.value`) as @averageTTFB,
avg(`trafficInsights.timeToFirstByte.ec2.value`) as @ec2TTFB,
avg(`trafficInsights.timeToFirstByte.cloudfront.value`) as @cloudfrontTTFB,
sum(`bytesIn` + `bytesOut`) as @totalBytes,
latest(@ec2Field) as @ec2,
latest(@currentExperienceField) as @currentExperience,
latest(@cloudfrontField) as @cloudfront,
count(*) by @networkName, @city, @subdivision, @country
| display @city, @subdivision, @country, @networkName, @totalBytes, @currentExperience,
 @averageTTFB, @ec2, @ec2TTFB, @cloudfront, @cloudfrontTTFB
| sort @totalBytes desc
```

Visualizza la disponibilità di Internet e RTT (p50, p90 e p95)

Per visualizzare la disponibilità di Internet e l'orario di andata e ritorno (p50, p90 e p95) per il traffico, puoi utilizzare la seguente query di Logs Insights. CloudWatch

Area geografica dell'utente finale: Chicago, IL, Stati Uniti

Rete dell'utente finale (ASN): AS7018

AWS ubicazione del servizio: regione Stati Uniti orientali (Virginia settentrionale)

Per visualizzare i log, procedi come indicato di seguito:

1. Nella AWS Management Console, accedi a CloudWatch Logs Insights.
2. Per Log Group (Gruppo di log), seleziona `/aws/internet-monitor/`*`monitorName`*`/byCity` e `/aws/internet-monitor/`*`monitorName`*`/byCountry`, quindi specifica un intervallo di tempo.
3. Aggiungi la query seguente, quindi eseguila.

La query restituisce tutti i dati sulle prestazioni degli utenti che si connettono da AS7018 a Chicago, IL verso la regione Stati Uniti orientali (Virginia settentrionale) nel periodo di tempo selezionato.

```
fields @timestamp,
internetHealth.availability.experienceScore as availabilityExperienceScore,
```

```
internetHealth.availability.percentageOfTotalTrafficImpacted as
 percentageOfTotalTrafficImpacted,
internetHealth.performance.experienceScore as performanceExperienceScore,
internetHealth.performance.roundTripTime.p50 as roundTripTimep50,
internetHealth.performance.roundTripTime.p90 as roundTripTimep90,
internetHealth.performance.roundTripTime.p95 as roundTripTimep95
 | filter clientLocation.country == `United States`
 and clientLocation.city == `Chicago`
 and serviceLocation == `us-east-1`
 and clientLocation.asn == 7018
```

Per ulteriori informazioni, consulta Analisi dei dati di registro con CloudWatch Logs Insights.

Utilizzo di Contributor Insights con Amazon CloudWatch Internet Monitor

CloudWatch Contributor Insights può aiutarti a identificare le sedi e le reti dei clienti principali (ASN o provider di servizi Internet) per la tua applicazione. Utilizza le seguenti regole di esempio per Contributor Insights per iniziare a usare regole utili con Amazon CloudWatch Internet Monitor. Per ulteriori informazioni, consulta Creazione di una regola di Approfondimenti sulle contribuzioni.

Per ulteriori informazioni sulla precisione della posizione dei client in Monitor Internet, consulta Informazioni e precisione sulla geolocalizzazione in Monitor Internet.

> **Note**
>
> Monitor Internet pubblica i dati ogni cinque minuti, quindi dopo aver configurato una regola di Approfondimenti sulle contribuzioni, devi impostare il periodo su cinque minuti per visualizzarne il grafico.

Visualizza le posizioni e gli ASN principali interessati da un impatto sulla disponibilità

Per visualizzare le posizioni client e gli ASN principali interessati da un calo della disponibilità, puoi utilizzare la seguente regola di Contributor Insights nell'editor della sintassi. Sostituisci *monitor-name* con il nome del tuo monitoraggio.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
```

```
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.availability.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Visualizza le posizioni client e gli ASN principali interessati da un impatto sulla latenza

Per visualizzare le posizioni client e gli ASN principali interessati da un aumento del tempo di andata e ritorno (latenza), puoi utilizzare la seguente regola di Contributor Insights nell'editor della sintassi. Sostituisci *monitor-name* con il nome del tuo monitoraggio.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.performance.percentageOfTotalTrafficImpacted"
    },
```

```
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Visualizza le posizioni client e gli ASN principali interessati dalla percentuale totale di traffico

Per visualizzare le posizioni client e gli ASN principali interessati dalla percentuale totale di traffico, puoi utilizzare la seguente regola di Contributor Insights nell'editor della sintassi. Sostituisci *monitor-name* con il nome del tuo monitoraggio.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.percentageOfTotalTraffic"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Utilizzo di CloudWatch metriche con Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor pubblica parametri sul tuo account, tra cui parametri relativi a prestazioni, disponibilità, tempo di andata e ritorno e velocità effettiva (byte al secondo), che puoi visualizzare in Metrics nella console. CloudWatch CloudWatch Per trovare tutte le metriche per il

tuo monitor, nella dashboard Metrics, consulta lo spazio dei nomi personalizzato. CloudWatch `AWS/InternetMonitor`

Le metriche vengono aggregate su tutto il traffico Internet diretto ai VPC, ai Network Load Balancer, alle CloudFront distribuzioni o alle WorkSpaces directory del monitor e su tutto il traffico verso ciascuna e l'edge location Internet monitorata. Regione AWS Le regioni sono definite dalla posizione del servizio, che può essere costituita da tutte le posizioni o da una regione specifica, ad esempio `us-east-1`.

Nota: le reti urbane sono posizioni dei client e ASN (in genere provider di servizi Internet o ISP).

Monitor Internet fornisce i seguenti parametri.

Parametro	Descrizione
PerformanceScore	Un punteggio delle prestazioni rappresenta la percentuale stimata di traffico che non registra un calo delle prestazioni.
AvailabilityScore	Un punteggio di disponibilità rappresenta la percentuale stimata di traffico che non registra un calo di disponibilità.
BytesIn	Byte trasferiti per il traffico Internet dell'appl icazione in tutte le reti urbane dell'applicazione.
BytesOut	Byte trasferiti in uscita per il traffico Internet dell'applicazione in tutte le reti urbane dell'appl icazione.
BytesInMonitored	Byte trasferiti in ingresso per il traffico Internet dell'applicazione in tutte le reti urbane monitorat e.
BytesOutMonitored	Byte trasferiti in uscita per il traffico Internet dell'applicazione in tutte le reti urbane monitorat e.
Tempo di andata e ritorno (RTT)	Tempo di andata e ritorno tra gli Regioni AWS ASN (in genere i provider di servizi Internet o

Parametro	Descrizione
	ISP) e le località (come le città) specifiche dei tuoi VPC, Network Load Balancer, distribuzioni o directory. CloudFront WorkSpaces
CityNetworksMonitored	Il numero di reti urbane di Monitor Internet monitorate per il traffico Internet delle applicazioni. Questo valore non è mai maggiore del limite superiore impostato come numero massimo di reti urbane per il monitor.
TrafficMonitoredPercent	La percentuale del traffico Internet totale dell'applicazione per questo monitor rappresentata (inclusa) dalle reti urbane monitorate da Monitor Internet. È inferiore a 100 (ovvero meno del 100%) se i client accedono all'applicazione da un numero di reti urbane superiore al limite massimo di reti urbane impostato per il monitor.
CityNetworksFor100 PercentTraffic	Il numero su cui impostare il limite massimo delle reti urbane se desideri monitorare il 100% del traffico Internet della tua applicazione in Monitor Internet.
CityNetworksFor99 PercentTraffic	Il numero su cui impostare il limite massimo delle reti urbane se desideri monitorare il 99% del traffico Internet della tua applicazione in Monitor Internet.
CityNetworksFor95 PercentTraffic	Il numero su cui impostare il limite massimo delle reti urbane se desideri monitorare il 95% del traffico Internet della tua applicazione in Monitor Internet.

Parametro	Descrizione
CityNetworksFor90 PercentTraffic	Il numero su cui impostare il limite massimo delle reti urbane se desideri monitorare il 90% del traffico Internet della tua applicazione in Monitor Internet.
CityNetworksFor75 PercentTraffic	Il numero su cui impostare il limite massimo delle reti urbane se desideri monitorare il 75% del traffico Internet della tua applicazione in Monitor Internet.
CityNetworksFor50 PercentTraffic	Il numero su cui impostare il limite massimo delle reti urbane se desideri monitorare il 50% del traffico Internet della tua applicazione in Monitor Internet.
CityNetworksFor25 PercentTraffic	Il numero su cui impostare il limite massimo delle reti urbane se desideri monitorare il 25% del traffico Internet della tua applicazione in Monitor Internet.

> **Note**
>
> Per esempi di utilizzo di diversi parametri per determinare i valori da scegliere per il massimo delle reti urbane per il monitor, consulta Scelta di un valore massimo per la rete urbana.

Per ulteriori informazioni, consulta Usa i CloudWatch parametri di Amazon.

Utilizzo di Amazon Athena per eseguire query delle misurazioni Internet nei file di log di Amazon S3

Puoi usare Amazon Athena per interrogare e visualizzare le misurazioni Internet che Amazon CloudWatch Internet Monitor pubblica su un bucket Amazon S3. In Monitor Internet è disponibile un'opzione per pubblicare le misurazioni Internet della tua applicazione in un bucket S3 per il traffico connesso a Internet per le reti urbane monitorate (posizioni dei clienti e ASN, in genere provider di

servizi Internet o ISP). Indipendentemente dal fatto che tu scelga di pubblicare le misurazioni su S3, Internet Monitor pubblica automaticamente le misurazioni Internet su CloudWatch Logs ogni cinque minuti per le prime 500 reti urbane (per volume di traffico) per ogni monitor.

Questo capitolo include i passaggi su come creare una tabella in Athena per le misurazioni su Internet che si trova in un file di log S3, quindi fornisce query di esempio per visualizzare diverse visualizzazioni delle misurazioni. Ad esempio, puoi interrogare le 10 reti urbane più colpite dall'impatto sulla latenza.

Utilizzo di Amazon Athena per creare una tabella per le misurazioni Internet in Monitor Internet

Per iniziare a utilizzare Athena con i file di log di S3 per Monitor Internet, devi prima creare una tabella per le misurazioni Internet.

Segui i passaggi di questa procedura per creare una tabella in Athena basata sui file di log di S3. Quindi, puoi eseguire query Athena sulla tabella, come questi esempi di query di misurazione su Internet, per ottenere informazioni sulle tue misurazioni.

Creazione di una tabella Athena

1. Apri la console Athena all'indirizzo https://console.aws.amazon.com/athena/.
2. Nell'editor di query Athena, inserisci un'istruzione di query per generare una tabella con le misurazioni Internet di Monitor Internet. Sostituisci il valore del parametro LOCATION con la posizione del bucket S3 in cui sono archiviate le misurazioni Internet di Monitor Internet.

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/'
```

```
TBLPROPERTIES ('skip.header.line.count' = '1');
```

3. Inserisci una istruzione per creare una partizione per leggere i dati. Ad esempio, la seguente query crea una singola partizione per una data e una posizione specificate:

```
ALTER TABLE internet_measurements
ADD PARTITION (year = 'YYYY', month = 'MM', day = 'dd')
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/YYYY/
MM/DD';
```

4. Seleziona Esegui.

Esempi di istruzioni Athena per misurazioni Internet

Di seguito è riportato un esempio di istruzione per generare una tabella:

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

Di seguito è riportato un esempio di istruzione per creare una partizione per leggere i dati:

```
ALTER TABLE internet_measurements
ADD PARTITION (year = '2023', month = '04', day = '07')
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/2023/04/07/'
```

Esempi di query Amazon Athena da utilizzare con le misurazioni Internet in Monitor Internet

Questa sezione include query di esempio che puoi utilizzare con Amazon Athena per ottenere informazioni sulle misurazioni Internet della tua applicazione pubblicate su Amazon S3.

Interroga le 10 posizioni client e gli ASN più interessati (in termini di percentuale totale di traffico)

Esegui questa query Athena per restituire le 10 principali reti urbane interessate (in termini di percentuale totale di traffico), ovvero le posizioni dei client e gli ASN, in genere provider di servizi Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(percentageoftotaltraffic) as percentageoftotaltraffic
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageoftotaltraffic desc
limit 10
```

Interroga le 10 posizioni client e gli ASN più interessati (in termini di disponibilità)

Esegui questa query Athena per restituire le 10 principali reti urbane interessate (in termini di percentuale totale di traffico), ovvero le posizioni dei client e gli ASN, in genere provider di servizi Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
                '$.availability.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Interroga le 10 posizioni client e gli ASN più interessati (in termini di latenza)

Esegui questa query Athena per restituire le 10 principali reti urbane interessate (in termini di impatto di latenza), ovvero le posizioni dei client e gli ASN, in genere provider di servizi Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
                '$.performance.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Interroga i dati salienti del traffico relativi alle posizioni dei client e agli ASN

Esegui questa query Athena per restituire i dati salienti del traffico, tra cui il punteggio di disponibilità, il punteggio delle prestazioni e il time-to-first byte per le reti urbane, ovvero le posizioni dei client e gli ASN, in genere i provider di servizi Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.subdivision') as subdivision,
    json_extract_scalar(clientLocation, '$.country') as country,
    avg(cast(json_extract_scalar(internetHealth, '$.availability.experiencescore') as
 double)) as availabilityScore,
    avg(cast(json_extract_scalar(internetHealth, '$.performance.experiencescore') as
 double)) performanceScore,
    avg(cast(json_extract_scalar(trafficinsights,
 '$.timetofirstbyte.currentexperience.value') as double)) as averageTTFB,
    sum(bytesIn) as bytesIn,
    sum(bytesOut) as bytesOut,
    sum(bytesIn + bytesOut) as totalBytes
FROM internet_measurements
where json_extract_scalar(clientLocation, '$.city') != 'N/A'
```

```
GROUP BY
json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.subdivision'),
    json_extract_scalar(clientLocation, '$.country')
ORDER BY totalBytes desc
limit 100
```

Per ulteriori informazioni sull'utilizzo di Athena, consulta la Guida per l'utente di Amazon Athena.

Utilizzo dell'interfaccia di interrogazione Amazon CloudWatch Internet Monitor

Un'opzione per comprendere meglio il traffico Internet per la tua AWS applicazione consiste nell'utilizzare l'interfaccia di interrogazione di Amazon CloudWatch Internet Monitor. Per utilizzare l'interfaccia di interrogazione, crei una query con filtri di dati a tua scelta, quindi esegui la query per restituire un sottoinsieme dei dati di Monitor Internet. L'esplorazione dei dati restituiti dalla query può fornire informazioni sulle prestazioni dell'applicazione su Internet.

È possibile interrogare ed esplorare tutti i parametri acquisiti da Monitor Internet con il monitor, tra cui i punteggi di disponibilità e prestazioni, i byte trasferiti, i tempi di andata e ritorno e il time to first byte (TTFB).

Monitor Internet utilizza l'interfaccia di interrogazione per fornire i dati che è possibile esplorare nel pannello di controllo della console di Monitor Internet. Utilizzando le opzioni di ricerca nel pannello di controllo, nella scheda Esploratore storico o nella scheda Informazioni sul traffico, puoi interrogare e filtrare i dati Internet per l'applicazione.

Se desideri una maggiore flessibilità per esplorare e filtrare i tuoi dati rispetto a quella fornita dalla dashboard, puoi utilizzare tu stesso l'interfaccia di interrogazione, utilizzando le operazioni dell'API Internet Monitor con AWS Command Line Interface o con un AWS SDK. Questa sezione presenta i tipi di query che puoi utilizzare con l'interfaccia di interrogazione e i filtri che puoi specificare per creare un sottoinsieme di dati al fine di ottenere approfondimenti sul traffico Internet per la tua applicazione.

Argomenti

- Come utilizzare l'interfaccia di interrogazione
- Esempi di query
- Ottenimento dei risultati della query
- Risoluzione dei problemi

Come utilizzare l'interfaccia di interrogazione

Per creare una query con l'interfaccia di interrogazione, è possibile scegliere un tipo di query e quindi specificare i valori di filtro per restituire uno specifico sottoinsieme desiderato dei dati del file di log. È quindi possibile lavorare con il sottoinsieme di dati, filtrarlo e ordinarlo ulteriormente, creare report e così via.

Il processo di query avviene in questo modo:

1. Quando si esegue una query, Monitor Internet restituisce un `query ID` univoco per la query. Questa sezione descrive i tipi di query disponibili e le opzioni per filtrare i dati nelle query. Per capire come funziona, puoi anche consultare la sezione sugli esempi di query.
2. Specificate l'ID della query con il nome del monitor con l'operazione GetQueryResultsAPI per restituire i risultati dei dati per la query. Ogni tipo di query restituisce un set diverso di campi di dati. Per ulteriori informazioni, consulta la sezione Ottenimento dei risultati della query.

L'interfaccia di interrogazione fornisce i seguenti tre tipi di query. Ogni tipo di query restituisce un diverso set di informazioni sul traffico proveniente dai file di log, come illustrato.

- Misurazioni: fornisce il punteggio di disponibilità, il punteggio delle prestazioni, il traffico totale e i tempi di andata e ritorno, a intervalli di 5 minuti.
- Posizioni principali: fornisce il punteggio di disponibilità, il punteggio delle prestazioni, il traffico totale e le informazioni sul time-to-first byte (TTFB), per le principali combinazioni di località e ASN che stai monitorando, in base al volume di traffico.
- Dettagli sulle sedi principali: fornisce TTFB per Amazon CloudFront, la tua configurazione attuale e la configurazione Amazon EC2 con le migliori prestazioni, a intervalli di 1 ora.

Con ognuno di questi tipi di query è possibile filtrare ulteriormente i dati specificando uno o più dei seguenti criteri:

- AWS posizione: per la AWS posizione, puoi specificare CloudFront o Regione AWS, ad esempio, e così `us-east-2` via`us-west-2`.
- ASN: specifica un ASN, che in genere è un provider di servizi Internet (ISP).
- Posizione del client: per l'ubicazione, specifica una città, un'area metropolitana, una regione o un paese.

- Geo: specifica `geo` per alcune query. È necessario per le query che utilizzano il tipo di query `Top locations`, ma non è consentito per altri tipi di query. Per capire quando specificare `geo` per i parametri del filtro, consulta la sezione sugli esempi di query.

Gli operatori che è possibile utilizzare per filtrare i dati sono `EQUALS` e `NOT_EQUALS`. Per informazioni dettagliate sui parametri di filtraggio, consulta il funzionamento dell'FilterParameterAPI.

Per visualizzare i dettagli sulle operazioni dell'interfaccia di interrogazione, consulta le seguenti operazioni API nella Guida di riferimento dell'API di Amazon CloudWatch Internet Monitor:

- Per creare ed eseguire una query, consulta il funzionamento dell'StartQueryAPI.
- Per interrompere una query, consulta l'operazione StopQueryAPI.
- Per restituire i dati per una query che hai creato, consulta il funzionamento dell'GetQueryResultsAPI.
- Per recuperare lo stato di una query, consulta l'operazione GetQueryStatusAPI.

Esempi di query

Per creare una query da utilizzare per recuperare un set di dati filtrato dal file di registro del monitor, si utilizza l'StartQueryoperazione API. È possibile specificare un tipo di query e i parametri di filtro per la query. Quindi, quando si utilizza l'operazione API dell'interfaccia di interrogazione di Monitor Internet per ottenere i risultati delle query utilizzando la query, verrà recuperato il sottoinsieme di dati con cui si desidera lavorare.

Per comprendere come funzionano i tipi di query e i parametri di filtro, vediamo alcuni esempi.

Esempio 1

Supponiamo che tu voglia recuperare tutti i dati del file di log del monitor per un paese specifico, ad eccezione di una città. L'esempio seguente mostra i parametri di filtro per una query che è possibile creare con l'operazione `StartQuery` per questo scenario.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "MEASUREMENTS"
   FilterParameters: [
      {
```

```
        Field: "country",
        Operator: "EQUALS",
        Values: ["Germany"]
       },
       {
        Field: "city",
        Operator: "NOT_EQUALS",
        Values: ["Berlin"]
       },
    ]
}
```

Esempio 2

Come altro esempio, supponiamo che tu voglia visualizzare le posizioni principali per area metropolitana. È possibile utilizzare la seguente query di esempio per questo scenario.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["metro"]
      },
    ]
}
```

Esempio 3

Supponiamo ora che tu voglia vedere le migliori combinazioni di rete urbana nell'area metropolitana di Los Angeles. Per fare ciò, specifica `geo=city` e quindi imposta `metro` su Los Angeles. Ora, la query restituisce le principali reti urbane dell'area metropolitana di Los Angeles anziché le principali reti metropolitane in generale.

Ecco la query di esempio che puoi usare:

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
```

```
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["city"]
      },
      {
       Field: "metro",
       Operator: "EQUALS",
       Values: ["Los Angeles"]
      }
    ]
}
```

Esempio 4

Infine, supponiamo che tu voglia recuperare i dati TTFB per una regione specifica (ad esempio, uno stato degli Stati Uniti).

Di seguito è riportato un esempio di query per questo scenario:

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATION_DETAILS"
   FilterParameters: [
      {
       Field: "subdivision",
       Operator: "EQUALS",
       Values: ["California"]
      },
    ]
}
```

Ottenimento dei risultati della query

Dopo aver definito una query, è possibile restituire una serie di risultati con la query eseguendo un'altra operazione dell'API Internet Monitor,. GetQueryResults Quando esegui `GetQueryResults`, specifichi l'ID della query per la query che hai definito, insieme al nome del monitor. `GetQueryResults` recupera i dati per la query specificata in un set di risultati.

Quando esegui una query, assicurati che l'esecuzione sia terminata prima di utilizzare `GetQueryResults` per rivedere i risultati. È possibile determinare se la query è stata completata utilizzando l'operazione GetQueryStatusAPI. Quando il valore `Status` della query è `SUCCEEDED`, puoi procedere con la revisione dei risultati.

Una volta completata la query, è possibile utilizzare le informazioni seguenti per facilitare la revisione dei risultati. Ogni tipo di query utilizzato per creare una query include un set univoco di campi di dati dei file di log, come descritto nell'elenco seguente:

Misurazioni

Il tipo di query `measurements` restituisce i seguenti dati:

```
timestamp, availability, performance, bytes_in, bytes_out, rtt_p50,
rtt_p90, rtt_p95
```

Posizioni migliori

Il tipo di query `top locations` raggruppa i dati per posizione e fornisce la media dei dati nel periodo di tempo. I dati restituiti includono quanto segue:

```
aws_location, city, metro, subdivision, country, asn, availability,
performance, bytes_in, bytes_out, current_fbl, best_ec2,
best_ec2_region, best_cf_fbl
```

Tieni presente che `city`, `metro` e `subdivision` vengono restituiti solo se scegli quel tipo di posizione per il campo `geo`. Vengono restituiti i seguenti campi di posizione, a seconda del tipo di posizione specificato per `geo`:

```
city = city, metro, subdivision, country
metro = metro, subdivision, country
subdivision = subdivision, country
country = country
```

Dettagli sulle posizioni migliori

Il tipo di query `top locations details` restituisce i dati raggruppati ora per ora. La query restituisce i seguenti dati:

```
timestamp, current_service, current_fbl, best_ec2_fbl, best_ec2_region,
best_cf_fbl
```

Quando si esegue l'operazione API `GetQueryResults`, Monitor Internet restituisce quanto segue nella risposta:

- Un array di stringhe di dati che contiene i risultati restituiti dalla query. Le informazioni vengono restituite in array allineati con il campo `Fields`, anch'essi restituiti dalla chiamata API. Utilizzando il campo `Fields`, è possibile analizzare le informazioni dal repository `Data` e quindi filtrarle o ordinarle ulteriormente per i propri scopi.
- Un array di campi che elenca i campi per i quali la query ha restituito i dati (nella risposta del campo `Data`). Ogni elemento dell'array è una coppia nome-tipo di dati, ad esempio `availability_score-float`.

Risoluzione dei problemi

Se vengono restituiti errori quando utilizzi le operazioni API dell'interfaccia di interrogazione, verifica di disporre delle autorizzazioni necessarie per utilizzare Amazon CloudWatch Internet Monitor. In particolare, accertati di disporre delle seguenti autorizzazioni:

```
internetmonitor:StartQuery
internetmonitor:GetQueryStatus
internetmonitor:GetQueryResults
internetmonitor:StopQuery
```

Queste autorizzazioni sono incluse nella AWS Identity and Access Management politica consigliata per l'uso della dashboard di Internet Monitor nella console. Per ulteriori informazioni, consulta Autorizzazioni IAM per Amazon CloudWatch Internet Monitor.

Creazione di allarmi con Amazon CloudWatch Internet Monitor

Puoi creare CloudWatch allarmi Amazon in base ai parametri di Amazon CloudWatch Internet Monitor, proprio come puoi fare per altri parametri Amazon CloudWatch .

Ad esempio, puoi creare un allarme basato sul parametro `PerformanceScore` di Monitor Internet e configurarlo per inviare una notifica quando il parametro è inferiore a un valore scelto. Puoi configurare gli allarmi per i parametri di Internet Monitor seguendo le stesse linee guida degli altri parametri. CloudWatch

Di seguito sono riportati alcuni esempi di parametri di Monitor Internet per le quali potresti scegliere di creare un allarme:

- PerformanceScore
- AvailabilityScore
- RoundtripTime

Per visualizzare tutti i parametri disponibili per Monitor Internet, consulta. Utilizzo di CloudWatch metriche con Amazon CloudWatch Internet Monitor

La procedura seguente fornisce un esempio di impostazione di un allarme PerformanceScoreaccedendo alla metrica nella dashboard. CloudWatch Quindi, segui i CloudWatch passaggi standard per creare un allarme in base a una soglia scelta e impostare una notifica o scegliere altre opzioni.

Per creare un allarme per PerformanceScorein CloudWatch Metrics

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Seleziona Parametri, quindi scegli Tutti i parametri.
3. Filtra per Monitor Internet scegliendo `AWS/InternetMonitor`.
4. Scegli MeasurementSource, MonitorName.
5. Nell'elenco, selezionate PerformanceScore.
6. Nella GraphedMetricsscheda, in Azioni, scegli l'icona a forma di campana per creare un allarme basato su una soglia statica.

Ora segui i CloudWatch passaggi standard per scegliere le opzioni per l'allarme. Ad esempio, puoi scegliere di ricevere una notifica tramite un messaggio Amazon SNS se PerformanceScoreè inferiore a un numero di soglia specifico. In alternativa, o in aggiunta, puoi aggiungere l'allarme a un pannello di controllo.

Ricorda:

- I parametri di Monitor Internet vengono generalmente calcolati e pubblicati entro 20 minuti.
- Quando crei un allarme basato sui parametri di Monitor Internet, assicurati di tenere conto del breve ritardo prima della pubblicazione quando imposti il periodo di ricerca posticipata di un allarme. Ti consigliamo di configurare i Periodi di valutazione con un periodo di ricerca posticipata di almeno 25 minuti.

Per ulteriori informazioni sull'uso degli CloudWatch allarmi con Internet Monitor, consulta il seguente post di blog: Using Amazon CloudWatch Internet Monitor for Enhanced Internet Observability.

Per ulteriori informazioni sulle opzioni disponibili quando crei un CloudWatch allarme, consulta. Crea un CloudWatch allarme basato su una soglia statica

Utilizzo di Amazon CloudWatch Internet Monitor con Amazon EventBridge

Gli eventi sanitari creati da Amazon CloudWatch Internet Monitor per problemi di rete vengono pubblicati con Amazon EventBridge, in modo che tu possa inviare notifiche su qualsiasi peggioramento dell'esperienza degli utenti finali con la tua applicazione.

Per EventBridge utilizzarlo per lavorare con gli eventi sanitari di Internet Monitor, segui le istruzioni qui.

Per impostare una regola per Internet Monitor in EventBridge

1. Nel AWS Management Console, in EventBridge, scegli Regole, quindi inserisci un nome e una descrizione. Crea la regola sul router di eventi Default (Predefinito).
2. Nel passaggio 2, seleziona Altro come origine dell'evento, quindi in Modello di eventi, cerca l'origine seguente.

```
{
  "source": ["aws.internetmonitor"]
}
```

3. Nel passaggio 3, per la destinazione, seleziona AWS Service and CloudWatch Logs Group, quindi seleziona un gruppo di log esistente o creane uno nuovo.
4. Aggiungi i tag desiderati, quindi crea la regola. Questo dovrebbe popolare il gruppo di CloudWatch log selezionato con gli eventi di. EventBridge

Per ulteriori informazioni su come EventBridge le regole funzionano con i pattern di eventi, consulta Amazon EventBridge event pattern nella Amazon EventBridge User Guide.

Risolvi gli errori di accesso a CloudWatch log e metriche

Per supportare alcune funzionalità, Amazon CloudWatch Internet Monitor deve interagire con determinate CloudWatch risorse Amazon, inclusi log e metriche. Se Internet Monitor non riesce ad

accedere alle CloudWatch risorse a cui richiede l'accesso, Internet Monitor imposta un codice di stato `FAULT_ACCESS_CLOUDWATCH` per il monitor.

Esistono diversi motivi per cui il monitor potrebbe presentare questo stato `FAULT_ACCESS_CLOUDWATCH`. Nelle sezioni seguenti sono elencate le possibili cause di questi errori e le procedure di risoluzione dei problemi suggerite.

Internet Monitor non è riuscito ad accedere ai CloudWatch log del tuo account

Internet Monitor pubblica i log di diagnostica relativi al traffico delle applicazioni controllato dal monitor. Pubblica questi registri in gruppi di log in CloudWatch Logs nella seguente posizione:. `/aws/internet-monitor/monitor_name/[byCity|byMetro|bySubdivision|byCountry]` Internet Monitor non è riuscito ad accedere a questi gruppi di log.

Stati di errore e possibili soluzioni:

- PutLogEvents errore di limitazione: il servizio Internet Monitor potrebbe essere stato limitato quando ha cercato di pubblicare i log del monitor su. CloudWatch Controlla i limiti di throttling per il tuo account e, se necessario, richiedi un aumento del limite.
- Gruppo di registro non trovato: disattiva e riattiva il monitor. L'attivazione di un monitor riavvia la creazione del gruppo di log, che potrebbe risolvere il problema.
- PutLogEvents errore di accesso negato: contatta l' AWS assistenza.
- PutLogEvents errore sconosciuto o generico: contatta l' AWS assistenza per ricevere assistenza.

Internet Monitor non è riuscito ad accedere alle CloudWatch metriche del tuo account

Internet Monitor fornisce CloudWatch metriche specifiche sul traffico delle applicazioni monitorato da un monitor. Si è verificato un errore quando Internet Monitor ha cercato di fornire queste metriche a. CloudWatch

Stati di errore e possibili soluzioni:

- PutMetricData errore di limitazione: il servizio Internet Monitor potrebbe essere stato limitato quando ha cercato di pubblicare le metriche del monitor su. CloudWatch Controlla i limiti di throttling per il tuo account e, se necessario, richiedi un aumento del limite.
- PutMetricData errore di accesso negato: contatta l'assistenza per ricevere assistenza. AWS
- PutMetricData errore sconosciuto o generico: contatta l' AWS assistenza per ricevere assistenza.

Protezione e riservatezza dei dati con Amazon CloudWatch Internet Monitor

Il modello di responsabilità AWS condivisa si applica alla protezione e alla privacy dei dati in Amazon CloudWatch Internet Monitor. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutto il AWS cloud. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Per ulteriori informazioni sulla privacy dei dati, consulta Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post sul blog The AWS Shared Responsibility Model e sul GDPR sul AWS Security Blog. Per ulteriori risorse sulla conformità ai requisiti GDPR, consulta la sezione Centro generale sulla protezione dei dati (GDPR).

Consigliamo vivamente di non inserire mai informazioni identificative sensibili, ad esempio i numeri di account dei clienti, indirizzi email o altre informazioni personali in campi a formato libero. Tutti i dati che inserisci in Amazon CloudWatch Internet Monitor o in altri servizi potrebbero essere inclusi nei log di diagnostica.

Identity and Access Management per Amazon CloudWatch Internet Monitor

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (chi ha effettuato l'accesso) e autorizzato (chi dispone di autorizzazioni) a utilizzare le risorse di Monitor Internet. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Important

Modifiche delle risorse di Monitor Internet il 24 febbraio 2023
Se hai creato policy IAM che includevano risorse Monitor Internet prima del 24 febbraio 2023, tieni presente le seguenti modifiche alle risorse e ai tipi di risorse di Monitor Internet.

- HealthEventsla risorsa è stata rinominata in. HealthEvent
- I formati ARN e Regex per la HealthEventrisorsa sono stati aggiornati.
- I formati ARN e Regex per la risorsa Monitor sono stati aggiornati.
- Le autorizzazioni a livello di risorsa per l'GetHealthEventazione sono ora supportate solo sul tipo di risorsa. HealthEvent Non sono supportate nella risorsa Monitor.
- TagResourceUntagResource, e ListTagsForResourceper il tipo di risorsa Monitor sono state aggiornate in modo da renderle obbligatorie.

Per ulteriori informazioni sulle azioni, le risorse e le chiavi di condizione che puoi specificare nelle politiche per gestire l'accesso alle AWS risorse in Internet Monitor, consulta Azioni, risorse e chiavi di condizione per Amazon CloudWatch Internet Monitor.

Indice

Come funziona Amazon CloudWatch Internet Monitor con IAM

Prima di utilizzare IAM per gestire l'accesso a Monitor Internet, scopri quali funzionalità di IAM sono disponibili per l'uso.

Per visualizzare le tabelle che mostrano una visione di alto livello simile su come AWS i servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Funzionalità IAM che puoi utilizzare con Amazon CloudWatch Internet Monitor

Funzionalità IAM	Supporto per Monitor Internet
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
Liste di controllo degli accessi (ACL)	No

Funzionalità IAM	Supporto per Monitor Internet
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	No
Ruoli collegati al servizio	Sì

Policy basate sull'identità per Monitor Internet

Supporta le policy basate su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Policy basate sulle risorse all'interno di Monitor Internet

Supporta le policy basate su risorse	No

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica.

Operazioni delle policy per Monitor Internet

Supporta le operazioni di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni di Internet Monitor, consulta Azioni definite da Amazon CloudWatch Internet Monitor nel Service Authorization Reference.

Le operazioni delle policy in Monitor Internet utilizzano il seguente prefisso prima dell'operazione:

```
internetmonitor
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "internetmonitor:action1",
      "internetmonitor:action2"
         ]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Describe`, includi la seguente azione:

```
"Action": "internetmonitor:Describe*"
```

Risorse delle policy per Monitor Internet

Supporta le risorse di policy	Sì

In Informazioni di riferimento sull'autorizzazione del servizio, è possibile visualizzare le seguenti informazioni relative a Monitor Internet:

- Per visualizzare un elenco dei tipi di risorse di Internet Monitor e dei relativi ARN, consulta Risorse definite da Amazon CloudWatch Internet Monitor.
- Per informazioni sulle azioni che puoi specificare con l'ARN di ogni risorsa, consulta Azioni definite da Amazon CloudWatch Internet Monitor.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Chiavi delle condizioni delle policy per Monitor Internet

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni

condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND` logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di Internet Monitor, consulta Condition keys for Amazon CloudWatch Internet Monitor nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta Azioni definite da Amazon CloudWatch Internet Monitor.

ACL in Monitor Internet

Supporta le ACL	No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con Monitor Internet

Supporta ABAC (tag nelle policy)	Parziale

Monitor Internet supporta parzialmente i tag nelle policy. Supporta il tagging per una risorsa: i monitor.

Per utilizzare i tag con Internet Monitor, usa AWS Command Line Interface o un AWS SDK. L'etichettatura per Internet Monitor non è supportata con. AWS Management Console

Per ulteriori informazioni sull'uso dei tag nelle policy in generale, consulta le seguenti informazioni.

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Che cos'è ABAC? nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con Monitor Internet

Supporta le credenziali temporanee	Sì

Alcuni Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente

e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Cambio di un ruolo (console) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni del principale tra servizi per Monitor Internet

Supporta l'inoltro delle sessioni di accesso (FAS)	Sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

Ruoli di servizio per Monitor Internet

Supporta i ruoli di servizio	No

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.

Ruolo collegato ai servizi per Monitor Internet

Supporta i ruoli collegati ai servizi	Sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni sul ruolo collegato al servizio per Monitor Internet, consulta Ruolo collegato ai servizi per Amazon Internet Monitor CloudWatch .

Per i dettagli sulla creazione o la gestione di ruoli collegati ai servizi in generale in AWS, consulta AWS Servizi che funzionano con IAM. Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

AWS politiche gestite per Amazon CloudWatch Internet Monitor

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

AWS politica gestita: CloudWatchInternetMonitorServiceRolePolicy

Questa policy è associata al ruolo collegato al servizio denominato AWSServiceRoleForInternetMonitorper consentire a Internet Monitor di accedere alle risorse del tuo account, come le risorse Amazon Virtual Private Cloud o Network Load Balancers, in modo che tu possa selezionarle quando crei un monitor. Per ulteriori informazioni, consulta Ruolo collegato ai servizi per Amazon Internet Monitor CloudWatch .

Autorizzazioni IAM per Amazon CloudWatch Internet Monitor

Per accedere alle azioni per lavorare con monitor e dati in Amazon CloudWatch Internet Monitor, gli utenti devono disporre delle autorizzazioni corrette.

Per ulteriori informazioni sulla sicurezza in Amazon CloudWatch, consultaGestione delle identità e degli accessi per Amazon CloudWatch.

Autorizzazioni per l'accesso in sola lettura in Amazon Internet Monitor CloudWatch

Per accedere alle azioni di sola lettura per lavorare con monitor e dati in Amazon CloudWatch Internet Monitor, gli utenti devono accedere come utente o ruolo con le seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "internetmonitor:Get*",
                "internetmonitor:List*",
                "internetmonitor:StartQuery",
                "internetmonitor:StopQuery",
                "logs:DescribeLogGroups",
                "logs:GetQueryResults",
                "logs:StartQuery",
                "logs:StopQuery"
                ],
            "Resource": "*"
        }
    ]
}
```

Autorizzazioni per l'accesso completo in Amazon CloudWatch Internet Monitor

Per creare un monitor in Amazon CloudWatch Internet Monitor e avere pieno accesso alle azioni per lavorare con monitor e dati in Internet Monitor, gli utenti devono accedere con un utente o un ruolo con le seguenti autorizzazioni:

- Autorizzazioni per creare un ruolo collegato ai servizi associato a Monitor Internet. Per ulteriori informazioni, consulta Ruolo collegato ai servizi per Amazon Internet Monitor CloudWatch .

- Autorizzazioni per operazioni che consentono l'accesso completo all'utilizzo di monitor e dati in Monitor Internet.

> **Note**
>
> Se crei una policy di autorizzazione basata su identità più restrittiva, gli utenti con tale policy potrebbero non avere accesso completo alla creazione e all'utilizzo di monitor e dati in Monitor Internet.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "internetmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "internetmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor"
        },
        {
```

```
            "Action": [
                "ec2:DescribeVpcs",
                "elasticloadbalancing:DescribeLoadBalancers",
                "workspaces:DescribeWorkspaceDirectories",
                "cloudfront:GetDistribution"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Ruolo collegato ai servizi per Amazon Internet Monitor CloudWatch

Amazon CloudWatch Internet Monitor utilizza un ruolo collegato al servizio AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente a Monitor Internet. Il ruolo collegato al servizio è predefinito da Internet Monitor e include tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Monitor Internet definisce le autorizzazioni del ruolo collegato al servizio e, salvo diversamente definito, solo Monitor Internet potrà assumere tale ruolo. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Puoi eliminare il ruolo solo dopo aver rimosso le risorse correlate. Questa limitazione protegge le risorse di Monitor Internet poiché impedisce la rimozione involontaria delle autorizzazioni per accedere alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Yes (Sì) nella colonna Service-linked roles (Ruoli collegati ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per Monitor Internet

Internet Monitor utilizza il ruolo collegato al servizio denominato. AWSServiceRoleForInternetMonitor Questo ruolo consente a Internet Monitor di accedere alle risorse del tuo account, come risorse Amazon Virtual Private Cloud, CloudFront distribuzioni Amazon, WorkSpaces directory Amazon e Network Load Balancers, in modo da poterle selezionare quando crei un monitor.

Questo ruolo collegato al servizio utilizza la policy gestita.
`CloudWatchInternetMonitorServiceRolePolicy`

Il ruolo AWSServiceRoleForInternetMonitorcollegato al servizio prevede che il ruolo venga assunto dal seguente servizio:

- `internetmonitor.amazonaws.com`

Per visualizzare le autorizzazioni per questa politica, consulta il Managed Policy CloudWatchInternetMonitorServiceRolePolicyReference.AWS

Creazione di un ruolo collegato ai servizi per Monitor Internet

Non è necessario creare manualmente un ruolo collegato ai servizi per Monitor Internet. La prima volta che crei un monitor, Internet Monitor lo crea AWSServiceRoleForInternetMonitorper te.

Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Modifica di un ruolo collegato ai servizi per Monitor Internet

Dopo che Monitor Internet ha creato un ruolo collegato ai servizi, non potrai modificarne il nome perché diverse entità potrebbero farvi riferimento. Puoi tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Monitor Internet

Se non è più necessario utilizzare una caratteristica o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, devi effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare manualmente.

Dopo aver rimosso le risorse dai monitor in Internet Monitor e aver eliminato i monitor, puoi eliminare il ruolo collegato al servizio. AWSServiceRoleForInternetMonitor

> **Note**
>
> Se il servizio Monitor Internet utilizza tale ruolo mentre tenti di eliminarlo, è possibile che l'operazione non abbia esito positivo. In questo caso, attendi alcuni minuti, quindi riprova.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Usa la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al servizio. AWSServiceRoleForInternetMonitor Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Aggiornamenti del ruolo collegato al servizio per Monitor Internet

Per gli aggiornamenti alla AWSServiceRoleForInternetMonitorpolitica AWS gestita per il ruolo collegato al servizio Internet Monitor, vedi CloudWatch Aggiornamenti alle politiche gestite. AWS Per ricevere avvisi automatici sulle modifiche alle policy gestite in CloudWatch, iscriviti al feed RSS nella pagina della cronologia dei CloudWatch documenti.

Quote in Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor ha le seguenti quote.

Risorsa	Quota predefinita
Monitoraggi per regione	50
Risorse per monitoraggio	50
Giorni in cui vengono mantenuti gli eventi di stato di Monitor Internet	400

Utilizzo di Amazon CloudWatch Network Monitor

Amazon CloudWatch Network Monitor offre visibilità sulle prestazioni della rete, connettendo le applicazioni AWS ospitate alle destinazioni locali e consente di identificare l'origine di qualsiasi peggioramento delle prestazioni di rete in pochi minuti. Monitor di rete è completamente gestito da AWS. Pertanto, non sarà necessario installare altri agenti per monitorare le prestazioni della rete. Puoi visualizzare rapidamente la perdita di pacchetti e la latenza delle connessioni di rete ibride, impostare avvisi e soglie e quindi intervenire per migliorare l'esperienza di rete degli utenti finali.

Monitor di rete è destinato agli operatori di rete e agli sviluppatori di applicazioni che desiderano informazioni in tempo reale sulle prestazioni della rete.

Funzionalità principali

- Usa Monitor di rete per il benchmark del tuo ambiente di rete ibrido in evoluzione con parametri di latenza e perdita di pacchetti in tempo reale.
- Quando ti connetti utilizzando AWS Direct Connect, Network Monitor diagnostica rapidamente il degrado della rete scrivendo il Network AWS Health Indicator sul tuo CloudWatch account. Questo parametro fornisce un punteggio probabilistico per determinare se il peggioramento della rete si è verificato all'interno di AWS.
- Monitor di rete offre un monitoraggio semplice con un approccio basato su agenti completamente gestito, il che significa che non è necessario installare agenti né sui VPC né on-premise. Devi solo specificare una sottorete VPC e un indirizzo IP on-premise per iniziare.
- Network Monitor pubblica le metriche su Metrics. CloudWatch Puoi creare dashboard per visualizzare i tuoi parametri e creare soglie e allarmi utilizzabili in base ai parametri specifici della tua applicazione.

Per ulteriori dettagli, consulta the section called “Funzionamento di Monitor di rete”.

Terminologia e componenti di Monitor di rete

- Monitor: un monitor include le risorse per cui desideri visualizzare le misurazioni delle prestazioni e della disponibilità della rete e su cui desideri ricevere avvisi relativi agli eventi di integrità. Quando si crea un monitor per un'applicazione, si aggiunge una risorsa AWS ospitata come origine di rete. Network Monitor crea quindi un elenco di tutte le possibili sonde tra le risorse AWS ospitate e gli indirizzi IP di destinazione.
- Sonde: una sonda è il traffico inviato dalla risorsa AWS ospitata all'indirizzo IP di destinazione locale. Le metriche di Network Monitor vengono scritte nell' CloudWatch account per ogni sonda configurata in un monitor.
- AWS fonte di rete: questa è la AWS fonte di origine di una sonda di monitoraggio di rete, che sarà una sottorete in uno qualsiasi dei tuoi VPC.
- Destinazione: questa è la destinazione dell'origine di rete nella tua rete on-premise AWS . La destinazione è una combinazione di indirizzi IP on-premise, protocolli di rete, porte e dimensioni dei pacchetti di rete. Sono supportati sia indirizzi IPv4 che IPv6.

Limitazioni e requisiti di Monitor di rete

- Monitor di rete supporta un massimo di quattro indirizzi IP di destinazione e fino a 24 sonde per monitor.
- Puoi avere fino a 100 monitor per account per ogni regione.
- Le sottoreti di monitoraggio devono appartenere allo stesso account del monitor.
- Network Monitor non fornisce il failover automatico della rete in caso di problemi di AWS rete.
- È previsto un costo per ogni sonda creata. Per informazioni sui prezzi, consulta the section called “Prezzi”.

Come funziona Amazon CloudWatch Network Monitor

Monitor di rete semplifica il monitoraggio fornendo una soluzione completamente gestita e senza agenti. Quando crei un monitor nella tua risorsa AWS ospitata, AWS crea e gestisce tutta l'infrastruttura in background per eseguire misurazioni del tempo di andata e ritorno e della perdita di pacchetti. Di conseguenza, è possibile scalare rapidamente il monitoraggio senza dover installare o disinstallare alcun agente all'interno dell'infrastruttura. AWS

Network Monitor concentra il monitoraggio sui percorsi seguiti dai flussi provenienti dalle risorse AWS ospitate anziché sul monitoraggio generale di tutti i flussi provenienti dall'utente. Regione AWS Se i carichi di lavoro sono distribuiti su più zone di disponibilità (AZ), Monitor di rete può monitorare i percorsi da ciascuna delle tue sottoreti private.

Monitor di rete pubblica i parametri relativi al tempo di andata e ritorno e alla perdita di pacchetti sull'account Amazon CloudWatch in base all'intervallo di aggregazione impostato al momento della creazione di un monitor. Puoi anche impostare soglie individuali di latenza e perdita di pacchetti per ogni monitor utilizzato. CloudWatch Ad esempio, potresti creare un allarme che ti avvisi se la media della perdita di pacchetti è superiore a una soglia statica dello 0,1% per un carico di lavoro sensibile alla perdita di pacchetti. È inoltre possibile utilizzare il rilevamento delle CloudWatch anomalie per segnalare la perdita di pacchetti o le metriche di latenza al di fuori degli intervalli desiderati.

Misurazioni della disponibilità e delle prestazioni

Network Monitor invia sonde attive periodiche dalla risorsa alle AWS destinazioni locali. Quando crei un monitor, specifica quanto segue:

- L'intervallo di aggregazione. Il tempo, in secondi, di CloudWatch ricezione dei risultati misurati. Questo periodo di tempo sarà ogni 30 o 60 secondi. Il periodo di aggregazione scelto per il monitor si applica a tutte le sonde del monitor.
- Il protocollo della sonda. Ogni sonda aggiunta a un monitor deve utilizzare i protocolli ICMP (Internet Control Message Protocol) o TCP (Transmission Control Protocol). Per ulteriori dettagli, consulta the section called “Protocolli di comunicazione”.
- La dimensione del pacchetto. La dimensione, in byte, di ogni pacchetto trasmesso tra la risorsa AWS ospitata e la destinazione su una singola sonda. Ogni sonda di un monitor può avere una propria dimensione di pacchetto.

Per i parametri,

- Il parametro del tempo di andata e ritorno, misurato in millisecondi, misura e registra una misura delle prestazioni e registra il tempo impiegato dalla sonda per essere trasmessa all'indirizzo IP di destinazione e per la ricezione della risposta associata.
- Il parametro della perdita di pacchetti misura la percentuale di pacchetti totali inviati e registra il numero di sonde trasmesse che non hanno ricevuto una risposta associata, che implica che tali pacchetti sono stati effettivamente persi lungo il percorso di rete.

Protocolli di comunicazione supportati

Le sonde basate su ICMP trasportano le richieste di eco ICMP dalle risorse AWS ospitate all'indirizzo di destinazione e si aspettano una risposta echo ICMP dall'indirizzo di destinazione. Monitor di rete utilizza le informazioni sui messaggi di richiesta e risposta echo ICMP per calcolare il tempo di andata e ritorno e i parametri di perdita di pacchetti.

Le sonde basate su TCP trasportano i pacchetti TCP SYN dalle risorse AWS ospitate all'indirizzo e alla porta di destinazione e si aspettano un pacchetto TCP SYN+ACK o RST dall'indirizzo e dalla porta di destinazione. Monitor di rete utilizza le informazioni sui messaggi TCP SYN e TCP SYN +ACK o RST per calcolare il tempo di andata e ritorno e i parametri di perdita di pacchetti. Inoltre, Monitor di rete cambia periodicamente le porte TCP di origine per aumentare la copertura di rete, il che può quindi aumentare la probabilità di rilevare la perdita di pacchetti.

AWS Indicatore di salute della rete

Network Monitor pubblica un parametro di indicatore di integrità della rete (Network Health Indicator, NHI) , che fornisce informazioni sulle prestazioni e sulla disponibilità della rete per le destinazioni

collegate tramite AWS Direct Connect. La metrica è una misura statistica dello stato del percorso di rete AWS controllato dalla risorsa AWS ospitata, dove viene distribuito il monitor, alla posizione Direct Connect.

Monitor di rete utilizza il rilevamento delle anomalie per calcolare i cali di disponibilità o il peggioramento delle prestazioni lungo i percorsi di rete.

> **Note**
>
> Ogni volta che crei un nuovo monitor, aggiungi una sonda o riattivi una sonda, l'NHI per quel monitor subirà un ritardo di alcune ore per consentire AWS la raccolta dei dati per eseguire il rilevamento delle anomalie.

Per fornire il parametro di integrità NHI, Monitor di rete applica la correlazione statistica tra set di dati AWS campione, nonché ai parametri di latenza e perdita di pacchetti di andata e ritorno per traffico che simula il tuo percorso di rete. Il parametro può essere una delle due variabili: 1 o 0. Il valore 1 indica che Network Monitor ha rilevato un deterioramento della rete all'interno del percorso di rete controllato. AWS Il valore 0 indica che Monitor di rete non ha rilevato alcun peggioramento della rete all'interno del percorso di rete. Ciò consente di risolvere i problemi di rete più rapidamente. È possibile impostare avvisi sul parametro NHI per ricevere informazioni sui problemi in corso nei percorsi di rete.

Supporto per gli indirizzi IPv4 e IPv6

Monitor di rete fornisce parametri di disponibilità e prestazioni su reti IPv4 o IPv6 e può monitorare indirizzi IPv4 o IPv6 da VPC dual-stack. Monitor di rete non consente la configurazione di destinazioni IPv4 e IPv6 all'interno dello stesso monitor, ma è possibile creare destinazioni separate solo per IPv4 e solo per IPv6.

Disponibilità nelle regioni

Network Monitor è attualmente disponibile nelle seguenti Regioni AWS versioni:

Regione	
Asia Pacifico	ap-east-1

Regione	
(Hong Kong)	
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada occidentale (Calgary)	ca-west-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1

Regione	
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Medio Oriente (Bahrein)	me-south-1
Sud America (São Paulo)	sa-east-1
US East (N. Virginia)	us-east-1
Stati Uniti orientali (Ohio)	us-east-2
Stati Uniti occidentali (California settentrionale)	us-west-1
US West (Oregon)	us-west-2

Creazione di un monitor di rete

I passaggi seguenti descrivono la creazione di un monitor e l'aggiunta delle sonde richieste. Per le sonde, sceglierai la sottorete di origine e fino a quattro indirizzi IP di destinazione per un massimo di 24 sonde per monitor. Puoi creare un monitor utilizzando la console Amazon CloudWatch oppure tramite la riga di comando o l'API.

Argomenti

Creazione di un monitor utilizzando la console

I seguenti passaggi descrivono la creazione di un monitor utilizzando la console Amazon CloudWatch . Sceglierai le sottoreti di origine e quindi aggiungerai fino a quattro destinazioni per creare fino a 24 sonde per monitor. Puoi creare un monitor utilizzando la console Amazon CloudWatch oppure tramite la riga di comando o l'SDK.

> **Important**
>
> Questi passaggi sono pensati per essere completati tutti in una volta. Non sarà possibile salvare un'operazione in corso per riprenderla in un secondo momento.

Definizione dei dettagli del monitor

Il primo passo per creare un monitor è definire i dettagli di base. Questo include l'assegnazione di un nome al monitor e la definizione del periodo di aggregazione. È anche possibile aggiungere i tag al monitor.

Per definire i dettagli del monitor

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.
2. Scegli Crea monitor.
3. In Nome monitor, inserisci il nome che desideri utilizzare per questo monitor.
4. Per il periodo di aggregazione, scegli la frequenza a cui desideri inviare le metriche. CloudWatch I periodi di aggregazione disponibili sono:

- 30 secondi
- 60 secondi

> **Note**
>
> Un periodo di aggregazione più breve consente di rilevare più rapidamente i problemi di rete; tuttavia, il periodo di aggregazione scelto può influire sulla struttura di fatturazione. Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

5. (Facoltativo) Nella sezione Tag, aggiungi le coppie Chiave e Valore per identificare più facilmente la risorsa, per poter cercare o filtrare informazioni specifiche.

   1. Scegli Aggiungi nuovo tag.
   2. Inserisci il nome della chiave e il valore associato.
   3. Per aggiungere un tag, scegli Aggiungi un nuovo tag.

      Puoi aggiungere più tag scegliendo Aggiungi nuovo tag oppure puoi rimuovere qualsiasi tag scegliendo Rimuovi.
   4. Se desideri associare i tag al monitor, tieni selezionata l'opzione Aggiungi tag alle sonde create dal monitor. Questa operazione aggiunge i tag alle sonde del monitor, il che può essere utile se utilizzi l'autenticazione o la misurazione basata su tag.

6. Seleziona Successivo per the section called “Seleziona l'origine e la destinazione”.

Seleziona l'origine e la destinazione

Un monitor di rete utilizza una AWS fonte per i VPC e le sottoreti associate nelle regioni in cui opera la rete. Una destinazione di monitor è la combinazione di indirizzi IP on-premise, protocolli di rete, porte e dimensioni dei pacchetti di rete.

La combinazione di origine e destinazione viene definita sonda. È possibile avere fino a quattro sonde per sottorete e fino a un totale di 24 sonde per monitor.

> **Important**
>
> Questi passaggi sono pensati per essere completati tutti in una volta. Non sarà possibile salvare un'operazione in corso per riprenderla in un secondo momento.

Per scegliere l'origine e la destinazione

1. Nell'origine di rete AWS, seleziona una o più sottoreti da includere nel monitor. Puoi scegliere un singolo VPC, che quindi sceglierà tutte le sottoreti all'interno di quel VPC, oppure puoi scegliere sottoreti specifiche. I VPC e le sottoreti scelti costituiranno l'origine del monitor di rete.
2. Per Destinazione 1, inserisci l'indirizzo IP di destinazione della rete on-premise. Sono supportati sia indirizzi IPv4 che IPv6.
3. Scegli Impostazioni avanzate.
4. Scegli il protocollo di rete per questa destinazione gestita dal cliente. Può essere:
   - ICMP
   - TCP
5. Se il protocollo è TCP, inserisci le seguenti informazioni. Altrimenti, passa alla fase successiva:
   1. Inserisci la porta utilizzata dalla rete per la connessione. La porta deve essere un numero compreso tra 1 e 65535.
   2. Inserisci la dimensione del pacchetto. Questa è la dimensione, in byte, di ogni pacchetto inviato sulla sonda tra l'origine e la destinazione. La dimensione del pacchetto deve essere un numero compreso tra 56 e 8500.
6. Scegli Aggiungi destinazione per aggiungere un'altra destinazione on-premise a questo monitor. Ripeti queste fasi per ogni destinazione che desideri aggiungere.
7. Al termine, scegli Successivo per confermare le sonde.

Conferma delle sonde

La conferma delle sonde consente di esaminare la combinazione delle sonde di rete per il monitor. Questa pagina mostra tutte le possibili combinazioni delle origini e delle destinazioni che hai scelto. Ad esempio, se hai sei sottoreti di origine e quattro IP di destinazione, avrai un totale di 24 possibili combinazioni di sonde.

Important

- Questi passaggi sono pensati per essere completati tutti in una volta. Non sarà possibile salvare un'operazione in corso per riprenderla in un secondo momento.
- La pagina Conferma sonde non indica se una sonda è valida. Pertanto, consigliamo di esaminare attentamente la pagina ed eliminare eventuali sonde non valide. Se non rimuovi le sonde non valide, potrebbero esserti addebitati dei costi.

Per confermare le sonde del monitor

1. Prerequisito: the section called “Seleziona l'origine e la destinazione”.
2. Nella pagina Conferma sonde, esamina l'elenco delle combinazioni di origine e destinazione.
3. Seleziona una o più sonde da rimuovere dal monitor, quindi scegli Rimuovi.

   Note

   Non ti viene richiesto di confermare l'eliminazione. Dopo aver eliminato una sonda, è necessario riconfigurarla. È possibile aggiungere nuovamente una sonda a un monitor dalla sezione Monitor di rete nella pagina Monitor di rete. Per ulteriori informazioni, consulta the section called “Aggiunta di una sonda a un monitor”.
4. Scegli Successivo per controllare i dettagli del monitor prima di crearlo.

Rivedi e crea

L'ultimo passaggio per creare un monitor e le sonde consiste nell'esaminare i dettagli sia del monitor che delle sonde. Puoi modificare qualsiasi informazione a questo punto. Una volta terminato il controllo e la creazione del monitor, sarà avviato il monitoraggio dei parametri e inizierai a ricevere i costi per eventuali sonde.

Important

- Questo passaggio deve essere completato tutto in una volta durante la creazione di un monitor e una sonda. Non sarà possibile salvare un'operazione in corso per riprenderla in un secondo momento.

- Se scegli di modificare una sezione, dovrai procedere alla creazione del monitor dal punto in cui stai effettuando la modifica. Tuttavia, non sarà necessario ricreare i passaggi successivi. Queste pagine mantengono le informazioni precedentemente inserite.

Per rivedere e creare un monitor

1. Nella pagina Rivedi e crea sonde, scegli Modifica per la sezione da modificare.
2. Apporta eventuali modifiche nella sezione.
3. Seleziona Successivo.
4. Effettua una delle seguenti operazioni:
   - Apporta le modifiche che desideri su altre pagine del monitor e seleziona Successivo finché non torni alla pagina Rivedi e crea.
   - Se nessun'altra pagina necessita di modifiche, scegli Successivo finché non torni alla pagina Rivedi e crea.
5. Scegli Crea monitor.

   La pagina Monitor di rete mostra lo stato attuale della creazione del monitor nella sezione Monitor di rete. Durante la creazione del monitor, lo stato è In attesa. Quando lo Stato diventa Attivo, puoi accedere alla dashboard di monitoraggio per visualizzare le metriche. CloudWatch

   Per ulteriori informazioni su come usare il pannello di controllo del monitoraggio, consulta the section called “Pannelli di controllo di Monitor di rete”.

> **Note**
>
> Potrebbero essere necessari diversi minuti prima che il monitor di rete appena aggiunto inizi a raccogliere i parametri di rete.

Creazione di un Monitor di rete utilizzando l'API o la riga di comando

Utilizza la riga di comando o l'API per visualizzare e creare un monitor di rete.

Per creare un monitor di rete utilizzando l'API o la riga di comando

1. Crea un monitor di rete usando create-monitor.
2. Crea una sonda di monitor di rete usando create-probe.

Utilizzo dei monitor e delle sonde di Monitor di rete

È possibile eseguire una delle seguenti attività con i monitor e le sonde, utilizzando la console Amazon CloudWatch o utilizzando la riga di comando o l'API.

Argomenti:

- Modifica di un monitor
- Eliminazione di un monitor
- Attivazione o disattivazione di una sonda
- Aggiunta di una sonda a un monitor
- Modifica di una sonda
- Eliminazione di una sonda
- Aggiunta o rimozione dei tag alle risorse utilizzando l'API o la riga di comando

Modifica di un monitor

Puoi modificare qualsiasi informazione di un Monitor di rete, ad esempio puoi rinominarlo, impostare un nuovo periodo di aggregazione o aggiungere o rimuovere i tag. La modifica delle informazioni di un monitor non modifica le sonde associate. Puoi modificare un monitor utilizzando la console Amazon CloudWatch oppure tramite la riga di comando o l'API.

Modifica di un monitor tramite la console

Usa la CloudWatch console per modificare un monitor.

Per modificare un monitor tramite la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.
2. Nella sezione Monitor di rete, scegli il monitor che desideri modificare.
3. Nella pagina del pannello di controllo del monitor, seleziona Modifica.

4. Inserisci il nuovo nome del monitor in Nome del monitor.
5. Per il periodo di aggregazione, scegli la frequenza a cui desideri inviare le metriche. CloudWatch I periodi validi sono:

   - 30 secondi
   - 60 secondi

   > **Note**
   >
   > Un periodo di aggregazione più breve consente di rilevare più rapidamente i problemi di rete; tuttavia, il periodo di aggregazione scelto può influire sulla struttura di fatturazione. Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

6. (Facoltativo) Nella sezione Tag, aggiungi le coppie Chiave e Valore per identificare più facilmente la risorsa, per poter cercare o filtrare informazioni specifiche. Puoi anche semplicemente modificare il valore di qualsiasi chiave corrente.

   1. Scegli Aggiungi nuovo tag.
   2. Inserisci il nome della chiave e il valore associato.
   3. Per aggiungere un tag, scegli Aggiungi un nuovo tag.

      Puoi aggiungere più tag scegliendo Aggiungi nuovo tag oppure puoi rimuovere qualsiasi tag scegliendo Rimuovi.
   4. Se desideri associare i tag al monitor, tieni selezionata l'opzione Aggiungi tag alle sonde create dal monitor. Questa operazione aggiunge i tag alle sonde del monitor, il che può essere utile se utilizzi l'autenticazione o la misurazione basata su tag.

7. Seleziona Salvataggio delle modifiche.

Modifica di un monitor tramite la CLI o l'API

Utilizza la riga di comando o l'API per visualizzare e modificare un monitor.

Per modificare un monitor utilizzando l'API o la riga di comando

1. Usa list-monitors per ottenere un elenco dei tuoi monitor se non conosci il nome del monitor. Prendi nota del nome del monitor che desideri modificare.

2. Usa edit-monitor con il nome del monitor del passaggio precedente.

Eliminazione di un monitor

Prima di poter eliminare un monitor, è necessario disattivare o eliminare tutte le sonde associate a quel monitor, indipendentemente dallo stato del monitor. Dopo aver disattivato o eliminato un monitor, non ti verrà più addebitato alcun costo per le relative sonde. Non è possibile recuperare un monitor eliminato. Puoi eliminare un monitor utilizzando la Amazon CloudWatch console o utilizzando la riga di comando/API.

Sebbene una sonda possa essere eliminata o disattivata, conserva CloudWatch comunque le metriche per 15 giorni.

Eliminazione di un monitor tramite la console

Usa la CloudWatch console per eliminare un monitor.

Eliminazione di un monitor tramite la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.
2. Nella sezione Monitor di rete, scegli il monitor che desideri eliminare.
3. Scegli Azioni, quindi Elimina.
4. Se hai delle sonde attive, ti verrà richiesto di disattivarle. Scegli Disattiva le sonde.

> **Note**
>
> Dopo aver selezionato Disattiva sonde, non è possibile annullare l'azione. Le sonde disattivate, tuttavia, non vengono rimosse dal monitor. Puoi riattivarle successivamente. Per informazioni, consulta the section called “Attivazione o disattivazione di una sonda”.

5. Nel campo di conferma inserisci **`confirm`** e quindi scegli Elimina.

Eliminazione di un monitor tramite l'API o la riga di comando

Elimina un monitor tramite l'API o la riga di comando.

Per eliminare un monitor di rete utilizzando l'API o la riga di comando

1. Avrai bisogno del nome del monitor che desideri eliminare. Se non conosci il nome, usa list-monitors per ottenere un elenco dei tuoi monitor. Prendi nota del nome del monitor che desideri eliminare.
2. Verifica se il monitor contiene delle sonde. Usa get-monitor con il nome del monitor del passaggio precedente. Questo restituisce un elenco di tutte le sonde associate a quel monitor.
3. Se il monitor contiene delle sonde, dovrai prima impostarle come inattive o eliminarle.
   - Per impostare una sonda come inattiva, usa update-probe e imposta lo stato su `INACTIVE`.
   - Per eliminare una sonda, usa delete-probe.
4. Una volta impostate su `INACTIVE` o eliminate le sonde, utilizza delete-monitor per eliminare il monitor. Le sonde inattive non vengono eliminate.

Attivazione o disattivazione di una sonda

È possibile attivare o disattivare una sonda di monitoraggio a seconda delle necessità. Potresti voler disattivare una sonda se attualmente non la usi, ma potresti volerla riutilizzare in futuro. Disattivando una sonda non perderai tempo a configurarla di nuovo. I costi delle sonde disattivate non vengono addebitati.

Puoi modificare lo stato di un monitor utilizzando la Amazon CloudWatch console o utilizzando la riga di comando o l'API.

Impostazione di una sonda su attiva o inattiva tramite la console

Usa la CloudWatch console per impostare una sonda come attiva o inattiva.

Per impostare una sonda su attiva o inattiva tramite la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.
2. Scegli la scheda Dettagli del monitor.
3. Nella sezione Sonde, scegli la sonda che desideri attivare o disattivare.
4. Scegli Azioni, quindi seleziona Attiva o Disattiva.

> **Note**
>
> Se stai riattivando una sonda disattivata, ti saranno addebitati nuovamente i costi relativi alla sonda.

Impostazione di una sonda su attiva o inattiva tramite la riga di comando o l'API

Attiva o disattiva una sonda tramite la riga di comando o l'API. Puoi utilizzare questo comando solo per una singola sonda.

Per impostare una sonda su attiva o inattiva tramite la riga di comando o l'API

1. Usa list-monitors per ottenere un elenco dei tuoi monitor se non conosci il nome del monitor. Prendi nota del nome del monitor per cui desideri modificare lo stato della sonda.
2. Usa get-monitor con il nome del monitor del passaggio precedente. Questo restituisce un elenco di tutte le sonde associate a quel monitor. Prendi nota dell'ID delle sonde per cui desideri modificare lo stato.
3. Usa update-probe e imposta lo stato della sonda che desideri modificare su `ACTIVE` o su `INACTIVE`.

Aggiunta di una sonda a un monitor

È possibile aggiungere una sonda a un monitor esistente. Tieni presente che se aggiungi delle sonde a un monitor, la struttura di fatturazione verrà aggiornata e mostrerà che è stata aggiunta una nuova sonda.

Aggiunta di una sonda a un monitor utilizzando la console

Per aggiungere una sonda a un monitor utilizzando la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.
2. Nella sezione Monitor di rete, esegui una delle operazioni seguenti:
   - Scegli il collegamento Nome del monitor al quale desideri aggiungere una sonda. Seleziona la scheda Dettagli del monitor, quindi nella sezione Sonde, seleziona Aggiungi sonda.
   - Seleziona la casella di controllo del monitor, seleziona Operazioni, quindi Aggiungi sonda.

3. Nella pagina Aggiungi sonda esegui queste operazioni:

    1. Nell'origine di rete AWS , scegli una sottorete da aggiungere al monitor.

        > **Note**
        >
        > È possibile aggiungere una sola sonda alla volta e fino a quattro sonde per monitor.

    2. Inserisci l'indirizzo IP di destinazione della rete on-premise. Sono supportati sia indirizzi IPv4 che IPv6.
    3. Scegli Impostazioni avanzate.
    4. Scegli il protocollo di rete per la destinazione. Questo può essere ICMP o TCP.
    5. Se il protocollo è TCP, inserisci le seguenti informazioni. Altrimenti, passa alla fase successiva:
        - Inserisci la porta utilizzata dalla rete per la connessione. La porta deve essere un numero compreso tra 1 e 65535.
        - Inserisci la dimensione del pacchetto. Questa è la dimensione, in byte, di ogni pacchetto inviato con la sonda tra l'origine e la destinazione. La dimensione del pacchetto deve essere un numero compreso tra 56 e 8500.
4. (Facoltativo) Nella sezione Tag, aggiungi le coppie Chiave e Valore per identificare più facilmente la risorsa, per poter cercare o filtrare informazioni specifiche.

    1. Scegli Aggiungi nuovo tag.
    2. Inserisci il nome della chiave e il valore associato.
    3. Per aggiungere il nuovo tag, seleziona Aggiungi un nuovo tag.

        Puoi aggiungere più tag scegliendo Aggiungi nuovo tag oppure puoi rimuovere qualsiasi tag scegliendo Rimuovi.
5. Seleziona Aggiungi sonda.

    Durante l'attivazione della sonda, lo stato visualizzato è In sospeso. Potrebbero essere necessari diversi minuti prima che la sonda diventi Attiva.

Aggiunta di una sonda a un monitor utilizzando l'API o la riga di comando

Aggiungi una sonda a un monitor utilizzando l'API o la riga di comando. Puoi utilizzare questo comando per aggiungere una sola sonda alla volta.

Per aggiungere una sonda a un monitor utilizzando l'API o la riga di comando

1. Usa list-monitors per ottenere un elenco dei tuoi monitor se non conosci il nome del monitor. Prendi nota del nome del monitor al quale desideri aggiungere una sonda.
2. Usa create-probe per aggiungere una sonda al monitor.

Modifica di una sonda

È possibile modificare qualsiasi informazione relativa a una sonda corrente, indipendentemente dal fatto che la sonda sia attivata o disattivata. Puoi modificare una sonda utilizzando la console Amazon CloudWatch oppure tramite la riga di comando o l'API.

Modifica di una sonda tramite la console

Per modificare una sonda tramite la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.

   Scegli il collegamento Nome per aprire il pannello di controllo del monitor.
2. Scegli la scheda Dettagli del monitor.
3. Nella sezione Sonde, scegli il collegamento per la sonda che desideri modificare.
4. Nella pagina del pannello di controllo della sonda, scegli Modifica oppure Operazioni, quindi seleziona Modifica.
5. Nella pagina Modifica sonda, inserisci il nuovo indirizzo IP della sonda di destinazione. Sono supportati sia indirizzi IPv4 che IPv6.
6. Scegli Impostazioni avanzate.
7. Scegli il protocollo di rete. Questo può essere ICMP o TCP.
8. Se il protocollo è TCP, inserisci le seguenti informazioni. Altrimenti, passa alla fase successiva:

   - Inserisci la porta utilizzata dalla rete per la connessione. La porta deve essere un numero compreso tra 1 e 65535.
   - Inserisci la dimensione del pacchetto. Questa è la dimensione, in byte, di ogni pacchetto inviato con la sonda tra l'origine e la destinazione. La dimensione del pacchetto deve essere un numero compreso tra 56 e 8500.
9. (Facoltativo) Nella sezione Tag, aggiungi le coppie Chiave e Valore per identificare più facilmente la risorsa, per poter cercare o filtrare informazioni specifiche.

1. Scegli Aggiungi nuovo tag.
2. Inserisci il nome della chiave e il valore associato.
3. Per aggiungere il nuovo tag, seleziona Aggiungi un nuovo tag.

   Puoi aggiungere più tag scegliendo Aggiungi nuovo tag oppure puoi rimuovere qualsiasi tag scegliendo Rimuovi.

10. Seleziona Salvataggio delle modifiche.

Modifica di una sonda utilizzando l'API o la riga di comando

Utilizza la riga di comando per modificare una sonda. Puoi utilizzare questo comando solo per una singola sonda.

Per modificare una sonda utilizzando l'API o la riga di comando

1. Usa list-monitors per ottenere un elenco dei tuoi monitor se non conosci il nome del monitor. Prendi nota del nome del monitor per cui desideri modificare lo stato della sonda.
2. Usa get-monitor con il nome del monitor del passaggio precedente. Questo restituisce un elenco di tutte le sonde associate a quel monitor. Annota l'ID delle sonde che desideri modificare.
3. Usa update-probe per modificare le informazioni della sonda.

Eliminazione di una sonda

Puoi eliminare una sonda anziché disattivarla se sai che non ti servirà più in futuro. Non è possibile ripristinare una sonda eliminata e sarebbe invece necessario ricrearla. Quando la sonda viene eliminata la relativa fatturazione si interrompe. Puoi eliminare una sonda utilizzando la console Amazon CloudWatch oppure la riga di comando o l'API.

Eliminazione di una sonda tramite la console

Per eliminare una sonda tramite la console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.
2. Nella sezione Monitor di rete, scegli il collegamento Nome per aprire il pannello di controllo del monitor.
3. Scegli la scheda Dettagli del monitor.

4. Seleziona la casella di controllo del monitor, seleziona Operazioni, quindi Elimina.
5. Nella finestra di dialogo Elimina sonda scegli Elimina per confermare l'eliminazione della sonda.
6. Seleziona Elimina per confermare che desideri eliminare la sonda.

   Lo stato della sonda nella sezione Sonde mostra Eliminazione in corso. Dopo l'eliminazione, la sonda viene rimossa dalla sezione Sonde.

Eliminazione di una sonda utilizzando l'API o la riga di comando

Elimina una sonda utilizzando l'API o la riga di comando. Puoi utilizzare questo comando solo per una singola sonda.

Per impostare una sonda su attiva o inattiva tramite la riga di comando o l'API

1. Usa list-monitors per ottenere un elenco dei tuoi monitor se non conosci il nome del monitor. Prendi nota del nome del monitor con la sonda che desideri eliminare.
2. Usa get-monitor con il nome del monitor del passaggio precedente. Questo restituisce un elenco di tutte le sonde associate a quel monitor. Annota l'ID delle sonde che desideri eliminare.
3. Usa delete-probe.

Aggiunta o rimozione dei tag alle risorse utilizzando l'API o la riga di comando

È possibile utilizzare la riga di comando o la CLI per aggiungere o aggiornare i tag delle risorse.

Per aggiornare i tag delle risorse di Monitor di rete utilizzando la riga di comando o la CLI

- Per elencare i tag delle risorse, usa list-tags-for-resources.
- Per aggiungere tag a una risorsa, usa tag-resource.
- Per rimuovere i tag da una risorsa, usa untag-resource.

Pannelli di controllo di Monitor di rete

Puoi utilizzare la dashboard di Amazon CloudWatch Network Monitor per visualizzare lo stato della AWS rete e sondare i tempi di andata e ritorno e la perdita di pacchetti. Puoi visualizzare questi parametri sia per i monitor che per le singole sonde.

Pannelli di controllo di Monitor di rete

- Pannello di controllo del monitor
- Pannello di controllo della sonda

Allarmi delle sonde

Puoi creare CloudWatch allarmi Amazon in base ai parametri di Amazon CloudWatch Network Monitor, proprio come puoi fare per altri parametri Amazon CloudWatch . Qualsiasi allarme creato verrà visualizzato nella colonna Stato della sonda della sezione Dettagli del monitor della dashboard di Network Monitor quando viene attivato l'allarme. Lo stato sarà OK o In allarme. Se non viene visualizzato lo stato di una sonda, non è stato creato alcun allarme per quella sonda.

Ad esempio, puoi creare un allarme basato sul parametro `PacketLoss` di Monitor di rete e configurarlo per inviare una notifica quando il parametro è superiore a un valore scelto. Gli allarmi per le metriche di Network Monitor vengono configurati seguendo le stesse linee guida utilizzate per le altre metriche. CloudWatch

Le seguenti metriche sono disponibili nella sezione `AWS/NetworkMonitor` Quando si crea un CloudWatch allarme per Network Monitor.

- HealthIndicator
- PacketLoss
- RTT (Tempo di andata e ritorno)

Per i passaggi da seguire per creare un allarme per Monitor di rete, consulta the section called “Creare un allarme basato su una soglia statica”.

Impostazione di un intervallo di tempo per i parametri

I parametri e gli eventi su entrambi i pannelli di controllo utilizzano un tempo predefinito di due ore, calcolato in base all'ora corrente. È possibile modificare l'impostazione predefinita per utilizzare una delle seguenti preimpostazioni:

- 1h: un'ora
- 2h: due ore
- 1d: un giorno
- 1w: una settimana

Puoi anche impostare un intervallo di tempo personalizzato. Seleziona Personalizza, scegli un tempo Assoluto o Relativo, quindi imposta l'intervallo di tempo su un orario a tua scelta. Per CloudWatch impostazione predefinita, il tempo relativo supporta solo 15 giorni dalla data odierna.

Inoltre, puoi scegliere l'ora visualizzata nei grafici in base al fuso orario UTC o al fuso orario locale.

Pannello di controllo del monitor

Puoi utilizzare la dashboard di Amazon CloudWatch Network Monitor per visualizzare lo stato della AWS rete e sondare i tempi di andata e ritorno e la perdita di pacchetti. Monitor di rete dispone di pannelli di controllo sia per i monitor che per le sonde.

Per accedere a un pannello di controllo

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.
2. Nella sezione Monitor di rete, scegli il collegamento Nome per aprire il pannello di controllo del monitor.

Panoramica

La pagina Panoramica contiene le seguenti informazioni del monitor:

- AWS Integrità della AWS rete: lo stato della rete mostra lo stato generale della sola AWS rete. Lo stato può essere Integro o Deteriorato. Uno stato integro indica che Network Monitor non ha rilevato alcun problema con la AWS rete. Lo stato Degradato indica che Network Monitor ha rilevato un problema con la AWS rete. La barra di stato in questa sezione mostra lo stato della rete per un periodo predefinito di un'ora. Passa il mouse su un punto qualsiasi della barra di stato per visualizzare ulteriori dettagli.
- Riepilogo del traffico della sonda: visualizza lo stato attuale del traffico tra le AWS sottoreti di origine nel monitor e gli indirizzi IP di destinazione. Il riepilogo del traffico della sonda mostra quanto segue:
  - Sonde in allarme: questo numero indica quante sonde si trovano in uno stato di deterioramento. Un allarme viene attivato quando viene attivato un parametro che hai impostato come allarme. Per informazioni sugli allarmi metrici di Network Monitor, vedere. the section called “Allarmi delle sonde”
  - Perdita di pacchetti: il numero di pacchetti persi dalla sottorete di origine all'indirizzo IP di destinazione. Viene rappresentato come percentuale del totale dei pacchetti inviati.

- Tempo di andata e ritorno: il tempo impiegato, in millisecondi, affinché un pacchetto proveniente dalla sottorete di origine raggiunga l'indirizzo IP di destinazione e poi ritorni.

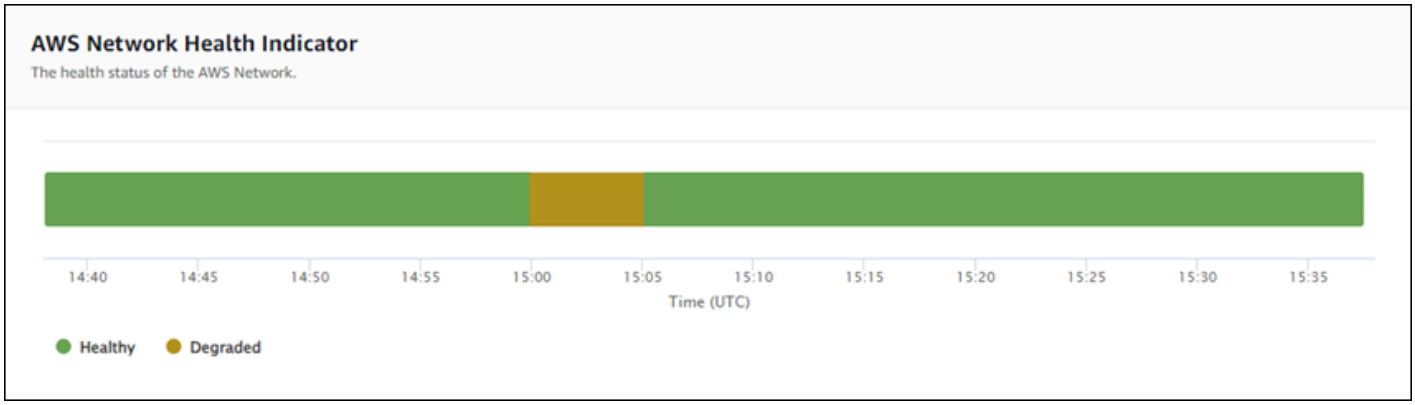
I dati sono rappresentati da un grafico interattivo che consente di visualizzare i dettagli.

Per impostazione predefinita, i dati vengono visualizzati per un periodo di due ore, calcolato in base alla data e all'ora correnti. Tuttavia, puoi modificare l'intervallo in base alle tue esigenze. Per ulteriori informazioni, consulta the section called “Impostazione di un intervallo di tempo per i parametri”.

Parametri di tracciamento

Il pannello di controllo di Monitor di rete mostra una rappresentazione grafica dei monitor e delle sonde. Sono disponibili i seguenti grafici:

- AWS Indicatore dello stato della rete: rappresenta lo stato della AWS rete in un determinato periodo. Lo stato può essere Integro o Deteriorato. Nell'esempio seguente, vedrai che dalle 15:00 UTC alle 15:05 UTC, la AWS rete era in uno stato degradato. Dopo le 15:05 la rete è tornata a uno stato integro. Puoi passare il mouse su qualsiasi sezione del grafico per visualizzare ulteriori dettagli.



> **Note**
>
> Il Network Health Indicator non indica lo stato della sonda ma solo della AWS rete.

- Perdita di pacchetti: questo grafico mostra una riga univoca che mostra la percentuale di perdita di pacchetti per ciascuna sonda del monitor. La legenda nella parte inferiore della pagina mostra ciascuna delle sonde del monitor, codificate a colori per renderle uniche. Il passaggio del mouse su una sonda in questo grafico mostra la sottorete di origine, l'IP di destinazione e la percentuale di perdita di pacchetti. Nell'esempio seguente, è stato impostato un allarme per la perdita di pacchetti

per una sonda da una sottorete all'indirizzo IP 127.0.0.1. L'allarme si è stato attivato quando la soglia di perdita di pacchetti è stata superata per la sonda. Il passaggio del mouse sul grafico mostra l'origine e la destinazione della sonda e mostra che per questa sonda il 21 novembre alle 02:41:30 si è verificata una perdita di pacchetti del 30,97%.

- Tempo di andata e ritorno: questo grafico presenta una riga per ogni sonda, in cui è mostrato il relativo tempo di andata e ritorno. La legenda nella parte inferiore della pagina mostra ciascuna delle sonde del monitor, codificate a colori per renderle uniche. Il passaggio del mouse su una sonda in questo grafico mostra la sottorete di origine, l'indirizzo IP di destinazione e il tempo di andata e ritorno. L'esempio seguente mostra che martedì 21 novembre alle 21:45:30, il tempo di andata e ritorno di una sonda da una sottorete all'indirizzo IP 127.0.0.1 è stato di 0,075 secondi.

Dettagli del monitor

La pagina dei dettagli del monitor mostra i dettagli sul monitor, comprese le sonde. In questa pagina puoi gestire i tag o aggiungere una sonda. Questa pagina è suddivisa nelle seguenti tre sezioni:

- Dettagli del monitor: questa pagina fornisce dettagli sul monitor. Le informazioni in questa sezione non possono essere modificate. Tuttavia, puoi scegliere il collegamento Nome ruolo per visualizzare i dettagli del ruolo collegato al servizio Monitor di rete.
- Sonde: questa sezione mostra un elenco di tutte le sonde associate al monitor. Scegli un collegamento VPC o ID sottorete per aprire i dettagli del VPC o della sottorete nella console Amazon VPC. Puoi anche modificare una sonda, incluse le opzioni di attivazione o disattivazione. Per ulteriori informazioni, consulta the section called “Utilizzo di monitor e sonde”.

  La sezione Probes mostra le informazioni su ogni sonda configurata per quel monitor, tra cui l'ID sonda, l'ID VPC, l'ID di sottorete, l'indirizzo IP, il protocollo e se lo stato della sonda è attivo o inattivo. Se hai impostato un allarme per una sonda, viene visualizzato lo stato corrente di tale allarme. OK indica che non ci sono metriche gli eventi che hanno attivato alcun allarme; In allarme indica che una metrica impostata in ha attivato un allarme. CloudWatch Se non viene visualizzato alcuno stato per una sonda, significa che non è stato impostato alcun allarme. CloudWatch Per informazioni sui tipi di allarmi con sonda Network Monitor che è possibile creare, vedere. the section called “Allarmi delle sonde”
- Tag: visualizza i tag correnti per un monitor. Puoi aggiungere o rimuovere tag scegliendo Gestisci tag. Questa operazione apre la pagina Modifica sonda. Per ulteriori informazioni sulla modifica dei tag, consulta the section called “Modifica di un monitor”.

Pannello di controllo della sonda

Puoi utilizzare la dashboard di Amazon CloudWatch Network Monitor per visualizzare lo stato della AWS rete e informazioni sui tempi di andata e ritorno specifici e sulla perdita di pacchetti per sonde specifiche. Sono disponibili due pannelli di controllo delle sonde, Panoramica e Dettagli sonda.

Puoi creare CloudWatch allarmi per impostare soglie metriche relative alla perdita di pacchetti e ai tempi di andata e ritorno. Quando viene raggiunta una soglia per una metrica, un allarme ti avvisa. CloudWatch Per informazioni sulla creazione degli allarmi delle sonde, consulta the section called “Allarmi delle sonde”.

Per accedere al pannello di controllo di una sonda

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/, quindi in Network Monitoring, scegli Network Monitor.
2. Nella sezione Monitor di rete, scegli il collegamento Nome per aprire il pannello di controllo del monitor.
3. Scegli il collegamento ID per visualizzare il pannello di controllo della relativa sonda.

Panoramica

La pagina Panoramica contiene le seguenti informazioni della sonda:

- AWS Dettagli del Network Health Indicator: forniscono lo stato generale della sola AWS rete. Lo stato può essere Integro o Deteriorato. Lo stato Degradato indica che c'è un problema con la AWS rete e non indica se c'è un problema con la sonda.
- Perdita di pacchetti: il numero di pacchetti persi dalla sottorete di origine all'indirizzo IP di destinazione per la sonda.
- Tempo di andata e ritorno: il tempo impiegato, in millisecondi, affinché un pacchetto proveniente dalla sottorete di origine raggiunga l'indirizzo IP di destinazione e poi ritorni.

Dettagli sonda

La pagina dei Dettagli sonda mostra i dettagli relativi a una sonda. In questa pagina è possibile modificare la sonda. Per ulteriori informazioni, consulta the section called “Utilizzo di monitor e sonde”.

- Dettagli sonda: questa pagina fornisce informazioni generali sulla sonda. Le informazioni in questa sezione non possono essere modificate.
- Origine e destinazione della sonda: questa sezione mostra i dettagli della sonda. Scegli un collegamento VPC o ID sottorete per aprire i dettagli del VPC o della sottorete nella console Amazon VPC. Puoi anche modificare una sonda, incluse le opzioni di attivazione o disattivazione.
- Tag: visualizza i tag correnti per un monitor. Puoi aggiungere o rimuovere tag scegliendo Gestisci tag. Questa operazione apre la pagina Modifica sonda. Per ulteriori informazioni sulla modifica dei tag, consulta the section called “Modifica di una sonda ”.

Quote di Monitor di rete

Di seguito sono riportate le quote di Monitor di rete:

Quota	Predefinita	Adattabile
Numero massimo di monitor per account Regione AWS	100	Sì
Numero massimo di sonde per monitor	24	Sì
Numero massimo di sonde sottorete per monitor	4	Sì

Sicurezza e protezione dei dati in Monitor di rete

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te e te. AWS Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei

AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per maggiori informazioni sui programmi di conformità che si applicano ad Amazon CloudWatch Network Monitor, consulta AWS Services in Scope by Compliance Program AWS .

- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza CloudWatch Network Monitor. I seguenti argomenti mostrano come configurare CloudWatch Network Monitor per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse CloudWatch di Network Monitor.

Argomenti

Protezione dei dati in Amazon CloudWatch Network Monitor

Il modello di responsabilità AWS condivisa modello si applica alla protezione dei dati in Amazon CloudWatch Network Monitor. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail

- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con CloudWatch Network Monitor o altro Servizi AWS utilizzando la console, l'API o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Sicurezza dell'infrastruttura in Amazon CloudWatch Network Monitor

In quanto servizio gestito, Amazon CloudWatch Network Monitor è protetto dalle procedure di sicurezza di rete AWS globali descritte nel white paper Amazon Web Services: Overview of Security Processes.

Utilizzi chiamate API AWS pubblicate per accedere a CloudWatch Network Monitor attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.0 o versioni successive. È consigliabile TLS 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Gestione delle identità e degli accessi per Amazon CloudWatch Network Monitor

AWS Identity and Access Management (IAM) è AWS un servizio che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può

essere autenticato (effettuato l'accesso) e autorizzato (dispone delle autorizzazioni) a utilizzare CloudWatch le risorse di Network Monitor. IAM è un AWS servizio che puoi utilizzare senza costi aggiuntivi. Puoi utilizzare le caratteristiche di IAM per consentire ad altri utenti, servizi e applicazioni di utilizzare le risorse AWS completamente o in modo limitato, senza condividere le credenziali di sicurezza.

Per impostazione predefinita, gli utenti IAM non dispongono dell'autorizzazione per creare, visualizzare o modificare le risorse AWS . Per consentire a un utente IAM di accedere alle risorse, ad esempio una rete globale, ed eseguire attività, è necessario:

- Creare una policy IAM che conceda all'utente l'autorizzazione per utilizzare le risorse specifiche e le operazioni API richieste.
- Collegare la policy all'utente IAM o al gruppo a cui appartiene l'utente

Quando si collega una policy a un utente o a un gruppo di utenti, tramite essa vengono concesse o rifiutate agli utenti le autorizzazioni per l'esecuzione delle attività specificate per le risorse indicate.

Chiavi di condizione

L'elemento `Condition` (o blocco di condizioni) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento condizione è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per ulteriori informazioni, consulta Elementi della policy JSON di IAM: operatori di condizione nella Guida per l'utente di AWS Identity and Access Management.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND` logica. Se specifichi più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'`OR`operazione logica. Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM.

È possibile allegare tag alle risorse CloudWatch di Network Monitor o passare i tag in una richiesta a Cloud WAN. Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`,

`aws:RequestTag/key-name` o `aws:TagKeys`. Per ulteriori informazioni, consulta Elementi delle policy JSON di IAM: condizioni nella Guida per l'utente di AWS Identity and Access Management.

Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente di AWS Identity and Access Management.

Contrassegnare le risorse di rete principali

Un tag è un'etichetta di metadati che l'utente o AWS assegna a una AWS risorsa. Ciascun tag è formato da una chiave e da un valore, Per i tag assegnati da te, puoi definire la chiave e il valore. Ad esempio, puoi definire la chiave come `purpose` e il valore di una risorsa come `test`. I tag consentono di eseguire le seguenti operazioni:

- Identifica e organizza le tue risorse. AWS Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate.
- Controlla l'accesso alle tue risorse. AWS Per ulteriori informazioni, consulta Controllo dell'accesso alle AWS risorse tramite tag nella Guida per l'utente di AWS Identify and Access Management.

Come funziona Amazon CloudWatch Network Monitor con IAM

Prima di utilizzare IAM per gestire l'accesso a CloudWatch Network Monitor, scopri quali funzionalità IAM sono disponibili per l'uso con CloudWatch Network Monitor.

Funzionalità IAM che puoi utilizzare con Amazon CloudWatch Network Monitor

Funzionalità IAM	CloudWatch Supporto per Network Monitor
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
Liste di controllo degli accessi (ACL)	No
ABAC (tag nelle policy)	Parziale

Funzionalità IAM	CloudWatch Supporto per Network Monitor
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
☻Ruoli di servizio	No
Ruoli collegati al servizio	Sì

Per avere una visione di alto livello di come CloudWatch Network Monitor e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Policy basate sull'identità per Amazon Network Monitor CloudWatch

Supporta le policy basate su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate sull'identità per Network Monitor CloudWatch

Per visualizzare esempi di politiche basate sull'identità di CloudWatch Network Monitor, vedere. Esempi di policy basate sull'identità per Amazon CloudWatch

Politiche basate sulle risorse all'interno di Network Monitor CloudWatch

Supporta le policy basate su risorse	No

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Azioni politiche per CloudWatch Network Monitor

Supporta le operazioni di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni di CloudWatch Network Monitor, consulta Azioni definite da Amazon CloudWatch Network Monitor nel Service Authorization Reference.

Le azioni politiche in CloudWatch Network Monitor utilizzano il seguente prefisso prima dell'azione:

```
networkmonitor
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "networkmonitor:action1",
      "networkmonitor:action2"
         ]
```

Per visualizzare esempi di politiche basate sull'identità di CloudWatch Network Monitor, vedere. Esempi di policy basate sull'identità per Amazon CloudWatch

Risorse relative alle policy per Network Monitor CloudWatch

Supporta le risorse di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di risorse di CloudWatch Network Monitor e dei relativi ARN, consulta Risorse definite da Amazon CloudWatch Network Monitor nel Service Authorization Reference. Per sapere con quali azioni puoi specificare l'ARN di ogni risorsa, consulta Azioni definite da Amazon CloudWatch Network Monitor.

Chiavi relative alle condizioni delle politiche per CloudWatch Network Monitor

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND` logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di CloudWatch Network Monitor, consulta Condition keys for Amazon CloudWatch Network Monitor nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta Azioni definite da Amazon CloudWatch Network Monitor.

ACL in CloudWatch Network Monitor

Supporta le ACL	No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con Network Monitor CloudWatch

Supporta ABAC (tag nelle policy)	Parziale

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Che cos'è ABAC? nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con CloudWatch Network Monitor

Supporta le credenziali temporanee	Sì

Alcune Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Cambio di un ruolo (console) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni principali per più servizi per Network Monitor CloudWatch

Supporta l'inoltro delle sessioni di accesso (FAS)	Sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

Ruoli di servizio per CloudWatch Network Monitor

Supporta i ruoli di servizio	No

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità CloudWatch di Network Monitor. Modifica i ruoli di servizio solo quando CloudWatch Network Monitor fornisce indicazioni in tal senso.

Utilizzo di un ruolo collegato al servizio per Network Monitor CloudWatch

Supporta i ruoli collegati ai servizi	Sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta Servizi AWS supportati da IAM. Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per Network Monitor CloudWatch

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare le risorse di CloudWatch Network Monitor. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Per dettagli sulle azioni e sui tipi di risorse definiti da CloudWatch Network Monitor, incluso il formato degli ARN per ciascun tipo di risorsa, consulta Azioni, risorse e chiavi di condizione per Amazon CloudWatch Network Monitor nel Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console Network CloudWatch Monitor
- Consentire agli utenti di visualizzare le loro autorizzazioni
- Risoluzione dei problemi relativi all'identità e all'accesso a CloudWatch Network Monitor

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse di CloudWatch Network Monitor nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso ad azioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer nella Guida per l'utente di IAM.

- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console Network CloudWatch Monitor

Per accedere alla console Amazon CloudWatch Network Monitor, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse di CloudWatch Network Monitor presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la console CloudWatch Network Monitor, collega anche il CloudWatch Network Monitor *ConsoleAccess* o la policy *ReadOnly* AWS gestita alle entità. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
```

```
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Risoluzione dei problemi relativi all'identità e all'accesso a CloudWatch Network Monitor

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con CloudWatch Network Monitor e IAM.

Argomenti

- Non sono autorizzato a eseguire un'azione in CloudWatch Network Monitor
- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse CloudWatch di Network Monitor

Non sono autorizzato a eseguire un'azione in CloudWatch Network Monitor

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `networkmonitor:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 networkmonitor:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `networkmonitor:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di trasferire un ruolo a CloudWatch Network Monitor.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in CloudWatch Network Monitor. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse CloudWatch di Network Monitor

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se CloudWatch Network Monitor supporta queste funzionalità, consultaCome CloudWatch funziona Amazon con IAM.
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente IAM.

AWS politiche gestite per CloudWatch Network Monitor

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le policy AWS gestite che scriverle da soli. Creare policy gestite dal cliente IAM per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste politiche coprono casi d'uso comuni e sono disponibili nel tuo AWS account. Per ulteriori informazioni sulle policy AWS gestite, consulta le policy AWS gestite nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi occasionalmente aggiungono altre autorizzazioni a una policy gestita da AWS per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i

servizi aggiornino una policy gestita da AWS quando viene avviata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy `ReadOnlyAccess` AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consulta la sezione Policy gestite da AWS per funzioni di processi nella Guida per l'utente di IAM.

AWS politica gestita: CloudWatchNetworkMonitorServiceRolePolicy

`CloudWatchNetworkMonitorServiceRolePolicy`è associato a un ruolo collegato al servizio che consente al servizio di eseguire azioni per conto dell'utente e accedere alle risorse associate a CloudWatch Network Monitor. Non è possibile allegare la policy alle identità IAM. Per ulteriori informazioni, consulta the section called “Ruoli collegati ai servizi”.

CloudWatch Network Monitoring aggiorna le policy AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per il monitoraggio CloudWatch della rete da quando questo servizio ha iniziato a tenere traccia di queste modifiche nel novembre 2023.

Modifica	Descrizione	Data
CloudWatchNetworkMonitorServiceRolePolicy: Nuova politica.	Nuova politica aggiunta a CloudWatch Network Monitor.	27 novembre 2023
the section called “`AWSServiceRoleForNetworkMonitor`”. Nuovo ruolo.	Nuovo ruolo aggiunto a CloudWatch Network Monitor.	27 novembre 2023

Autorizzazioni IAM per CloudWatch Network Monitor

Per utilizzare Amazon CloudWatch Network Monitor, gli utenti devono disporre delle autorizzazioni corrette.

Per ulteriori informazioni sulla sicurezza in Amazon CloudWatch, consultaGestione delle identità e degli accessi per Amazon CloudWatch.

Autorizzazioni necessarie per visualizzare un monitoraggio

Per visualizzare un monitor per Amazon CloudWatch Network Monitor in AWS Management Console, devi accedere come utente o ruolo con le seguenti autorizzazioni:

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "networkmonitor:Get*",
                "networkmonitor:List*"
                ],
            "Resource": "*"
        }
    ]
}
```

Autorizzazioni necessarie per creare un monitoraggio

Per creare un monitor in Amazon CloudWatch Network Monitor, gli utenti devono avere l'autorizzazione a creare un ruolo collegato al servizio associato a Network Monitor. Per ulteriori informazioni sul ruolo collegato al servizio, consulta Utilizzo di un ruolo collegato al servizio per Network Monitor CloudWatch .

Per creare un monitor per Amazon CloudWatch Network Monitor in AWS Management Console, devi accedere come utente o ruolo con le autorizzazioni incluse nella seguente politica.

> **Note**
>
> Se crei una policy delle autorizzazioni basata su identità più restrittiva, gli utenti con tale policy non saranno in grado di creare un monitor.

```
{
"Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "networkmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "networkmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:GetRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor"
        },
        {
            "Action": [
                "ec2:CreateSecurityGroup",
                "ec2:CreateNetworkInterface",
                "ec2:CreateTags"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Utilizzo di un ruolo collegato al servizio per Network Monitor CloudWatch

Amazon CloudWatch Network Monitor utilizza il seguente ruolo collegato al servizio per le autorizzazioni necessarie per chiamare altri AWS servizi per tuo conto:

- `AWSServiceRoleForNetworkMonitor`

`AWSServiceRoleForNetworkMonitor`

CloudWatch Network Monitoring utilizza il ruolo collegato al servizio denominato `AWSServiceRoleForNetworkMonitor` per aggiornare e gestire i monitor di rete. CloudWatch

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi `AWSServiceRoleForNetworkMonitor` considera attendibile il seguente servizio:

- `networkmonitor.amazonaws.com`

`CloudWatchNetworkMonitorServiceRolePolicy` è associato al ruolo collegato ai servizi e concede al servizio l'accesso alle risorse VPC ed EC2 nel tuo account e alla gestione dei monitor di rete che sono stati creati.

Raggruppamenti di autorizzazioni

La policy è raggruppata nei seguenti set di autorizzazioni:

- **`cloudwatch`**- Ciò consente al responsabile del servizio di pubblicare le metriche di monitoraggio della rete sulle risorse. CloudWatch
- **`ec2`**: consente al principale del servizio di descrivere i VPC e le sottoreti nel tuo account per creare o aggiornare monitor e sonde. Consente inoltre al principale del servizio di creare, modificare ed eliminare gruppi di sicurezza, interfacce di rete e le relative autorizzazioni per configurare il monitor o la sonda per inviare il traffico di monitoraggio agli endpoint.

Per ulteriori informazioni sulla policy, consulta the section called “AWS politiche gestite”.

Quanto segue mostra `CloudWatchNetworkMonitorServiceRolePolicy`.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
```

```
    "Sid": "PublishCw",
    "Effect": "Allow",
    "Action": "cloudwatch:PutMetricData",
    "Resource": "*",
    "Condition": {
     "StringEquals": {
      "cloudwatch:namespace": "AWS/NetworkMonitor"
     }
    }
   },
   {
    "Sid": "DescribeAny",
    "Effect": "Allow",
    "Action": [
     "ec2:DescribeNetworkInterfaces",
     "ec2:DescribeNetworkInterfaceAttribute",
     "ec2:DescribeVpcs",
     "ec2:DescribeNetworkInterfacePermissions",
     "ec2:DescribeSubnets",
     "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
   },
   {
    "Sid": "DeleteModifyEc2Resources",
    "Effect": "Allow",
    "Action": [
     "ec2:AuthorizeSecurityGroupEgress",
     "ec2:CreateNetworkInterfacePermission",
     "ec2:DeleteNetworkInterfacePermission",
     "ec2:RevokeSecurityGroupEgress",
     "ec2:ModifyNetworkInterfaceAttribute",
     "ec2:DeleteNetworkInterface",
     "ec2:DeleteSecurityGroup"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:security-group/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/ManagedByCloudWatchNetworkMonitor": "true"
     }
    }
```

```
  }
 ]
}
```

Creazione del ruolo collegato ai servizi

`AWSServiceRoleForNetworkMonitor`

Non devi creare manualmente il ruolo `AWSServiceRoleForNetworkMonitor`.

- CloudWatch Network Monitor crea il `AWSServiceRoleForNetworkMonitor` ruolo quando crei il tuo primo monitor di rete. Questo ruolo verrà applicato a tutti i monitor che creerai successivamente.

Per creare un ruolo collegato ai servizi, è necessario disporre delle autorizzazioni richieste. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Modifica del ruolo collegato ai servizi

È possibile modificare la descrizione del `AWSServiceRoleForNetworkMonitor` che utilizza IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione del ruolo collegato ai servizi

Se non è più necessario utilizzare CloudWatch Network Monitor, si consiglia di eliminare il `AWSServiceRoleForNetworkMonitor` ruolo.

È possibile eliminare i ruoli collegati ai servizi solo dopo aver eliminato il monitor di rete. Per informazioni su come eliminare il monitor di rete, consulta Eliminazione di un monitor di rete.

Per eliminare i ruoli collegati ai servizi, puoi utilizzare la console IAM, l'interfaccia a riga di comando IAM CLI o l'API IAM. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Dopo aver eliminato `AWSServiceRoleForNetworkMonitor` CloudWatch Network Monitor creerà nuovamente il ruolo quando creerai un nuovo monitor.

Regioni supportate per il ruolo CloudWatch collegato al servizio Network Monitor

CloudWatch Network Monitor supporta il ruolo collegato al servizio in tutti i paesi in Regioni AWS cui il servizio è disponibile. Per ulteriori informazioni, consulta Endpoint AWS nella Riferimenti generali di AWS.

Eliminazione del ruolo collegato ai servizi

Se non è più necessario utilizzare CloudWatch Network Monitor, si consiglia di eliminare il `AWSServiceRoleForNetworkMonitor` ruolo.

È possibile eliminare i ruoli collegati ai servizi solo dopo aver eliminato il monitor di rete. Per informazioni su come eliminare il monitor di rete, consulta Eliminazione di un monitor di rete.

Per eliminare i ruoli collegati ai servizi, puoi utilizzare la console IAM, l'interfaccia a riga di comando IAM CLI o l'API IAM. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Dopo aver eliminato `AWSServiceRoleForNetworkMonitor` CloudWatch Network Monitor creerà nuovamente il ruolo quando creerai un nuovo monitor.

Prezzi

Con Amazon CloudWatch Network Monitor, non ci sono costi iniziali o impegni a lungo termine. I prezzi di Monitor di rete comprendono i due componenti seguenti:

- una tariffa oraria per risorsa monitorata e
- CloudWatch commissioni relative alle metriche.

Quando si crea un monitor di rete, si associano le risorse da monitorare. Per Network Monitor queste saranno sottoreti nel tuo (Amazon Virtual Private Cloud VPC). Ogni risorsa monitorata consente di creare fino a quattro sonde da ciascuna sottorete dei VPC verso quattro destinazioni. Per tenere sotto controllo la fattura, puoi modificare la copertura della sottorete e la copertura IP on-premise riducendo il numero di risorse monitorate.

Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Monitoraggio dell'infrastruttura

Gli argomenti di questa sezione spiegano CloudWatch le funzionalità che possono aiutarti a ottenere visibilità operativa delle tue AWS risorse.

Argomenti

Container Insights

Usa CloudWatch Container Insights per raccogliere, aggregare e riepilogare metriche e log dalle tue applicazioni e microservizi containerizzati. Container Insights è disponibile per le piattaforme Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) e Kubernetes su Amazon EC2. Container Insights supporta la raccolta di metriche dai cluster distribuiti AWS Fargate sia per Amazon ECS che per Amazon EKS.

CloudWatch raccoglie automaticamente i parametri per molte risorse, come CPU, memoria, disco e rete. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che consentono di isolare i problemi e risolverli in modo rapido. Puoi anche impostare CloudWatch allarmi sulle metriche raccolte da Container Insights.

Container Insights raccoglie dati come eventi di log delle prestazioni tramite Embedded Metric Format. Questi eventi di log delle prestazioni sono elementi che usano uno schema JSON strutturato che consente ai dati ad alta cardinalità di essere acquisiti e archiviati su larga scala. Da questi dati, CloudWatch crea metriche aggregate a livello di cluster, nodo, pod, task e servizio come metriche. CloudWatch Le metriche raccolte da Container Insights sono disponibili nei dashboard CloudWatch automatici e sono visualizzabili anche nella sezione Metriche della console. CloudWatch I parametri non sono visibili fino a quando le attività del container non sono in esecuzione da qualche tempo.

Quando si implementa Approfondimenti sui container, viene creato automaticamente un gruppo di log per gli eventi del log delle prestazioni. Non è necessario creare questo gruppo di log da soli.

Per aiutarti a gestire i costi di Container Insights, CloudWatch non crea automaticamente tutte le metriche possibili dai dati di registro. Tuttavia, puoi visualizzare metriche aggiuntive e livelli di granularità aggiuntivi utilizzando CloudWatch Logs Insights per analizzare gli eventi non elaborati del registro delle prestazioni.

Con la versione originale di Approfondimenti sui container, i parametri raccolti e i log importati vengono addebitati come parametri personalizzati. Con Approfondimenti sui container con osservabilità migliorata per Amazon EKS, i parametri e i log di Approfondimenti sui container vengono addebitati per osservazione anziché per parametro archiviato o log importato. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

In Amazon EKS e Kubernetes, Container Insights utilizza una versione containerizzata dell' CloudWatch agente per scoprire tutti i container in esecuzione in un cluster. Quindi raccoglie i dati sulle prestazioni a ogni livello dello stack delle prestazioni.

Container Insights supporta la crittografia AWS KMS key per i log e le metriche che raccoglie. Per abilitare questa crittografia, è necessario abilitare manualmente la AWS KMS crittografia per il gruppo di log che riceve i dati di Container Insights. In questo modo, Approfondimenti sui container crittografa questi dati utilizzando la chiave KMS fornita. Sono supportate solo le chiavi simmetriche. Non utilizzare chiavi KMS asimmetriche per crittografare i gruppi di log.

Per ulteriori informazioni, consulta Encrypt Log Data in CloudWatch Logs Using. AWS KMS

Approfondimenti sui container con osservabilità migliorata per Amazon EKS

Il 6 novembre 2023 è stata rilasciata una nuova versione di Approfondimenti sui container. Questa versione supporta l'osservabilità migliorata per i cluster Amazon EKS in esecuzione su Amazon EC2 e può raccogliere parametri più dettagliati da questi cluster. Dopo l'installazione, raccoglie automaticamente la telemetria dettagliata dell'infrastruttura e i log dei container per i cluster Amazon EKS. Puoi quindi impiegare pannelli di controllo accurati e immediatamente utilizzabili per approfondire la telemetria delle applicazioni e dell'infrastruttura.

Approfondimenti sui container con osservabilità migliorata per Amazon EKS raccoglie parametri granulari su integrità, prestazioni e stato fino al livello del container, oltre a parametri del piano di controllo. Per ulteriori informazioni sui parametri aggiuntivi e le dimensioni che possono essere raccolti, consulta la sezione Parametri di Container Insights per Amazon EKS e Kubernetes.

Se hai installato Container Insights utilizzando l' CloudWatch agente su un cluster Amazon EKS su Amazon EC2 dopo il 6 novembre 2023, disponi di Container Insights con osservabilità migliorata

per Amazon EKS. In alternativa, puoi aggiornare un cluster Amazon EKS a questa nuova versione seguendo le istruzioni riportate alla pagina Aggiornamento a Approfondimenti sui container con osservabilità migliorata per Amazon EKS.

Container Insights supporta l'osservabilità CloudWatch tra account. Utilizzi un unico account di monitoraggio per monitorare e risolvere i problemi delle applicazioni che si estendono su più AWS account all'interno di una singola regione. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Container Insights con osservabilità migliorata per Amazon EKS supporta anche i nodi di lavoro Windows.

Approfondimenti sui container con osservabilità migliorata per Amazon EKS non è supportato su Fargate.

> **Note**
>
> Puoi scoprire se disponi di cluster che possono essere aggiornati ad Approfondimenti sui container con osservabilità migliorata per Amazon EKS accedendo alla console Approfondimenti sui container. Per farlo, scegli Insights, Container Insights nel pannello di navigazione della CloudWatch console. Nella console Approfondimenti sui container, un banner ti informa se sono presenti cluster Amazon EKS che possono essere aggiornati e contiene collegamenti alla pagina di aggiornamento.

Piattaforme supportate

Container Insights è disponibile per le piattaforme Amazon Elastic Container Service, Amazon Elastic Kubernetes Service e Kubernetes su istanze Amazon EC2.

- Per Amazon ECS, Approfondimenti sui container raccoglie i parametri a livello di cluster, attività e servizio nelle istanze Linux e Windows Server. Può raccogliere i parametri a livello di istanza solo sulle istanze Linux.

  Per Amazon ECS, i parametri di rete sono disponibili solo per i container in modalità di rete `bridge` e modalità di rete `awsvpc`. Non sono disponibili per i container in modalità di rete `host`.
- Per le piattaforme Amazon Elastic Kubernetes Service e Kubernetes sulle istanze Amazon EC2, Container Insights è supportato solo sulle istanze Linux.

CloudWatch immagine del contenitore dell'agente

Amazon fornisce un'immagine del contenitore dell' CloudWatch agente su Amazon Elastic Container Registry. Per maggiori informazioni sull'utilizzo di CloudTrail con Amazon ECR, consulta cloudwatch-agent.

Regioni supportate

Container Insights per Amazon ECS è supportato nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Sydney)
- Canada occidentale (Calgary)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Spagna)

- Europa (Stoccolma)
- Europa (Zurigo)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti orientali)
- AWS GovCloud (Stati Uniti occidentali)
- Cina (Pechino)
- Cina (Ningxia)

Regioni supportate per Amazon EKS e Kubernetes

Container Insights per Amazon EKS e Kubernetes è supportato nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- US West (Oregon)
- Asia Pacifico (Hong Kong)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Cina (Pechino)
- China (Ningxia)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europe (Paris)

- Europa (Stoccolma)
- Medio Oriente (Bahrein)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti orientali)
- AWS GovCloud (Stati Uniti occidentali)

Configurazione di Container Insights

Il processo di configurazione di Container Insights è diverso per Amazon ECS, per Amazon EKS e per Kubernetes.

Argomenti

- Configurazione di Container Insights su Amazon ECS
- Configurazione di Container Insights su Amazon EKS e Kubernetes

Configurazione di Container Insights su Amazon ECS

Puoi utilizzare una o entrambe le opzioni seguenti per abilitare Container Insights su cluster di Amazon ECS:

- Usa AWS Management Console o the AWS CLI per iniziare a raccogliere metriche a livello di cluster, di attività e a livello di servizio.
- Implementa l' CloudWatch agente come servizio daemon per iniziare a raccogliere parametri a livello di istanza sui cluster ospitati su istanze Amazon EC2.

Argomenti

- Configurazione di Container Insights su Amazon ECS per parametri a livello di cluster e di servizio
- Configurazione di Container Insights su Amazon ECS utilizzando AWS Distro per OpenTelemetry
- Implementazione dell' CloudWatch agente per raccogliere parametri a livello di istanza EC2 su Amazon ECS
- Implementazione della AWS distribuzione per raccogliere parametri OpenTelemetry a livello di istanza EC2 sui cluster Amazon ECS
- Configurazione FireLens per l'invio di log a CloudWatch Logs

Configurazione di Container Insights su Amazon ECS per parametri a livello di cluster e di servizio

Puoi abilitare Container Insights su cluster Amazon ECS nuovi ed esistenti. Container Insights raccoglie i parametri a livello di cluster, attività e servizio. Puoi abilitare Container Insights utilizzando la console Amazon ECS o il AWS CLI.

Se utilizzi Amazon ECS in un'istanza Amazon EC2 e desideri raccogliere i parametri di rete e di storage da Container Insights, avvia l'istanza utilizzando un'AMI che include l'agente di Amazon ECS versione 1.29. Per informazioni su come aggiornare la versione dell'agente, consulta Aggiornamento dell'agente del container di Amazon ECS

Puoi utilizzare il AWS CLI per impostare l'autorizzazione a livello di account per abilitare Container Insights per tutti i nuovi cluster Amazon ECS creati nel tuo account. A questo scopo, immetti il comando seguente.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

> **Note**
>
> Se la AWS KMS chiave gestita dal cliente che utilizzi per i parametri di Amazon ECS Container Insights non è già configurata per funzionare CloudWatch, devi aggiornare la policy delle chiavi per consentire i log crittografati nei log. CloudWatch È inoltre necessario associare la propria AWS KMS chiave al gruppo di log sotto. `/aws/ecs/containerinsights/ClusterName/performance` Per ulteriori informazioni, consulta Crittografare i dati di registro in CloudWatch Logs using. AWS Key Management Service

Configurazione di Container Insights su cluster Amazon ECS esistenti

Per abilitare Container Insights su un cluster Amazon ECS esistente, immetti il comando seguente. È necessario eseguire la versione 1.16.200 o successiva di AWS CLI affinché il seguente comando funzioni.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Configurazione di Container Insights su cluster Amazon ECS nuovi

Sono disponibili due modi per abilitare Container Insights su nuovi cluster Amazon ECS. Puoi configurare Amazon ECS in modo che tutti i nuovi cluster siano abilitati per Container Insights

per impostazione predefinita. In caso contrario, puoi abilitare un nuovo cluster al momento della creazione.

Utilizzo di AWS Management Console

Puoi abilitare Approfondimenti sui container su tutti i nuovi cluster per impostazione predefinita o su un singolo cluster durante la sua creazione.

Attivazione di Approfondimenti sui container su tutti i nuovi cluster per impostazione predefinita

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel riquadro di navigazione, scegli Account Settings (Impostazioni account).
3. Scegli Aggiorna.
4. Per utilizzare CloudWatch Container Insights come impostazione predefinita per i cluster, in CloudWatchContainer Insights, seleziona o CloudWatch deseleziona Container Insights.
5. Seleziona Salvataggio delle modifiche.

Se non hai utilizzato la procedura precedente per abilitare Container Insights su tutti i nuovi cluster per impostazione predefinita, utilizza le fasi seguenti per creare un cluster con Container Insights abilitato.

Creazione di un cluster con Approfondimenti sui container attivato

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
4. In Cluster configuration (Configurazione del cluster), inserisci un nome univoco in Cluster name (Nome del cluster).

   Il nome può contenere fino a 255 lettere (maiuscole e minuscole), numeri e trattini.
5. Per attivare Approfondimenti sui container, espandi Monitoraggio e poi attiva Usa Approfondimenti sui container (Usa Approfondimenti sui container).

A questo punto puoi creare definizioni di attività, eseguire le attività e avviare i servizi nel cluster. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Creazione di una definizione di processo

- Esecuzione delle attività
- Creazione di un servizio

Configurazione di Container Insights su nuovi cluster Amazon ECS utilizzando AWS CLI

Per abilitare Container Insights su tutti i nuovi cluster per impostazione predefinita, immetti il comando seguente.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

Se non hai utilizzato il comando precedente per abilitare Container Insights su tutti i nuovi cluster per impostazione predefinita, puoi immettere il comando seguente per creare un cluster con Container Insights abilitato. Per il funzionamento del comando seguente, è necessario eseguire la versione 1.16.200 o successiva di AWS CLI .

```
aws ecs create-cluster --cluster-name myCICluster --settings
 "name=containerInsights,value=enabled"
```

Disabilitazione di Container Insights su cluster Amazon ECS

Per disabilitare Container Insights su un cluster Amazon ECS esistente, immetti il comando seguente.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=disabled
```

Configurazione di Container Insights su Amazon ECS utilizzando AWS Distro per OpenTelemetry

Utilizza questa sezione se desideri utilizzare AWS Distro OpenTelemetry per configurare CloudWatch Container Insights su un cluster Amazon ECS. Per ulteriori informazioni su AWS Distro for Open Telemetry, consulta Distro for.AWS OpenTelemetry

In queste fasi si presuppone che tu disponga già di un cluster su Amazon ECS. Per ulteriori informazioni sull'utilizzo di AWS Distro for Open Telemetry con Amazon ECS e sulla configurazione di un cluster Amazon ECS per questo scopo, consulta Configurazione di AWS Distro for OpenTelemetry Collector in Amazon Elastic Container Service.

Fase 1: creazione di un ruolo dell'attività

Il primo passaggio consiste nella creazione di un ruolo di attività nel cluster che verrà utilizzato da Collector. AWS OpenTelemetry

Creare un ruolo di attività per AWS Distro per OpenTelemetry

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Policy e Crea policy.
3. Scegli la scheda JSON e copia la policy seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Scegli Verifica policy.
5. Nel campo Name (Nome), inserisci **`AWSDistroOpenTelemetryPolicy`**, quindi scegli Create policy (Crea policy).
6. Nel pannello di navigazione sinistro, scegli Roles (Ruoli), quindi Create role (Crea ruolo).
7. Nell'elenco dei servizi, scegli Elastic Container Service.
8. In basso nella pagina, scegli Elastic Container Service Task (Attività di Elastic Container Service) e quindi Next: Permissions (Avanti: autorizzazioni).
9. Nell'elenco delle politiche, cerca AWSDistroOpenTelemetryPolicy.
10. Seleziona la casella di controllo accanto a AWSDistroOpenTelemetryPolicy.
11. Scegli Next: Tags (Successivo: Tag), quindi Next: Review (Successivo: Verifica).
12. In Role name (Nome ruolo) inserisci **`AWSOpenTelemetryTaskRole`** e quindi seleziona Create role (Crea ruolo).

Fase 2: creazione di un ruolo di esecuzione dell'attività

Il passaggio successivo consiste nella creazione di un ruolo di esecuzione delle attività per il AWS OpenTelemetry Collector.

Creare un ruolo di esecuzione delle attività per AWS Distro for OpenTelemetry

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione sinistro, scegli Roles (Ruoli), quindi Create role (Crea ruolo).
3. Nell'elenco dei servizi, scegli Elastic Container Service.
4. In basso nella pagina, scegli Elastic Container Service Task (Attività di Elastic Container Service) e quindi Next: Permissions (Avanti: autorizzazioni).
5. Nell'elenco delle politiche, cerca AmazonECS e seleziona la casella di controllo accanto a TaskExecutionRolePolicy AmazonECS. TaskExecutionRolePolicy
6. Nell'elenco delle politiche, cerca CloudWatchLogsFullAccesse seleziona la casella di controllo accanto a. CloudWatchLogsFullAccess
7. Nell'elenco delle politiche, cerca AmazonSSM ReadOnlyAccess e seleziona la casella di controllo accanto a AmazonSSM. ReadOnlyAccess
8. Scegli Next: Tags (Successivo: Tag), quindi Next: Review (Successivo: Verifica).
9. In Role name (Nome ruolo) inserisci **`AWSOpenTelemetryTaskExecutionRole`** e quindi seleziona Create role (Crea ruolo).

Fase 3: creazione di una definizione di attività

La fase successiva consiste nella creazione di una definizione di attività.

Per creare una definizione di attività per Distro for AWS OpenTelemetry

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di attività).
3. Scegli Create new task definitions (Crea nuova definizione di attività) e Create new Task Definition (Crea nuova definizione attività).
4. Per Task definition family (Famiglia della definizione di attività) specifica un nome univoco per la definizione di attività.
5. Configura i tuoi container, quindi scegli Successivo.

6. In Parametri e registrazione, seleziona Usa raccolta di parametri.
7. Seleziona Successivo.
8. Seleziona Crea.

Per ulteriori informazioni sull'utilizzo del AWS OpenTelemetry collector con Amazon ECS, consulta Configurazione di AWS Distro for OpenTelemetry Collector in Amazon Elastic Container Service.

Fase 4: esecuzione dell'attività

La fase finale consiste nell'esecuzione dell'attività creata.

Per eseguire l'attività per Distro per AWS OpenTelemetry

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione sinistro, scegli Task Definitions (Definizioni di attività), quindi seleziona l'attività appena creata.
3. Scegli Operazioni, Implementa, Esegui attività.
4. Scegli Deploy (Implementa), Run task (Esegui processo).
5. Nella sezione Opzioni di calcolo, da Cluster esistente, scegli il cluster.
6. Scegli Crea.
7. Successivamente, puoi verificare le nuove metriche nella CloudWatch console.
8. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
9. Nel pannello di navigazione a sinistra scegli Metrics (Parametri).

   Dovresti vedere uno spazio dei nomi ECS/ ContainerInsights. Scegli lo spazio dei nomi, dovresti vedere otto parametri.

Implementazione dell' CloudWatch agente per raccogliere parametri a livello di istanza EC2 su Amazon ECS

Per distribuire l' CloudWatch agente per raccogliere parametri a livello di istanza dai cluster Amazon ECS ospitati su un'istanza EC2, utilizza una configurazione di avvio rapido con una configurazione predefinita o installa l'agente manualmente per poterlo personalizzare.

Entrambi i metodi richiedono che tu disponga già di almeno un cluster Amazon ECS distribuito con un tipo di avvio EC2 e che il contenitore dell' CloudWatch agente abbia accesso all'Amazon EC2

Instance Metadata Service (IMDS). Per ulteriori informazioni su IMDS, consulta Metadati dell'istanza e dati utente.

Questi metodi presuppongono inoltre che tu abbia installato il. AWS CLI Inoltre, per eseguire i comandi nelle seguenti procedure, devi accedere a un account o ruolo con le politiche IAM FullAccess e FullAccessAmazoneCS_.

Argomenti

- Configurazione rapida tramite AWS CloudFormation
- Configurazione manuale e personalizzata

Configurazione rapida tramite AWS CloudFormation

Per utilizzare la configurazione rapida, inserisci il seguente comando da utilizzare per installare l'agente. AWS CloudFormation Sostituisci *cluster-name* e *cluster-region* con il nome e la regione del cluster Amazon ECS.

Questo comando crea i ruoli IAM CWAGentecs e CWagenTecs TaskRole. ExecutionRole Se questi ruoli esistono già nell'account, utilizza `ParameterKey=CreateIAMRoles,ParameterValue=False` anziché `ParameterKey=CreateIAMRoles,ParameterValue=True` quando immetti il comando. In caso contrario, il comando avrà esito negativo.

```
ClusterName=cluster-name
Region=cluster-region
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

(Alternativa) Utilizzo dei propri ruoli IAM

Se desideri utilizzare il tuo ruolo di attività ECS personalizzato e il tuo ruolo di esecuzione dell'attività ECS anziché i ExecutionRole ruoli CWagenTecs e CWagenTecs, assicurati innanzitutto TaskRole che il ruolo da utilizzare come ruolo di attività ECS sia associato. CloudWatchAgentServerPolicy

Inoltre, assicurati che al ruolo da utilizzare come ruolo di esecuzione delle attività ECS siano allegate sia le politiche AmazonECS che quelle di CloudWatchAgentServerPolicyAmazonECS. TaskExecutionRolePolicy Quindi, immetti il comando seguente: Nel comando, sostituisci *task-role-arn*con l'ARN del tuo ruolo di attività ECS personalizzato e sostituiscilo con *execution-role-arn*l'ARN del tuo ruolo di esecuzione dell'attività ECS personalizzato.

```
ClusterName=cluster-name
Region=cluster-region
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=TaskRoleArn,ParameterValue=${TaskRoleArn} \
                 ParameterKey=ExecutionRoleArn,ParameterValue=${ExecutionRoleArn} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

Risoluzione dei problemi relativi alla configurazione rapida

Per verificare lo stato dello AWS CloudFormation stack, inserisci il seguente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stacks --stack-name CWAgentECS-$ClusterName-$Region --
region $Region
```

Se `StackStatus` è diverso da `CREATE_COMPLETE` o `CREATE_IN_PROGRESS`, controlla gli eventi di stack per trovare l'errore. Inserire il seguente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name CWAgentECS-$ClusterName-$Region
 --region $Region
```

Per verificare lo stato del servizio daemon `cwagent`, immetti il seguente comando. Nell'output, deve essere mostrato che `runningCount` è uguale a `desiredCount` nella sezione `deployment`. In caso contrario, controlla la sezione `failures` nell'output.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services cwagent-daemon-service --cluster $ClusterName --
region $Region
```

Puoi anche utilizzare la console CloudWatch Logs per controllare il registro dell'agente. Cerca il gruppo di log ecs-cwagent-daemon-service/ecs/.

Eliminazione dello stack per l'agente AWS CloudFormation CloudWatch

Se è necessario eliminare lo AWS CloudFormation stack, immettere il seguente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation delete-stack --stack-name CWAgentECS-${ClusterName}-${Region} --
region ${Region}
```

Configurazione manuale e personalizzata

Segui i passaggi in questa sezione per distribuire manualmente l' CloudWatch agente per raccogliere parametri a livello di istanza dai tuoi cluster Amazon ECS ospitati su istanze EC2.

Ruoli e policy IAM necessari

Sono richiesti due ruoli IAM. Se non esistono già è necessario crearli. Per ulteriori informazioni su questi ruoli, consulta Ruoli IAM per attività e Ruolo per l'esecuzione di attività Amazon ECS.

- Un ruolo di attività ECS, utilizzato dall'agente per pubblicare le metriche. CloudWatch Se questo ruolo esiste già, è necessario assicurarsi che ad esso sia collegata la policy `CloudWatchAgentServerPolicy`.
- Un ruolo di esecuzione delle attività ECS, utilizzato dall'agente Amazon ECS per avviare l' CloudWatch agente. Se questo ruolo esiste già, è necessario assicurarsi che le policy `AmazonECSTaskExecutionRolePolicy` e `CloudWatchAgentServerPolicy` siano collegate ad esso.

Se questi ruoli non sono ancora disponibili, puoi utilizzare i seguenti comandi per crearli e collegare le policy necessarie. Questo primo comando crea il ruolo attività ECS.

```
aws iam create-role --role-name CWAgentECSTaskRole \
```

```
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Dopo aver immesso il comando precedente, annota il valore dell'output `Arn` del comando come "»TaskRoleArn. Questo sarà richiesto in seguito durante la creazione della definizione di attività. Quindi immetti il seguente comando per collegare le policy necessarie.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSTaskRole
```

Questo comando successivo crea il ruolo per l'esecuzione di attività ECS.

```
aws iam create-role --role-name CWAgentECSExecutionRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Dopo aver immesso il comando precedente, annotate il valore `Arn` dell'output del comando come "ExecutionRoleArn». Questo sarà richiesto in seguito durante la creazione della definizione di attività. Quindi immetti i seguenti comandi per collegare le policy necessarie.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSExecutionRole

aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy \
    --role-name CWAgentECSExecutionRole
```

Creazione della definizione di attività e avvio del servizio Daemon

Crea una definizione di attività e usala per avviare l'agente come servizio daemon. CloudWatch Per creare la definizione di attività, immetti il seguente comando. Nelle prime righe, sostituisci i segnaposto con i valori effettivi per la distribuzione. *`logs-region è la regione`* in cui si trova CloudWatch Logs e *`cluster-region è la regione`* in cui si trova il cluster. *`task-role-arn`*è l'Arn del ruolo dell'attività ECS che stai utilizzando ed *`execution-role-arn`*è l'Arn del ruolo di esecuzione dell'attività ECS.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-json
```

Quindi esegui il seguente comando per avviare il servizio daemon. Sostituisci *cluster-name* e *cluster-region* con il nome e la regione del cluster Amazon ECS.

> **Important**
>
> Rimuovi tutte le strategie dei provider di capacità prima di eseguire questo comando. In caso contrario, il comando non funzionerà.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

Se viene visualizzato questo messaggio di errore, `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, un servizio daemon denominato `cwagent-daemon-service` è già stato creato. Elimina innanzitutto tale servizio, utilizzando il comando riportato di seguito come un esempio.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
```

```
    --region ${Region} \
    --force
```

(Facoltativo) Configurazione avanzata

Facoltativamente, puoi utilizzare SSM per specificare altre opzioni di configurazione per l' CloudWatchagente nei tuoi cluster Amazon ECS ospitati su istanze EC2. Le opzioni sono le seguenti:

- `metrics_collection_interval`— Con quale frequenza, in secondi, l'agente raccoglie i parametri. CloudWatch Il valore predefinito è 60. L'intervallo varia tra 1 e 172.000.
- `endpoint_override` (Facoltativo): specifica un endpoint diverso a cui inviare i log. Potrebbe essere richiesto se si sta pubblicando da un cluster in un VPC e si desidera inviare i dati di log a un endpoint VPC.

  Il valore di `endpoint_override` deve essere una stringa che è un URL.
- `force_flush_interval`: specifica in secondi la quantità massima di tempo in cui i log rimangono nel buffer di memoria prima di essere inviati al server. Indipendentemente dall'impostazione di questo campo, se la dimensione dei log nel buffer raggiunge 1 MB, i log vengono immediatamente inviati al server. Il valore di default è 5 secondi.
- `region`: per impostazione predefinita, l'agente pubblica i parametri nella stessa regione in cui si trova l'istanza di container Amazon ECS. Per sovrascriverla, è possibile specificare una regione diversa qui. Ad esempio, `"region" : "us-east-1"`

Di seguito è riportato un esempio di configurazione personalizzata:

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "ecs": {
                "metrics_collection_interval": 30
            }
        },
        "force_flush_interval": 5
    }
}
```

Per personalizzare la configurazione degli CloudWatch agenti nei contenitori Amazon ECS

1. Assicurati che la ReadOnlyAccess policy AmazonSSM sia associata al tuo ruolo di Amazon ECS Task Execution. A questo scopo, è possibile immettere il seguente comando. Questo esempio presuppone che il ruolo di Amazon ECS Task Execution sia CWagenTecs. ExecutionRole Se utilizzi un ruolo diverso, sostituire tale nome del ruolo nel comando seguente.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
AmazonSSMReadOnlyAccess \
        --role-name CWAgentECSExecutionRole
```

2. Creare il file di configurazione personalizzato simile all'esempio precedente. Assegnare a questo file il nome `/tmp/ecs-cwagent-daemon-config.json`.

3. Eseguire il seguente comando per inserire questa configurazione nel Parameter Store. Sostituire *`cluster-region`* con la regione del cluster Amazon ECS. Per eseguire questo comando, devi accedere a un utente o a un ruolo con la politica AmazonSSM. FullAccess

```
Region=cluster-region
aws ssm put-parameter \
    --name "ecs-cwagent-daemon-service" \
    --type "String" \
    --value "`cat /tmp/ecs-cwagent-daemon-config.json`" \
    --region $Region
```

4. Scaricare il file di definizione dell'attività in un file locale, ad esempio `/tmp/cwagent-ecs-instance-metric.json`

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/
cwagent-ecs-instance-metric/cwagent-ecs-instance-metric.json -o /tmp/cwagent-ecs-
instance-metric.json
```

5. Modificare il file di definizione dell'attività. Rimuovere la sezione seguente:

```
"environment": [
                {
                    "name": "USE_DEFAULT_CONFIG",
                    "value": "True"
                }
            ],
```

Sostituire tale sezione con la seguente:

```
"secrets": [
                {
                    "name": "CW_CONFIG_CONTENT",
                    "valueFrom": "ecs-cwagent-daemon-service"
                }
            ],
```

6. Riavviare l'agente come servizio daemon effettuando la seguente procedura:

   a. Esegui il comando seguente.

   ```
   TaskRoleArn=task-role-arn
   ExecutionRoleArn=execution-role-arn
   AWSLogsRegion=logs-region
   Region=cluster-region
   cat /tmp/cwagent-ecs-instance-metric.json \
       | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
   ${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
       | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-
   json
   ```

   b. Eseguire il comando seguente per avviare il servizio daemon. Sostituisci *cluster-name* e *cluster-region* con il nome e la regione del cluster Amazon ECS.

   ```
   ClusterName=cluster-name
   Region=cluster-region
   aws ecs create-service \
       --cluster ${ClusterName} \
       --service-name cwagent-daemon-service \
       --task-definition ecs-cwagent-daemon-service \
       --scheduling-strategy DAEMON \
       --region ${Region}
   ```

   Se viene visualizzato questo messaggio di errore, `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, un servizio daemon denominato `cwagent-daemon-service` è già stato creato. Elimina innanzitutto tale servizio, utilizzando il comando riportato di seguito come un esempio.

```
ClusterName=cluster-name
Region=Region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

Implementazione della AWS distribuzione per raccogliere parametri OpenTelemetry a livello di istanza EC2 sui cluster Amazon ECS

Utilizza i passaggi di questa sezione per utilizzare AWS Distro per raccogliere parametri OpenTelemetry a livello di istanza EC2 su un cluster Amazon ECS. Per ulteriori informazioni su Distro for, consulta Distro for. AWS OpenTelemetry AWS OpenTelemetry

In queste fasi si presuppone che tu disponga già di un cluster su Amazon ECS. Questo cluster deve essere implementato con il tipo di avvio EC2. Per ulteriori informazioni sull'utilizzo di AWS Distro for Open Telemetry con Amazon ECS e sulla configurazione di un cluster Amazon ECS per questo scopo, consulta Configurazione di AWS Distro for OpenTelemetry Collector nei parametri a livello di istanza di Amazon Elastic Container Service for ECS EC2.

Argomenti

- Configurazione rapida utilizzando AWS CloudFormation
- Configurazione manuale e personalizzata

Configurazione rapida utilizzando AWS CloudFormation

Scarica il file AWS CloudFormation modello per l'installazione di AWS Distro for OpenTelemetry collector for Amazon ECS su EC2. Esegui il seguente comando curl.

```
curl -O https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/ecs/aws-otel-ec2-instance-metrics-daemon-deployment-cfn.yaml
```

Dopo aver scaricato il file modello, aprilo e sostituisci *PATH_TO_ CloudFormation _TEMPLATE con il percorso in cui hai salvato il file modello*. Quindi esportate i seguenti parametri ed eseguite il AWS CloudFormation comando, come illustrato nel comando seguente.

- Cluster_Name: il nome del cluster Amazon ECS

- AWS_Region: la regione in cui verranno inviati i dati
- PATH_TO_ CloudFormation _TEMPLATE — Il percorso in cui è stato salvato il file modello. AWS CloudFormation
- comando: per consentire a AWS Distro for OpenTelemetry collector di raccogliere i parametri a livello di istanza per Amazon ECS su Amazon EC2, devi specificare questo parametro. `--config=/etc/ecs/otel-instance-metrics-config.yaml`

```
ClusterName=Cluster_Name
Region=AWS_Region
command=--config=/etc/ecs/otel-instance-metrics-config.yaml
aws cloudformation create-stack --stack-name AOCECS-${ClusterName}-${Region} \
--template-body file://PATH_TO_CloudFormation_TEMPLATE \
--parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
ParameterKey=CreateIAMRoles,ParameterValue=True \
ParameterKey=command,ParameterValue=${command} \
--capabilities CAPABILITY_NAMED_IAM \
--region ${Region}
```

Dopo aver eseguito questo comando, utilizza la console Amazon ECS per verificare se l'attività è in esecuzione.

Risoluzione dei problemi relativi alla configurazione rapida

Per verificare lo stato dello stack, inserisci il seguente comando. AWS CloudFormation

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack --stack-name AOCECS-$ClusterName-$Region --region
 $Region
```

Se il valore di `StackStatus` è diverso da `CREATE_COMPLETE` o `CREATE_IN_PROGRESS`, controlla gli eventi di stack per trovare l'errore. Inserire il seguente comando.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name AOCECS-$ClusterName-$Region --
region $Region
```

Per verificare lo stato del servizio daemon `AOCECS`, immetti il seguente comando. Nell'output, deve essere mostrato che `runningCount` è uguale a `desiredCount` nella sezione di implementazione. In caso contrario, controlla la sezione degli errori nell'output.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services AOCECS-daemon-service --cluster $ClusterName --
region $Region
```

Puoi anche utilizzare la console CloudWatch Logs per controllare il registro dell'agente. Cerca il gruppo di log /aws/ecs/containerinsights/ {} /performance. ClusterName

Configurazione manuale e personalizzata

Segui i passaggi in questa sezione per distribuire manualmente la AWS Distro e raccogliere parametri OpenTelemetry a livello di istanza dai cluster Amazon ECS ospitati su istanze Amazon EC2.

Fase 1: ruoli e policy necessari

Sono richiesti due ruoli IAM. Se non esistono già è necessario crearli. Per ulteriori informazioni su questi ruoli, consulta Creazione della policy IAM e Creazione del ruolo IAM.

Fase 2: creazione di una definizione dell'attività

Crea una definizione di attività e usala per avviare Distro for as a daemon service. AWS OpenTelemetry

Per utilizzare il modello di definizione dell'attività per creare la definizione dell'attività, segui le istruzioni in Creare una definizione di attività ECS EC2 per un'istanza EC2 con OTel Collector. AWS

Per utilizzare la console Amazon ECS per creare la definizione dell'attività, segui le istruzioni in Installa AWS OTel Collector creando la definizione delle attività tramite AWS console per i parametri delle istanze Amazon ECS EC2.

Fase 3: avvio del servizio daemon

Per avviare AWS Distro for OpenTelemetry as a daemon service, segui le istruzioni in Esegui la tua attività su Amazon Elastic Container Service (Amazon ECS) usando il servizio daemon.

(Facoltativo) Configurazione avanzata

Facoltativamente, puoi utilizzare SSM per specificare altre opzioni di configurazione per la AWS Distro per i OpenTelemetry tuoi cluster Amazon ECS ospitati su istanze Amazon EC2. Per ulteriori informazioni sulla creazione di un file di configurazione, consulta Configurazione personalizzata. OpenTelemetry Per ulteriori informazioni sulle opzioni che è possibile utilizzare nel file di configurazione, consulta AWS Container Insights Receiver.

Configurazione FireLens per l'invio di log a CloudWatch Logs

FireLens per Amazon ECS consente di utilizzare i parametri di definizione delle attività per indirizzare i log ad Amazon CloudWatch Logs per l'archiviazione e l'analisi dei log. FireLens funziona con Fluent Bit e Fluentd. Forniamo un'immagine AWS per Fluent Bit, oppure puoi usare la tua immagine Fluent Bit o Fluentd. La creazione di definizioni di attività Amazon ECS con una FireLens configurazione è supportata utilizzando AWS gli SDK e AWS CLI. AWS Management Console Per ulteriori informazioni sui CloudWatch log, consulta What is Logs? CloudWatch .

Ci sono considerazioni chiave quando si utilizza FireLens per Amazon ECS. Per ulteriori informazioni, consulta Considerazioni.

Per trovare le immagini AWS per Fluent Bit, consulta Uso dell'immagine AWS for Fluent Bit.

Per creare una definizione di attività che utilizza una FireLens configurazione, vedere Creazione di una definizione di attività che utilizza una FireLens configurazione.

Esempio

Il seguente esempio di definizione delle attività mostra come specificare una configurazione di log che inoltri i log a un gruppo di log CloudWatch Logs. Per ulteriori informazioni, consulta What Is Amazon CloudWatch Logs? nella Amazon CloudWatch Logs User Guide.

Nelle opzioni di configurazione del log, specifica il nome del gruppo di log e la regione in cui esiste. Per fare in modo che Fluent Bit crei il gruppo di log per tuo conto, specifica `"auto_create_group":"true"`. Puoi anche specificare l'ID dell'attività come prefisso del flusso di log utile durante l'operazione di filtro. Per ulteriori informazioni, consulta Fluent Bit Plugin for CloudWatch Logs.

```
{
 "family": "firelens-example-cloudwatch",
 "taskRoleArn": "arn:aws:iam::123456789012:role/ecs_task_iam_role",
 "containerDefinitions": [
```

```
  {
   "essential": true,
   "image": "906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-bit:latest",
   "name": "log_router",
   "firelensConfiguration": {
    "type": "fluentbit"
   },
   "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
     "awslogs-group": "firelens-container",
     "awslogs-region": "us-west-2",
     "awslogs-create-group": "true",
     "awslogs-stream-prefix": "firelens"
    }
   },
   "memoryReservation": 50
   },
   {
    "essential": true,
    "image": "nginx",
    "name": "app",
    "logConfiguration": {
     "logDriver":"awsfirelens",
     "options": {
     "Name": "cloudwatch",
     "region": "us-west-2",
     "log_key": "log",
                                "log_group_name": "/aws/ecs/containerinsights/
$(ecs_cluster)/application",
     "auto_create_group": "true",
     "log_stream_name": "$(ecs_task_id)"
    }
   },
   "memoryReservation": 100
  }
 ]
}
```

Configurazione di Container Insights su Amazon EKS e Kubernetes

Approfondimenti sui container è supportato nelle versioni 1.23 e successive di Amazon EKS. Il metodo di installazione rapida è supportato solo nelle versioni 1.24 e successive.

Il processo generale per la configurazione di Container Insights su Amazon EKS o Kubernetes è il seguente:

1. Verificare di disporre dei prerequisiti necessari.
2. Configura il componente aggiuntivo Amazon CloudWatch Observability EKS, l' CloudWatch agente o la AWS distribuzione OpenTelemetry sul tuo cluster a cui inviare i parametri. CloudWatch

   > **Note**
   >
   > Per utilizzare Container Insights con osservabilità migliorata per Amazon EKS, devi utilizzare il componente aggiuntivo Amazon CloudWatch Observability EKS o l'agente. CloudWatch Per ulteriori informazioni su questa versione di Approfondimenti sui container, consulta la pagina Approfondimenti sui container con osservabilità migliorata per Amazon EKS.
   > Per utilizzare Container Insights con Fargate, è necessario utilizzare AWS Distro for. OpenTelemetry Approfondimenti sui container con osservabilità migliorata per Amazon EKS non è supportato su Fargate.

   > **Note**
   >
   > Container Insights ora supporta i nodi di lavoro Windows in un cluster Amazon EKS. Container Insights con osservabilità migliorata per Amazon EKS è supportato anche su Windows. Per informazioni sull'attivazione di Container Insights su Windows, consulta CloudWatch Utilizzo dell'agente con l'osservabilità avanzata di Container Insights abilitata.

   Configura Fluent Bit o Fluentd per inviare i log ai Logs. CloudWatch (Questa opzione è abilitata per impostazione predefinita se installi il componente aggiuntivo Amazon CloudWatch Observability EKS.)

   Puoi eseguire questi passaggi contemporaneamente come parte della configurazione di avvio rapido se utilizzi l' CloudWatch agente o eseguirli separatamente.
3. (Facoltativo) Impostare la registrazione del piano di controllo Amazon EKS.
4. (Facoltativo) Configura l' CloudWatch agente come endpoint StatSD sul cluster a cui inviare le metriche StatsD. CloudWatch
5. (Facoltativo) Abilita i log di accesso di App Mesh Envoy.

Con la versione originale di Approfondimenti sui container, i parametri raccolti e i log importati vengono addebitati come parametri personalizzati. Con Approfondimenti sui container con osservabilità migliorata per Amazon EKS, i parametri e i log di Approfondimenti sui container vengono addebitati per osservazione anziché per parametro archiviato o log importato. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Argomenti

Verifica dei prerequisiti di

Prima di installare Container Insights su Amazon EKS o Kubernetes, verifica quanto segue. Questi prerequisiti si applicano indipendentemente dal fatto che tu stia utilizzando l' CloudWatch agente o AWS Distro OpenTelemetry per configurare Container Insights sui cluster Amazon EKS.

- Disponi di un cluster Amazon EKS o Kubernetes funzionale con nodi collegati in una delle regioni che supporta Container Insights per Amazon EKS e Kubernetes. Per l'elenco delle regioni supportate, consulta Container Insights.
- Disponi di `kubectl` installato e in esecuzione. Per ulteriori informazioni, consulta la pagina relativa all'installazione di `kubectl` nella Guida per l'utente di Amazon EKS.
- Se utilizzi Kubernetes in esecuzione AWS anziché Amazon EKS, sono necessari anche i seguenti prerequisiti:
  - Assicurati che il cluster Kubernetes abbia abilitato il controllo degli accessi basato su ruoli (RBAC). Per ulteriori informazioni, consulta Utilizzo dell'autorizzazione RBAC nella documentazione di riferimento di Kubernetes.
  - Il kubelet ha abilitato la modalità di autorizzazione Webhook. Per ulteriori informazioni, consulta Autenticazione/autorizzazione Kubelet nella documentazione di riferimento di Kubernetes.

È inoltre necessario concedere le autorizzazioni IAM per consentire ai nodi di lavoro Amazon EKS di inviare parametri e log. CloudWatch Ci sono due modi per effettuare questa operazione:

- Collega una policy al ruolo IAM dei nodi di lavoro. Questo metodo funziona sia per i cluster Amazon EKS che per altri cluster Kubernetes.
- Utilizza un ruolo IAM per gli account di servizio per il cluster e collega la policy a questo ruolo. Questo metodo funziona solo per i cluster Amazon EKS.

La prima opzione concede le autorizzazioni CloudWatch per l'intero nodo, mentre l'utilizzo di un ruolo IAM per l'account di servizio consente di CloudWatch accedere solo ai pod daemonset appropriati.

Collegamento di una policy al ruolo IAM dei nodi di lavoro

Attieniti alla seguente procedura per collegare la policy al ruolo IAM dei nodi di lavoro. Questa procedura funziona sia per i cluster Amazon EKS che per i cluster Kubernetes esterni ad Amazon EKS.

Per collegare la policy necessaria al ruolo IAM per i nodi worker

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Seleziona una delle istanze dei nodi worker e scegli il ruolo IAM nella descrizione.
3. Nella pagina del ruolo IAM, scegli Attach policies (Collega policy).
4. Nell'elenco delle politiche, seleziona la casella di controllo accanto a. CloudWatchAgentServerPolicy Se necessario, utilizzare la casella di ricerca per trovare questa policy.
5. Scegli Collega policy.

Se stai eseguendo un cluster Kubernetes all'esterno di Amazon EKS, è possibile che tu non disponga già di un ruolo IAM associato ai nodi worker. In questo caso, occorre innanzitutto collegare un ruolo IAM all'istanza e quindi aggiungere la policy come illustrato nelle fasi precedenti. Per ulteriori informazioni sul collegamento di un ruolo a un'istanza, consulta Collegamento di un ruolo IAM a un'istanza nella Guida per l'utente di Amazon EC2 per le istanze Windows.

Se stai eseguendo un cluster Kubernetes all'esterno di Amazon EKS e desideri raccogliere ID volume EBS nei parametri, devi aggiungere un'altra policy al ruolo IAM collegato all'istanza. Aggiungi quanto segue come una policy inline. Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente di IAM.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Utilizzo di un ruolo dell'account del servizio IAM

Questo metodo funziona solo nei cluster Amazon EKS.

Per concedere l'autorizzazione all' CloudWatch utilizzo del ruolo di un account di servizio IAM

1. Se non l'hai già fatto, abilita i ruoli IAM per gli account del servizio nel cluster. Per ulteriori informazioni, consulta Abilitazione dei ruoli IAM per gli account di servizio nel cluster.
2. Se non l'hai già fatto, configura l'account del servizio per utilizzare il ruolo IAM. Per ulteriori informazioni, consulta la pagina Configurazione di un account di servizio Kubernetes per assumere un ruolo IAM.

   Quando crei il ruolo, collega la policy CloudWatchAgentServerPolicyIAM al ruolo oltre alla policy che crei per il ruolo. Inoltre, l'account di servizio Kubernetes associato collegato a questo ruolo deve essere creato nello spazio dei `amazon-cloudwatch` nomi, dove i daemonset CloudWatch e Fluent Bit verranno distribuiti nei passaggi successivi
3. In caso contrario, collega il ruolo IAM a un account del servizio nel cluster. Per ulteriori informazioni, consulta la pagina Configurazione di un account di servizio Kubernetes per assumere un ruolo IAM.

CloudWatch Utilizzo dell'agente con l'osservabilità avanzata di Container Insights abilitata

Utilizza le istruzioni in una delle seguenti sezioni per configurare Container Insights su un cluster Amazon EKS o un cluster Kubernetes utilizzando l'agente. CloudWatch Le istruzioni per l'installazione rapida sono supportate solo nelle versioni 1.24 e successive di Amazon EKS.

> **Note**
>
> Puoi installare Approfondimenti sui container seguendo le istruzioni riportate in una delle sezioni successive. Non è necessario seguire tutti e tre i set di istruzioni.

Argomenti

Installa il componente aggiuntivo Amazon CloudWatch Observability EKS

Puoi utilizzare il componente aggiuntivo Amazon EKS per installare Approfondimenti sui container con osservabilità migliorata per Amazon EKS. Il componente aggiuntivo installa l' CloudWatch agente per inviare i parametri dell'infrastruttura dal cluster, installa Fluent Bit per inviare i log dei container e consente inoltre di inviare la telemetria delle prestazioni dell'applicazione. CloudWatch Application Signals

Quando utilizzi il componente aggiuntivo Amazon EKS versione 1.5.0 o successiva, Container Insights è abilitato su entrambi i nodi di lavoro Linux e Windows del cluster. Attualmente, Application Signals non è supportato su Windows in Amazon EKS.

Il componente aggiuntivo Amazon EKS non è supportato per i cluster che eseguono Kubernetes anziché Amazon EKS.

Per ulteriori informazioni sul componente aggiuntivo Amazon CloudWatch Observability EKS, consulta. Installa l' CloudWatch agente utilizzando il componente aggiuntivo Amazon CloudWatch Observability EKS

Per installare il componente aggiuntivo Amazon CloudWatch Observability EKS

1. Innanzitutto, configura le autorizzazioni necessarie collegando la policy CloudWatchAgentServerPolicyIAM ai nodi di lavoro. A questo scopo, immetti il comando seguente. *my-worker-node-role*Sostituiscila con il ruolo IAM utilizzato dai nodi di lavoro Kubernetes.

   ```
   aws iam attach-role-policy \
   --role-name my-worker-node-role \
   ```

```
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

2. Per installare il componente aggiuntivo, immetti il seguente comando:

```
aws eks create-addon --cluster-name my-cluster-name --addon-name amazon-cloudwatch-observability
```

Configurazione di Quick Start per Container Insights su Amazon EKS e Kubernetes

> **Important**
>
> Se stai installando Container Insights su un cluster Amazon EKS, ti consigliamo di utilizzare il componente aggiuntivo Amazon CloudWatch Observability EKS per l'installazione, anziché seguire le istruzioni in questa sezione. Inoltre, per recuperare reti di elaborazione accelerate, devi utilizzare il componente aggiuntivo Amazon CloudWatch Observability EKS. Per ulteriori informazioni e istruzioni, consulta Installa il componente aggiuntivo Amazon CloudWatch Observability EKS.

Per completare la configurazione di Container Insights, puoi seguire le istruzioni di avvio rapido in questa sezione. Se esegui l'installazione su un cluster Amazon EKS e segui le istruzioni riportate in questa sezione a partire dal 6 novembre 2023, sul cluster installerai Approfondimenti sui container con osservabilità migliorata per Amazon EKS.

> **Important**
>
> Prima di completare la procedura descritta in questa sezione, è necessario verificare i prerequisiti, incluse le autorizzazioni IAM. Per ulteriori informazioni, consulta la pagina Verifica dei prerequisiti di .

In alternativa, puoi seguire le istruzioni nelle due sezioni seguenti, Configura l' CloudWatch agente per raccogliere le metriche del cluster e Invia log a Logs CloudWatch . Queste sezioni forniscono ulteriori dettagli di configurazione su come l' CloudWatch agente funziona con Amazon EKS e Kubernetes, ma richiedono l'esecuzione di ulteriori passaggi di installazione.

Con la versione originale di Approfondimenti sui container, i parametri raccolti e i log importati vengono addebitati come parametri personalizzati. Con Approfondimenti sui container con

osservabilità migliorata per Amazon EKS, i parametri e i log di Approfondimenti sui container vengono addebitati per osservazione anziché per parametro archiviato o log importato. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatchdei prezzi di Amazon.

> **Note**
>
> Amazon ha ora lanciato Fluent Bit come soluzione di log predefinita per Container Insights con significativi miglioramenti delle prestazioni. Ti consigliamo di utilizzare Fluent Bit anziché Fluentd.

Avvio rapido con l'operatore dell' CloudWatch agente e Fluent Bit

Esistono due configurazioni per Fluent Bit: una versione ottimizzata e una versione che offre un'esperienza più simile a Fluentd. La configurazione Quick Start utilizza la versione ottimizzata. Per maggiori dettagli sulla configurazione compatibile con Fluentd, consulta Configura Fluent Bit come DaemonSet per inviare i log ai Logs CloudWatch .

L'operatore CloudWatch agente è un contenitore aggiuntivo che viene installato in un cluster Amazon EKS. È modellato sull' OpenTelemetry Operator for Kubernetes. L'operatore gestisce il ciclo di vita delle risorse Kubernetes in un cluster. Installa CloudWatch Agent, DCGM Exporter (NVIDIA) e AWS Neuron Monitor su un cluster Amazon EKS e li gestisce. Fluent Bit e CloudWatch Agent for Windows vengono installati direttamente in un cluster Amazon EKS senza che l'operatore li gestisca.

Per una soluzione di autorità di certificazione più sicura e ricca di funzionalità, l'operatore dell' CloudWatch agente richiede cert-manager, una soluzione ampiamente adottata per la gestione dei certificati TLS in Kubernetes. L'utilizzo di cert-manager semplifica il processo di ottenimento, rinnovo, gestione e utilizzo di questi certificati. Garantisce che i certificati siano validi e aggiornati e tenta di rinnovarli in un momento configurato prima della scadenza. cert-manager facilita anche l'emissione di certificati da una varietà di fonti supportate, tra cui Certificate Manager AWS Private Certificate Authority.

Per distribuire Container Insights utilizzando il Quick Start

1. Installa cert-manager se non è già installato nel cluster. Per ulteriori informazioni, consulta Installazione di cert-manager.
2. Installa le definizioni personalizzate delle risorse (CRD) inserendo il seguente comando.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl
 apply --server-side -f -
```

3. Installa l'operatore inserendo il seguente comando. Sostituisci *my-cluster-name* con il nome del tuo cluster Amazon EKS o Kubernetes e sostituiscilo *my-cluster-region* con il nome della regione in cui vengono pubblicati i log. Ti consigliamo di utilizzare la stessa regione in cui è distribuito il cluster per ridurre i costi di trasferimento dei AWS dati in uscita.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

   Ad esempio, per implementare Container Insights sul cluster denominato `MyCluster` e pubblicare i log e i parametri negli Stati Uniti occidentali (Oregon), immetti il comando seguente.

```
ClusterName='MyCluster'
RegionName='us-west-2'
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

Migrazione da Container Insights

Se hai già configurato Container Insights in un cluster Amazon EKS e desideri migrare a Container Insights con una migliore osservabilità per Amazon EKS, consulta Aggiornamento a Approfondimenti sui container con osservabilità migliorata per Amazon EKS

Eliminazione di Container Insights

Se desideri rimuovere Container Insights dopo aver utilizzato la configurazione di avvio rapido, inserisci i seguenti comandi.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
```

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Configura l' CloudWatch agente per raccogliere le metriche del cluster

> **Important**
>
> Se stai installando Container Insights su un cluster Amazon EKS, ti consigliamo di utilizzare il componente aggiuntivo Amazon CloudWatch Observability EKS per l'installazione, anziché seguire le istruzioni in questa sezione. Per ulteriori informazioni e istruzioni, consulta Installa il componente aggiuntivo Amazon CloudWatch Observability EKS.

Per configurare Container Insights per raccogliere i parametri, puoi seguire le fasi in Configurazione di Quick Start per Container Insights su Amazon EKS e Kubernetes oppure le fasi in questa sezione. Nei passaggi seguenti, configuri l' CloudWatch agente in modo che sia in grado di raccogliere metriche dai tuoi cluster.

Se esegui l'installazione su un cluster Amazon EKS e segui le istruzioni riportate in questa sezione a partire dal 6 novembre 2023, sul cluster installerai Approfondimenti sui container con osservabilità migliorata per Amazon EKS.

Fase 1: Creare un namespace per CloudWatch

Usa il passaggio seguente per creare uno spazio dei nomi Kubernetes richiesto. `amazon-cloudwatch` CloudWatch Puoi ignorare questa fase se questo spazio dei nomi è già stato creato.

Per creare uno spazio dei nomi per CloudWatch

- Inserire il seguente comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Fase 2: creazione di un account di servizio nel cluster

Utilizza il passaggio seguente per creare un account di servizio per l' CloudWatch agente, se non ne hai già uno.

Per creare un account di servizio per l' CloudWatch agente

- Inserire il seguente comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cwagent/cwagent-serviceaccount.yaml
```

Se non hai seguito i passaggi precedenti, ma disponi già di un account di servizio per l' CloudWatch agente che desideri utilizzare, devi assicurarti che abbia le seguenti regole. Inoltre, nelle restanti fasi dell'installazione di Container Insights devi utilizzare il nome di quell'account di servizio invece di `cloudwatch-agent`.

```
rules:
  - apiGroups: [""]
    resources: ["pods", "nodes", "endpoints"]
    verbs: ["list", "watch"]
  - apiGroups: [ "" ]
    resources: [ "services" ]
    verbs: [ "list", "watch" ]
  - apiGroups: ["apps"]
    resources: ["replicasets", "daemonsets", "deployments", "statefulsets"]
    verbs: ["list", "watch"]
  - apiGroups: ["batch"]
    resources: ["jobs"]
    verbs: ["list", "watch"]
  - apiGroups: [""]
    resources: ["nodes/proxy"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes/stats", "configmaps", "events"]
    verbs: ["create", "get"]
  - apiGroups: [""]
    resources: ["configmaps"]
    resourceNames: ["cwagent-clusterleader"]
    verbs: ["get","update"]
```

```
  - nonResourceURLs: ["/metrics"]
    verbs: ["get", "list", "watch"]
```

Passaggio 3: crea un account ConfigMap per l' CloudWatchagente

Utilizza i seguenti passaggi per creare un messaggio ConfigMap per l' CloudWatch agente.

Per creare un file ConfigMap per l' CloudWatch agente

1. Scaricate il file ConfigMap YAML sul vostro host `kubectl` client eseguendo il seguente comando:

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-configmap.yaml
   ```

2. Modifica il file YAML scaricato come segue:

   - cluster_name: nella sezione `kubernetes`, sostituisci `{{cluster_name}}` con il nome del cluster. Rimuovi i caratteri `{{}}`. In alternativa, se stai utilizzando un cluster Amazon EKS, puoi eliminare il campo `"cluster_name"` e il valore. In tal caso, l' CloudWatch agente rileva il nome del cluster dai tag Amazon EC2.

3. (Facoltativo) Apporta ulteriori modifiche in ConfigMap base ai tuoi requisiti di monitoraggio, come segue:

   - metrics_collection_interval: nella sezione `kubernetes`, puoi specificare ogni quanto l'agente raccoglie i parametri. Il valore predefinito è 60 secondi. L'intervallo di raccolta cadvisor predefinito in Kubelet è 15 secondi, quindi non impostare questo valore su meno di 15 secondi.
   - endpoint_override — Nella `logs` sezione, puoi specificare l'endpoint CloudWatch Logs se desideri sovrascrivere l'endpoint predefinito. Questo può essere necessario se stai pubblicando da un cluster in un VPC e desideri che i dati vadano a un endpoint VPC.
   - force_flush_interval — Nella **`logs`** sezione, puoi specificare l'intervallo per il raggruppamento in batch degli eventi di registro prima che vengano pubblicati nei registri. CloudWatch Il valore predefinito è 5 secondi.
   - region: per impostazione predefinita, l'agente pubblica i parametri nella regione in cui si trova il nodo worker. Per sostituire questa impostazione, è possibile aggiungere un campo `region` nella sezione `agent`, ad esempio `"region":"us-west-2"`.

- sezione statsd — Se desideri che l'agente CloudWatch Logs venga eseguito anche come listener StatsD in ogni nodo di lavoro del cluster, puoi aggiungere una `statsd` sezione alla `metrics` sezione, come nell'esempio seguente. Per informazioni sulle altre opzioni StatsD per questa sezione, consulta Recupero dei parametri personalizzati con StatsD .

```
"metrics": {
  "metrics_collected": {
    "statsd": {
      "service_address":":8125"
    }
  }
}
```

Di seguito un esempio completo della sezione JSON.

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "kubernetes": {
                "cluster_name": "MyCluster",
                "metrics_collection_interval": 60
            }
        },
        "force_flush_interval": 5,
        "endpoint_override": "logs.us-east-1.amazonaws.com"
    },
    "metrics": {
        "metrics_collected": {
            "statsd": {
                "service_address": ":8125"
            }
        }
    }
}
```

4. Crea il file ConfigMap nel cluster eseguendo il comando seguente.

```
kubectl apply -f cwagent-configmap.yaml
```

Fase 4: Implementare l' CloudWatch agente come DaemonSet

Per completare l'installazione dell' CloudWatch agente e iniziare a raccogliere le metriche dei container, procedi nel seguente modo.

Per distribuire l' CloudWatch agente come DaemonSet

1. • Se non desideri utilizzare StatsD sul cluster, immetti il comando seguente.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-
daemonset.yaml
```

   • Se desideri utilizzare StatsD, procedi nel modo seguente:

   a. Scarica il file DaemonSet YAML sull'host `kubectl` client eseguendo il comando seguente.

```
curl -O  https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-
mode/daemonset/container-insights-monitoring/cwagent/cwagent-daemonset.yaml
```

   b. Rimuovi il commento della sezione `port` nel file `cwagent-daemonset.yaml` come descritto di seguito:

```
ports:
  - containerPort: 8125
    hostPort: 8125
    protocol: UDP
```

   c. Distribuisci l' CloudWatch agente nel tuo cluster eseguendo il comando seguente.

```
kubectl apply -f cwagent-daemonset.yaml
```

   d. Distribuisci l' CloudWatch agente sui nodi Windows del cluster eseguendo il comando seguente. Il listener StatSD non è supportato dall' CloudWatch agente in Windows.

```
kubectl apply -f cwagent-daemonset-windows.yaml
```

2. Convalida che l'agente venga implementato eseguendo il seguente comando.

```
kubectl get pods -n amazon-cloudwatch
```

Al termine, l' CloudWatch agente crea un gruppo di log denominato `/aws/containerinsights/Cluster_Name/performance` e invia gli eventi del registro delle prestazioni a questo gruppo di log. Se lo configuri anche come listener StatsD, l'agente ascolta anche i parametri StatsD sulla porta 8125 con l'indirizzo IP del nodo dove è pianificato il pod dell'applicazione.

Risoluzione dei problemi

Se l'agente non viene implementato correttamente, prova quanto segue:

- Per ottenere l'elenco di pod esegui il seguente comando.

```
kubectl get pods -n amazon-cloudwatch
```

- Esegui il comando seguente e controlla gli eventi nella parte inferiore dell'output.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Esegui il comando seguente per controllare i log.

```
kubectl logs pod-name  -n amazon-cloudwatch
```

Utilizzo di AWS Distro per OpenTelemetry

Puoi configurare Container Insights per raccogliere metriche dai cluster Amazon EKS utilizzando AWS Distro for collector. OpenTelemetry Per ulteriori informazioni su Distro for, AWS consulta Distro for OpenTelemetry.AWS OpenTelemetry

> **Important**
>
> Se si installa utilizzando AWS Distro for OpenTelemetry, si installa Container Insights ma non si ottiene Container Insights con osservabilità migliorata per Amazon EKS. Non raccoglierai i parametri dettagliati supportati in Approfondimenti sui container con osservabilità migliorata per Amazon EKS.

La modalità di configurazione di Container Insights varia a seconda che il cluster sia ospitato su istanze Amazon EC2 o su AWS Fargate (Fargate).

Cluster Amazon EKS ospitati su Amazon EC2

Se non lo hai già fatto, assicurati di aver soddisfatto i prerequisiti inclusi i ruoli IAM necessari. Per ulteriori informazioni, consulta la pagina Verifica dei prerequisiti di .

Amazon fornisce un grafico Helm che puoi utilizzare per configurare il monitoraggio di Amazon Elastic Kubernetes Service su Amazon EC2. Questo monitoraggio utilizza AWS Distro for OpenTelemetry (ADOT) Collector per le metriche e Fluent Bit per i log. Pertanto, il grafico Helm è utile per i clienti che utilizzano Amazon EKS su Amazon EC2 e desiderano raccogliere metriche e log da inviare a Container Insights. CloudWatch Per ulteriori informazioni su questo grafico di Helm, consulta il grafico ADOT Helm per EKS sulle metriche e i log di EC2 su Amazon Container Insights. CloudWatch

In alternativa, puoi usare le istruzioni nel resto di questa sezione.

Innanzitutto, distribuisci AWS Distro for OpenTelemetry collector come file utilizzando il seguente comando. DaemonSet

```
curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/eks/otel-container-insights-infra.yaml |
kubectl apply -f -
```

Utilizza il comando seguente per confermare che il raccoglitore è in esecuzione.

```
kubectl get pods -l name=aws-otel-eks-ci -n aws-otel-eks
```

Se l'output di questo comando include più pod nello stato `Running`, il raccoglitore è in esecuzione e raccoglie parametri dal cluster. Il raccoglitore crea un gruppo di log denominato `aws/containerinsights/`*`cluster-name`*`/performance` e invia gli eventi di log delle prestazioni allo stesso.

Per informazioni su come visualizzare le metriche di Container Insights in, consulta. CloudWatch Visualizzazione dei parametri di Container Insights

AWS ha anche fornito la documentazione GitHub relativa a questo scenario. Se si desidera personalizzare i parametri e i log pubblicati da Container Insights, consulta la pagina https://aws-otel.github.io/docs/getting-started/container-insights/eks-infra.

Cluster Amazon EKS ospitati su Fargate

Per istruzioni su come configurare e distribuire un ADOT Collector per raccogliere i parametri di sistema dai carichi di lavoro distribuiti in un cluster Amazon EKS su Fargate e inviarli a Container CloudWatch Insights, consulta Container Insights EKS Fargate nella distribuzione per la documentazione. AWS OpenTelemetry

Invia log a Logs CloudWatch

Per inviare i log dai tuoi contenitori ad Amazon CloudWatch Logs, puoi utilizzare Fluent Bit o Fluentd. Per ulteriori informazioni, consulta Fluent Bit e Fluentd.

Se non stai già utilizzando Fluentd, ti consigliamo di usare Fluent Bit per i seguenti motivi:

- Fluent Bit ha un'impronta ridotta delle risorse ed è più efficiente in termini di risorse con l'utilizzo della memoria e della CPU rispetto a Fluentd. Per un confronto più dettagliato, consulta la pagina Confronto delle prestazioni di Fluent Bit e Fluentd.
- L'immagine Fluent Bit è sviluppata e gestita da. AWS Ciò offre AWS la possibilità di adottare le nuove funzionalità di immagine Fluent Bit e di rispondere ai problemi molto più rapidamente.

Argomenti

Confronto delle prestazioni di Fluent Bit e Fluentd

Le tabelle seguenti mostrano il vantaggio in termini di prestazioni che Fluent Bit ha rispetto a Fluentd negli utilizzi della memoria e della CPU. I seguenti numeri sono solo per riferimento e potrebbero cambiare a seconda dell'ambiente.

Log al secondo	Utilizzo della CPU di Fluentd	Utilizzo della CPU di Fluent Bit con configurazione compatibile con Fluentd	Utilizzo della CPU di Fluent Bit con configurazione ottimizzata
100	0,35 vCPU	0,02 vCPU	0,02 vCPU
1.000	0,32 vCPU	0,14 vCPU	0,11 vCPU
5.000	0,85 vCPU	0,48 vCPU	0,30 vCPU
10.000	0,94 vCPU	0,60 vCPU	0,39 vCPU

Log al secondo	Utilizzo della memoria di Fluentd	Utilizzo della memoria di Fluent Bit con configurazione compatibile con Fluentd	Utilizzo della memoria di Fluent Bit con configurazione ottimizzata
100	153 MB	46 MB	37 MB
1.000	270 MB	45 MB	40 MB
5.000	320 MB	55 MB	45 MB
10.000	375 MB	92 MB	75 MB

Configura Fluent Bit come DaemonSet per inviare i log ai Logs CloudWatch

Le seguenti sezioni aiutano a implementare Fluent Bit per inviare i log dai contenitori ai Logs. CloudWatch

Argomenti

- Differenze se stai già usando Fluentd
- Configurazione di Fluent Bit
- Supporto per log multi-linea

- Riduzione del volume di log da Fluent Bit (facoltativo)
- Risoluzione dei problemi
- Dashboard

Differenze se stai già usando Fluentd

Se state già utilizzando Fluentd per inviare i log dai contenitori ai CloudWatch log, leggete questa sezione per vedere le differenze tra Fluentd e Fluent Bit. Se non stai già utilizzando Fluentd con Container Insights, puoi passare a Configurazione di Fluent Bit.

Offriamo due configurazioni predefinite per Fluent Bit:

- Configurazione ottimizzata di Fluent Bit: una configurazione allineata con le best practice di Fluent Bit.
- Configurazione compatibile con Fluentd: una configurazione il più possibile allineata al comportamento di Fluentd.

Nell'elenco seguente vengono illustrate in dettaglio le differenze tra Fluentd e ciascuna configurazione di Fluent Bit.

- Differenze nei nomi dei flussi di log: se utilizzi la configurazione ottimizzata di Fluent Bit, i nomi dei flussi di log saranno diversi.

  In `/aws/containerinsights/Cluster_Name/application`

  - La configurazione ottimizzata di Fluent Bit invia i log a *`kubernetes-nodeName`*`-application.var.log.containers.`*`kubernetes-podName_kubernetes-namespace_kubernetes-container-name-kubernetes-containerID`*
  - Fluentd invia i log a *`kubernetes-podName_kubernetes-namespace_kubernetes-containerName_kubernetes-containerID`*

  In `/aws/containerinsights/Cluster_Name/host`

  - La configurazione ottimizzata di Fluent Bit invia i log a *`kubernetes-nodeName`*`.`*`host-log-file`*
  - Fluentd invia i log a *`host-log-file`*`-`*`Kubernetes-NodePrivateIp`*

  In `/aws/containerinsights/Cluster_Name/dataplane`

  - La configurazione ottimizzata di Fluent Bit invia i log a *kubernetes-nodeName*.*dataplaneServiceLog*
  - Fluentd invia i log a *dataplaneServiceLog*-*Kubernetes-nodeName*
- I file di log `kube-proxy` e `aws-node` che Container Insights scrive si trovano in posizioni diverse. Nella configurazione di Fluentd, sono in `/aws/containerinsights/Cluster_Name/application`. Nella configurazione ottimizzata di Fluent Bit, sono in `/aws/containerinsights/Cluster_Name/dataplane`.
- La maggior parte dei metadati ad esempio `pod_name` e `namespace_name` sono gli stessi in Fluent Bit e Fluentd, ma i seguenti sono diversi.
  - La configurazione ottimizzata Fluent Bit utilizza `docker_id` e Fluentd usa `Docker.container_id`.
  - Entrambe le configurazioni di Fluent Bit non utilizzano i seguenti metadati. Sono presenti solo in Fluentd: `container_image_id`, `master_url`, `namespace_id` e `namespace_labels`.

Configurazione di Fluent Bit

Per configurare Fluent Bit per raccogliere log dai container, puoi seguire le fasi in Configurazione di Quick Start per Container Insights su Amazon EKS e Kubernetes o seguire le fasi in questa sezione.

Con entrambi i metodi, il ruolo IAM collegato ai nodi del cluster deve disporre di autorizzazioni sufficienti. Per ulteriori informazioni sulle autorizzazioni necessarie per l'esecuzione di un cluster Amazon EKS, consulta Policy, ruoli e autorizzazioni IAM di Amazon EKS nella Guida per l'utente Amazon EKS.

Nei passaggi seguenti, configuri Fluent Bit come DaemOnset per inviare i log ai Logs. CloudWatch Una volta completata questa fase, Fluent Bit crea i seguenti gruppi di log se non esistono già.

> **Important**
>
> Se hai già configurato FluentD in Container Insights e FluentD non funziona come previsto (questo può succedere se usi `containerd` il runtime), devi disinstallarlo prima di installare DaemonSet Fluent Bit per evitare che FluentBit elabori i messaggi di registro degli errori FluentD. Altrimenti, è necessario disinstallare FluentD immediatamente dopo aver installato con successo Fluent Bit. La disinstallazione di Fluentd dopo l'installazione di Fluent Bit garantisce la continuità della registrazione durante questo processo di migrazione. È necessario solo uno tra Fluent Bit o FluentD per inviare i log ai Logs. CloudWatch

Nome del gruppo di log	Origine del log
/aws/containerinsights/ *Cluster_Name* /application	Tutti i file di log in /var/log/containers
/aws/containerinsights/ *Cluster_Name* /host	I log da /var/log/dmesg , /var/log/secure e /var/log/messages
/aws/containerinsights/ *Cluster_Name* /dataplane	I log in /var/log/journal per kubelet.service kubeproxy.service e docker.service .

Per installare Fluent Bit per inviare i log dai contenitori ai Logs CloudWatch

1. Se non hai già uno spazio dei nomi chiamato amazon-cloudwatch, creane uno inserendo il seguente comando:

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

2. Esegui il comando seguente per creare un ConfigMap nome cluster-info con il nome del cluster e la regione a cui inviare i log. Sostituisci *cluster-name* e *cluster-region* con il nome e la regione del cluster.

```
ClusterName=cluster-name
RegionName=cluster-region
FluentBitHttpPort='2020'
FluentBitReadFromHead='Off'
[[ ${FluentBitReadFromHead} = 'On' ]] && FluentBitReadFromTail='Off'||
 FluentBitReadFromTail='On'
[[ -z ${FluentBitHttpPort} ]] && FluentBitHttpServer='Off' ||
 FluentBitHttpServer='On'
kubectl create configmap fluent-bit-cluster-info \
--from-literal=cluster.name=${ClusterName} \
--from-literal=http.server=${FluentBitHttpServer} \
--from-literal=http.port=${FluentBitHttpPort} \
--from-literal=read.head=${FluentBitReadFromHead} \
--from-literal=read.tail=${FluentBitReadFromTail} \
```

```
--from-literal=logs.region=${RegionName} -n amazon-cloudwatch
```

In questo comando, il `FluentBitHttpServer` per il monitoraggio dei parametri dei plugin è attivo per impostazione predefinita. Per disattivarlo, cambia la terza riga del comando in `FluentBitHttpPort=''` (stringa vuota) nel comando.

Inoltre, per impostazione predefinita, Fluent Bit legge i file di log dalla coda e acquisisce solo i nuovi log dopo che è stato implementato. Se avere l'effetto contrario, imposta `FluentBitReadFromHead='On'` e raccoglierà tutti i log nel file system.

3. Scarica e implementa il daemonset Fluent BIt nel cluster eseguendo uno dei seguenti comandi.

   - Se desideri la configurazione ottimizzata di Fluent Bit per computer Linux, esegui questo comando.

     ```
     kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
     cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
     deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-
     bit.yaml
     ```

   - Se desideri la configurazione ottimizzata di Fluent Bit per computer Windows, esegui questo comando.

     ```
     kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
     cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
     deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-
     windows.yaml
     ```

   - Se utilizzi computer Linux e desideri una configurazione Fluent Bit più simile a Fluentd, esegui questo comando.

     ```
     kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
     cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
     deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-
     compatible.yaml
     ```

   Important

   La configurazione del daemset Fluent Bit di default imposta il livello di registro su INFO, il che può comportare costi di inserimento dei log più elevati. CloudWatch Se si desidera

ridurre il volume e i costi di importazione dei log, è possibile modificare il livello di registrazione in ERROR.
Per ulteriori informazioni sulla riduzione del volume di log, consulta Riduzione del volume di log da Fluent Bit (facoltativo)

4. Convalida l'implementazione immettendo il seguente comando. Ogni nodo deve avere un pod denominato fluent-bit-*.

```
kubectl get pods -n amazon-cloudwatch
```

Le fasi precedenti creano le seguenti risorse nel cluster:

- Un account di servizio denominato `Fluent-Bit` nello spazio dei nomi `amazon-cloudwatch`. Questo account di servizio viene utilizzato per eseguire il DaemonSet Fluent Bit. Per ulteriori informazioni, consulta Gestione degli account di servizio nella documentazione di riferimento di Kubernetes.
- Un ruolo cluster denominato `Fluent-Bit-role` nello spazio dei nomi `amazon-cloudwatch`. Questo ruolo di cluster concede le autorizzazioni `get`, `list` e `watch` sui log di pod all'account di servizio `Fluent-Bit`. Per ulteriori informazioni, consulta Panoramica delle API nella documentazione di riferimento di Kubernetes.
- Un nome nel namespace. ConfigMap `Fluent-Bit-config` `amazon-cloudwatch` ConfigMap Contiene la configurazione che deve essere utilizzata da Fluent Bit. Per ulteriori informazioni, consulta Configurare un pod per utilizzare a ConfigMap nella documentazione di Kubernetes Tasks.

Se si desidera verificare la configurazione di Fluent Bit, procedi come segue.

Verifica l'impostazione di Fluent Bit

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
3. Assicurati di trovarti nella regione in cui è stato implementato Fluent Bit.
4. Controlla l'elenco dei gruppi di log nella regione. Verrà visualizzato un codice analogo al seguente:

   - `/aws/containerinsights/Cluster_Name/application`

- `/aws/containerinsights/Cluster_Name/host`
- `/aws/containerinsights/Cluster_Name/dataplane`

5. Passa a uno di questi gruppi di log e controlla Last Event Time (Ora ultimo evento) per i flussi di log. Se è recente rispetto a quando è stato implementato Fluent Bit, l'installazione viene verificata.

   Potrebbe esserci un leggero ritardo nella creazione del gruppo di log `/dataplane`. Questo è normale in quanto questi gruppi di log vengono creati solo quando Fluent Bit inizia a inviare i log per quel gruppo di log.

Supporto per log multi-linea

Per informazioni su come utilizzare Fluent Bit con log multi-linea, consulta le sezioni della documentazione di Fluent Bit riportate di seguito.

- Multiline Parsing
- Multiline and Containers (v1.8)
- Multiline Core (v1.8)
- Always use multiline in the tail input

Riduzione del volume di log da Fluent Bit (facoltativo)

Per impostazione predefinita, inviamo i log delle applicazioni Fluent Bit e i metadati Kubernetes a. CloudWatch Se desideri ridurre il volume di dati a cui inviare CloudWatch, puoi impedire l'invio a una o entrambe queste fonti di dati. CloudWatch

Per arrestare i log dell'applicazione Fluent Bit, rimuovi la sezione seguente dal file `Fluent-Bit.yaml`.

```
[INPUT]
        Name                tail
        Tag                 application.*
        Path                /var/log/containers/fluent-bit*
        Parser              docker
        DB                  /fluent-bit/state/flb_log.db
        Mem_Buf_Limit       5MB
        Skip_Long_Lines     On
        Refresh_Interval    10
```

Per evitare che i metadati Kubernetes vengano aggiunti agli eventi di registro a cui vengono inviati CloudWatch, aggiungi i seguenti filtri alla sezione del `application-log.conf` file. `Fluent-Bit.yaml` Sostituisci <Metadata_1>e i campi simili con gli identificatori di metadati effettivi.

```
application-log.conf: |
    [FILTER]
        Name                nest
        Match               application.*
        Operation           lift
        Nested_under        kubernetes
        Add_prefix          Kube.

    [FILTER]
        Name                modify
        Match               application.*
        Remove              Kube.<Metadata_1>
        Remove              Kube.<Metadata_2>
        Remove              Kube.<Metadata_3>

    [FILTER]
        Name                nest
        Match               application.*
        Operation           nest
        Wildcard            Kube.*
        Nested_under        kubernetes
        Remove_prefix       Kube.
```

Risoluzione dei problemi

Se non visualizzi questi gruppi di log e stai controllando nella regione corretta, controllare i log per i pod daemonSet Fluent Bit per individuare l'errore.

Esegui il comando seguente e accertati che lo stato sia `Running`.

```
kubectl get pods -n amazon-cloudwatch
```

Se i log contengono errori correlati alle autorizzazioni IAM, controlla il ruolo IAM collegato ai nodi del cluster. Per ulteriori informazioni sulle autorizzazioni necessarie per l'esecuzione di un cluster Amazon EKS, consulta Policy, ruoli e autorizzazioni IAM di Amazon EKS nella Guida per l'utente Amazon EKS.

Se lo stato del pod è `CreateContainerConfigError`, ottieni l'errore preciso eseguendo il seguente comando.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Dashboard

Puoi creare un pannello di controllo per monitorare i parametri di ciascun plug-in in esecuzione. Puoi visualizzare i dati per i byte di input e output e per le velocità di elaborazione dei record, nonché gli errori di output e le percentuali di nuovi tentativi/fallimenti. Per visualizzare questi parametri, dovrai installare l' CloudWatch agente con la raccolta di metriche Prometheus per i cluster Amazon EKS e Kubernetes. Per ulteriori informazioni su come configurare il pannello di controllo, consulta Installa l' CloudWatch agente con la raccolta di metriche Prometheus sui cluster Amazon EKS e Kubernetes.

> **Note**
>
> Prima di configurare questo pannello di controllo, è necessario impostare Container Insights per i parametri Prometheus. Per ulteriori informazioni, consulta la pagina Monitoraggio dei parametri di Container Insights Prometheus.

Per creare un pannello di controllo per i parametri Prometheus di Fluent Bit

1. Crea variabili di ambiente, sostituendo i valori a destra nelle righe seguenti in modo che corrispondano all'implementazione.

   ```
   DASHBOARD_NAME=your_cw_dashboard_name
   REGION_NAME=your_metric_region_such_as_us-west-1
   CLUSTER_NAME=your_kubernetes_cluster_name
   ```

2. Crea il pannello di controllo eseguendo il comando seguente.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
   prometheus/sample_cloudwatch_dashboards/fluent-bit/cw_dashboard_fluent_bit.json \
   | sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
   | sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
   | xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --
   dashboard-body
   ```

(Facoltativo) Configura Fluentd per inviare log a Logs DaemonSet CloudWatch

> **Warning**
>
> Il supporto di Container Insights per Fluentd è ora in modalità di manutenzione, il che significa che non AWS fornirà ulteriori aggiornamenti per Fluentd e che prevediamo di renderlo obsoleto nelle prossime future. Inoltre, la configurazione corrente di Fluentd per Approfondimenti sui container utilizza una versione precedente dell'immagine `fluent/fluentd-kubernetes-daemonset:v1.10.3-debian-cloudwatch-1.0` di Fluentd che non ha gli ultimi miglioramenti e patch di sicurezza. Per l'ultima immagine di Fluentd supportata dalla comunità open source, vedi. fluentd-kubernetes-daemonset
> Ti consigliamo vivamente di migrare all'utilizzo FluentBit con Container Insights ogni volta che è possibile. L'utilizzo FluentBit come log forwarder per Container Insights offre significativi miglioramenti delle prestazioni.
> Per ulteriori informazioni, consulta Configura Fluent Bit come DaemonSet per inviare i log ai Logs CloudWatch e Differenze se stai già usando Fluentd.

Per configurare Fluentd per raccogliere log dai container, puoi seguire i passaggi riportati in Configurazione di Quick Start per Container Insights su Amazon EKS e Kubernetes o la procedura in questa sezione. Nei passaggi seguenti, si configura Fluentd per inviare i log DaemonSet ai log. CloudWatch Una volta completata questa fase, Fluentd crea i seguenti gruppi di log se non esistono già.

Nome del gruppo di log	Origine del log
`/aws/containerinsights/`*`Cluster_Name`*`/application`	Tutti i file di log in `/var/log/containers`
`/aws/containerinsights/`*`Cluster_Name`*`/host`	I log da `/var/log/dmesg` , `/var/log/secure` e `/var/log/messages`
`/aws/containerinsights/`*`Cluster_Name`*`/dataplane`	I log in `/var/log/journal` per `kubelet.service` `kubeproxy.service` e `docker.service` .

Fase 1: Creare uno spazio dei nomi per CloudWatch

Usa il passaggio seguente per creare uno spazio dei nomi Kubernetes richiesto. `amazon-cloudwatch` CloudWatch Puoi ignorare questa fase se questo spazio dei nomi è già stato creato.

Per creare uno spazio dei nomi per CloudWatch

- Inserire il seguente comando.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Fase 2: installazione di Fluentd

Avvia questo processo scaricando Fluentd. Al termine di queste fasi, l'implementazione crea le risorse seguenti nel cluster:

- Un account di servizio denominato `fluentd` nello spazio dei nomi `amazon-cloudwatch`. Questo account di servizio viene utilizzato per eseguire Fluentd. DaemonSet Per ulteriori informazioni, consulta Gestione degli account di servizio nella documentazione di riferimento di Kubernetes.
- Un ruolo cluster denominato `fluentd` nello spazio dei nomi `amazon-cloudwatch`. Questo ruolo di cluster concede le autorizzazioni `get`, `list` e `watch` sui log di pod all'account di servizio `fluentd`. Per ulteriori informazioni, consulta Panoramica delle API nella documentazione di riferimento di Kubernetes.
- Un ConfigMap nome `fluentd-config` nel namespace. `amazon-cloudwatch` ConfigMap Contiene la configurazione che deve essere utilizzata da Fluentd. Per ulteriori informazioni, consulta Configurare un pod per utilizzare a ConfigMap nella documentazione di Kubernetes Tasks.

Installazione di Fluentd

1. Crea un ConfigMap nome `cluster-info` con il nome del cluster e la AWS regione a cui verranno inviati i log. Esegui il comando seguente, aggiornando i segnaposto con i nomi del cluster e della regione.

```
kubectl create configmap cluster-info \
--from-literal=cluster.name=cluster_name \
```

```
--from-literal=logs.region=region_name -n amazon-cloudwatch
```

2. Scarica e distribuisci Fluentd DaemonSet nel cluster eseguendo il comando seguente. Accertarti di utilizzare l'immagine di container con l'architettura corretta. Il manifesto di esempio funziona solo su istanze x86 e inserirà `CrashLoopBackOff` se disponi di istanze avanzate Advanced RISC Machine (ARM) nel cluster. Il daemonSet Fluentd non dispone di un'immagine Docker multi-architettura ufficiale che consente di utilizzare un tag per più immagini sottostanti e consentire al runtime di container di estrarre quella giusta. L'immagine ARM di Fluentd utilizza un tag diverso con un suffisso `arm64`.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/fluentd/fluentd.yaml
```

> **Note**
>
> A causa di una recente modifica per ottimizzare la configurazione di Fluentd e ridurre al minimo l'impatto delle richieste API di Fluentd sugli endpoint dell'API Kubernetes, l'opzione “Watch” (Guarda) per i filtri Kubernetes è stata disabilitata per impostazione predefinita. Per maggiori dettagli, consulta _metadata_filter. fluent-plugin-kubernetes

3. Convalida l'implementazione eseguendo il seguente comando. Ogni nodo deve avere un pod denominato `fluentd-cloudwatch-*`.

```
kubectl get pods -n amazon-cloudwatch
```

Fase 3: verifica della configurazione Fluentd

Per verificare la configurazione di Fluentd, completa le seguenti operazioni.

Verifica della configurazione di Fluentd per Approfondimenti sui container

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log). Assicurati di trovarti nella Regione in cui è stato implementato Fluentd sui tuoi container.

   Nell'elenco dei gruppi di log nella regione, dovrebbe essere visualizzato quanto segue:

- /aws/containerinsights/*Cluster_Name*/application
- /aws/containerinsights/*Cluster_Name*/host
- /aws/containerinsights/*Cluster_Name*/dataplane

Se questi gruppi di log sono visualizzati, la configurazione di Fluentd è verificata.

Supporto per log multi-linea

Il 19 agosto 2019 abbiamo aggiunto il supporto per log multi-linea per i log raccolti da Fluentd.

Per impostazione predefinita, lo starter della voce del log multi-linea è qualsiasi carattere senza spazi vuoti. Questo significa che tutte le righe del log che iniziano con un carattere che non contiene spazi sono considerate come una nuova voce di log multi-linea.

Se i log dell'applicazione utilizzano uno starter multi-linea diverso, puoi supportarli apportando due modifiche al file `fluentd.yaml`.

Innanzitutto, escludili dal supporto multi-linea predefinito aggiungendo i nomi di percorso dei file di log a un campo `exclude_path` nella sezione `containers` di `fluentd.yaml`. Di seguito è riportato un esempio.

```
<source>
  @type tail
  @id in_tail_container_logs
  @label @containers
  path /var/log/containers/*.log
  exclude_path ["full_pathname_of_log_file*", "full_pathname_of_log_file2*"]
```

Quindi, aggiungi un blocco per i file di log al file `fluentd.yaml`. L'esempio seguente viene utilizzato per il file di registro dell' CloudWatch agente, che utilizza un'espressione regolare timestamp come starter multilinea. Puoi copiare questo blocco e aggiungerlo a `fluentd.yaml`. Modifica le righe indicate per riflettere il nome del file di log dell'applicazione e lo starter multi-linea che desideri utilizzare.

```
<source>
```

```
  @type tail
  @id in_tail_cwagent_logs
  @label @cwagentlogs
  path /var/log/containers/cloudwatch-agent*
  pos_file /var/log/cloudwatch-agent.log.pos
  tag *
  read_from_head true
 <parse>
    @type json
    time_format %Y-%m-%dT%H:%M:%S.%NZ
 </parse>
</source>
```

```
<label @cwagentlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_cwagent
  </filter>

  <filter **>
    @type record_transformer
    @id filter_cwagent_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
 </filter>

  <filter **>
    @type concat
    key log
    multiline_start_regexp /^\d{4}[-/]\d{1,2}[-/]\d{1,2}/
    separator ""
    flush_interval 5
    timeout_label @NORMAL
 </filter>

 <match **>
    @type relabel
    @label @NORMAL
 </match>
</label>
```

(Facoltativo) Riduzione del volume di log da Fluentd

Per impostazione predefinita, inviamo i registri delle applicazioni Fluentd e i metadati Kubernetes a. CloudWatch Se desideri ridurre il volume di dati a cui inviare CloudWatch, puoi interrompere l'invio a una o entrambe queste fonti di dati. CloudWatch

Per arrestare i log dell'applicazione Fluentd, rimuovi la sezione seguente dal file `fluentd.yaml`.

```
<source>
  @type tail
  @id in_tail_fluentd_logs
  @label @fluentdlogs
  path /var/log/containers/fluentd*
  pos_file /var/log/fluentd.log.pos
  tag *
  read_from_head true
   <parse>
   @type json
   time_format %Y-%m-%dT%H:%M:%S.%NZ
  </parse>
</source>

<label @fluentdlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_fluentd
  </filter>

  <filter **>
    @type record_transformer
    @id filter_fluentd_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
   </filter>

   <match **>
     @type relabel
     @label @NORMAL
   </match>
</label>
```

Per evitare che i metadati Kubernetes vengano aggiunti agli eventi di registro a cui vengono inviati CloudWatch, aggiungi una riga alla sezione del `record_transformer` file. `fluentd.yaml` Nell'origine di log in cui desideri rimuovere questi metadati, aggiungi la riga seguente.

```
remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
```

Ad esempio:

```
<filter **>
  @type record_transformer
  @id filter_containers_stream_transformer
  <record>
    stream_name ${tag_parts[3]}
  </record>
  remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
</filter>
```

Risoluzione dei problemi

Se non vedi questi gruppi di log e stai cercando nella regione corretta, controlla i log dei pod Fluentd per cercare l'errore DaemonSet .

Esegui il comando seguente e accertati che lo stato sia `Running`.

```
kubectl get pods -n amazon-cloudwatch
```

Nei risultati del comando precedente, annota il nome del pod che inizia con `fluentd-cloudwatch`. Utilizza questo nome del pod nel comando seguente.

```
kubectl logs pod_name -n amazon-cloudwatch
```

Se i log contengono errori correlati alle autorizzazioni IAM, controlla il ruolo IAM collegato ai nodi del cluster. Per ulteriori informazioni sulle autorizzazioni necessarie per l'esecuzione di un cluster Amazon EKS, consulta Policy, ruoli e autorizzazioni IAM di Amazon EKS nella Guida per l'utente Amazon EKS.

Se lo stato del pod è `CreateContainerConfigError`, ottieni l'errore preciso eseguendo il seguente comando.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Se lo stato del pod è `CrashLoopBackOff`, assicurati che l'architettura dell'immagine di container Fluentd sia la stessa del nodo quando è stato installato Fluentd. Se il cluster ha entrambi i nodi x86 e ARM64, puoi utilizzare un'etichetta kubernetes.io/arch per posizionare le immagini sul nodo corretto. Per ulteriori informazioni, consulta kuberntes.io/arch.

(Facoltativo) Impostare la registrazione del piano di controllo Amazon EKS

Se utilizzi Amazon EKS, puoi facoltativamente abilitare la registrazione del piano di controllo di Amazon EKS, per fornire log di audit e diagnostica direttamente dal piano di controllo di Amazon EKS a Logs. CloudWatch Per ulteriori informazioni, consulta Registrazione del piano di controllo Amazon EKS.

(Facoltativo) Abilitazione di log di accesso di App Mesh Envoy

Puoi configurare Container Insights Fluentd per inviare i log di accesso di App Mesh Envoy a Logs. CloudWatch Per ulteriori informazioni, consulta la pagina Registrazione.

Per inviare i log di accesso di Envoy a Logs CloudWatch

1. Configura Fluentd nel cluster. Per ulteriori informazioni, consulta la pagina (Facoltativo) Configura Fluentd per inviare log a Logs DaemonSet CloudWatch .
2. Configurare i log di accesso di Envoy per i nodi virtuali. Per istruzioni, consulta Registrazione. Assicurarti di configurare il percorso di log in modo che **`/dev/stdout`** si trovi in ogni nodo virtuale.

Al termine, i registri di accesso dell'inviato vengono inviati al gruppo di log `/aws/containerinsights/Cluster_Name/application`.

(Facoltativo) Abilitazione di funzione Use_Kubelet per cluster di grandi dimensioni

Per impostazione predefinita, la funzionalità Use_Kubelet è disabilitata nel plug-in Kubernetes. FluentBit L'abilitazione di questa funzione può ridurre il traffico verso il server API e mitigare il problema che il server API è un collo di bottiglia. Ti consigliamo di abilitare questa funzione per cluster di grandi dimensioni.

Per abilitare Use_Kubelet, aggiungere innanzitutto i nodi e i nodi/autorizzazioni proxy alla configurazione ClusterRole.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: fluent-bit-role
rules:
  - nonResourceURLs:
      - /metrics
    verbs:
      - get
  - apiGroups: [""]
    resources:
      - namespaces
      - pods
      - pods/logs
      - nodes
      - nodes/proxy
    verbs: ["get", "list", "watch"]
```

Nella DaemonSet configurazione, questa funzionalità richiede l'accesso alla rete host. La versione dell'immagine per `amazon/aws-for-fluent-bit` dovrebbe 2.12.0 o successiva, oppure la versione dell'immagine bit fluent dovrebbe essere 1.7.2 o successiva.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluent-bit
  namespace: amazon-cloudwatch
  labels:
    k8s-app: fluent-bit
    version: v1
    kubernetes.io/cluster-service: "true"
spec:
  selector:
    matchLabels:
      k8s-app: fluent-bit
  template:
    metadata:
      labels:
        k8s-app: fluent-bit
        version: v1
        kubernetes.io/cluster-service: "true"
    spec:
```

```
      containers:
      - name: fluent-bit
        image: amazon/aws-for-fluent-bit:2.19.0
        imagePullPolicy: Always
        env:
            - name: AWS_REGION
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: logs.region
            - name: CLUSTER_NAME
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: cluster.name
            - name: HTTP_SERVER
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: http.server
            - name: HTTP_PORT
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: http.port
            - name: READ_FROM_HEAD
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: read.head
            - name: READ_FROM_TAIL
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: read.tail
            - name: HOST_NAME
              valueFrom:
                fieldRef:
                  fieldPath: spec.nodeName
            - name: HOSTNAME
              valueFrom:
                fieldRef:
                  apiVersion: v1
                  fieldPath: metadata.name
```

```
            - name: CI_VERSION
              value: "k8s/1.3.8"
        resources:
            limits:
              memory: 200Mi
            requests:
              cpu: 500m
              memory: 100Mi
        volumeMounts:
        # Please don't change below read-only permissions
        - name: fluentbitstate
          mountPath: /var/fluent-bit/state
        - name: varlog
          mountPath: /var/log
          readOnly: true
        - name: varlibdockercontainers
          mountPath: /var/lib/docker/containers
          readOnly: true
        - name: fluent-bit-config
          mountPath: /fluent-bit/etc/
        - name: runlogjournal
          mountPath: /run/log/journal
          readOnly: true
        - name: dmesg
          mountPath: /var/log/dmesg
          readOnly: true
      terminationGracePeriodSeconds: 10
      hostNetwork: true
      dnsPolicy: ClusterFirstWithHostNet
      volumes:
      - name: fluentbitstate
        hostPath:
          path: /var/fluent-bit/state
      - name: varlog
        hostPath:
          path: /var/log
      - name: varlibdockercontainers
        hostPath:
          path: /var/lib/docker/containers
      - name: fluent-bit-config
        configMap:
          name: fluent-bit-config
      - name: runlogjournal
        hostPath:
```

```
          path: /run/log/journal
      - name: dmesg
        hostPath:
          path: /var/log/dmesg
      serviceAccountName: fluent-bit
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
        effect: NoSchedule
      - operator: "Exists"
        effect: "NoExecute"
      - operator: "Exists"
        effect: "NoSchedule"
```

La configurazione del plug-in Kubernetes dovrebbe essere simile alla seguente:

```
[FILTER]
        Name                kubernetes
        Match               application.*
        Kube_URL            https://kubernetes.default.svc:443
        Kube_Tag_Prefix     application.var.log.containers.
        Merge_Log           On
        Merge_Log_Key       log_processed
        K8S-Logging.Parser  On
        K8S-Logging.Exclude Off
        Labels              Off
        Annotations         Off
        Use_Kubelet         On
        Kubelet_Port        10250
        Buffer_Size         0
```

Aggiornamento o eliminazione di Container Insights su Amazon EKS e Kubernetes

Utilizza i passaggi descritti in queste sezioni per aggiornare l'immagine del contenitore dell' CloudWatch agente o per rimuovere Container Insights da un cluster Amazon EKS o Kubernetes.

Argomenti

- Aggiornamento a Approfondimenti sui container con osservabilità migliorata per Amazon EKS
- Aggiornamento dell'immagine del contenitore CloudWatch dell'agente
- Eliminazione dell' CloudWatch agente e di Fluent Bit for Container Insights

Aggiornamento a Approfondimenti sui container con osservabilità migliorata per Amazon EKS

> **Important**
>
> Se stai aggiornando o installando Container Insights su un cluster Amazon EKS, ti consigliamo di utilizzare il componente aggiuntivo Amazon CloudWatch Observability EKS per l'installazione, anziché seguire le istruzioni in questa sezione. Inoltre, per recuperare i parametri di elaborazione accelerata, devi utilizzare il componente aggiuntivo Amazon CloudWatch Observability EKS. Per ulteriori informazioni e istruzioni, consulta Installa il componente aggiuntivo Amazon CloudWatch Observability EKS.

Approfondimenti sui container con osservabilità migliorata per Amazon EKS è la versione più recente di Approfondimenti sui container. Raccoglie parametri dettagliati dai cluster che eseguono Amazon EKS e offre pannelli di controllo accurati e immediatamente utilizzabili per approfondire la telemetria delle applicazioni e dell'infrastruttura. Per ulteriori informazioni su questa versione di Approfondimenti sui container, consulta la pagina Approfondimenti sui container con osservabilità migliorata per Amazon EKS.

Se hai installato la versione originale di Approfondimenti sui container su un cluster Amazon EKS e desideri aggiornarla alla versione più recente con osservabilità migliorata, segui le istruzioni riportate in questa sezione.

> **Important**
>
> Prima di completare i passaggi di questa sezione, devi aver verificato i prerequisiti, incluso cert-manager. Per ulteriori informazioni, consulta Avvio rapido con l'operatore dell' CloudWatch agente e Fluent Bit.

Aggiornamento di un cluster Amazon EKS ad Approfondimenti sui container con osservabilità migliorata per Amazon EKS

1. Installa l'operatore dell' CloudWatch agente inserendo il seguente comando. Sostituisci *`my-cluster-name`*con il nome del tuo cluster Amazon EKS o Kubernetes e sostituiscilo *`my-cluster-region`*con il nome della regione in cui vengono pubblicati i log. Ti consigliamo di utilizzare la stessa regione in cui è distribuito il cluster per ridurre i costi di trasferimento dei AWS dati in uscita.

```
ClusterName=my-cluster-name
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 apply -f -
```

Se si nota un errore causato da un conflitto di risorse, è probabile che sul cluster siano già ClusterRoleBinding installati l' CloudWatch agente e Fluent Bit con i relativi componenti associati, ad esempio ServiceAccount, il ClusterRole e il. Quando l'operatore dell' CloudWatch agente tenta di installare l' CloudWatch agente e i componenti associati, se rileva modifiche nei contenuti, per impostazione predefinita fallisce l'installazione o l'aggiornamento per evitare di sovrascrivere lo stato delle risorse sul cluster. Si consiglia di eliminare qualsiasi CloudWatch agente esistente con la configurazione di Container Insights precedentemente installato sul cluster e quindi di installare l' CloudWatch operatore dell'agente.

2. (Facoltativo) Per applicare una configurazione Fluent Bit personalizzata esistente, è necessario aggiornare la configmap associata al daemonset Fluent Bit. L'operatore CloudWatch agente fornisce una configurazione predefinita per Fluent Bit ed è possibile sovrascrivere o modificare la configurazione predefinita in base alle esigenze. Per applicare una configurazione personalizzata, segui questi passaggi.

   a. Apri la configurazione esistente immettendo il seguente comando.

   ```
   kubectl edit cm fluent-bit-config -n amazon-cloudwatch
   ```

   b. Apporta le modifiche al file, quindi accedi `:wq` per salvare il file ed esci dalla modalità di modifica.

   c. Riavvia Fluent Bit inserendo il seguente comando.

   ```
   kubectl rollout restart fluent-bit -n amazon-cloudwatch
   ```

Aggiornamento dell'immagine del contenitore CloudWatch dell'agente

> **Important**
>
> Se stai aggiornando o installando Container Insights su un cluster Amazon EKS, ti consigliamo di utilizzare il componente aggiuntivo Amazon CloudWatch Observability EKS

per l'installazione, anziché seguire le istruzioni in questa sezione. Inoltre, per recuperare i parametri di elaborazione accelerata, devi utilizzare il componente aggiuntivo Amazon CloudWatch Observability EKS o l'operatore dell'agente. CloudWatch Per ulteriori informazioni e istruzioni, consulta Installa il componente aggiuntivo Amazon CloudWatch Observability EKS.

Se devi aggiornare l'immagine del container alla versione più recente, utilizza le fasi in questa sezione.

Per aggiornare l'immagine del container

1. Verifica se la `amazoncloudwatchagent` Customer Resource Definition (CRD) esiste già inserendo il seguente comando.

   ```
   kubectl get crds amazoncloudwatchagents.cloudwatch.aws.amazon.com -n amazon-cloudwatch
   ```

   Se questo comando restituisce un errore relativo alla mancanza del CRD, nel cluster non è configurato Container Insights con osservabilità avanzata per Amazon EKS con l' CloudWatch operatore agente. In questo caso, consulta Aggiornamento a Approfondimenti sui container con osservabilità migliorata per Amazon EKS.
2. Applica il file `cwagent-version.yaml` più recente immettendo il comando seguente.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/main/k8s-quickstart/cwagent-version.yaml | kubectl apply -f -
   ```

Eliminazione dell' CloudWatch agente e di Fluent Bit for Container Insights

Se hai installato Container Insights utilizzando il componente aggiuntivo CloudWatch Observability per Amazon EKS, puoi eliminare Container Insights e l' CloudWatch agente inserendo il seguente comando:

> **Note**
>
> Il componente aggiuntivo Amazon EKS ora supporta Container Insights sui nodi di lavoro Windows. Se elimini il componente aggiuntivo Amazon EKS, viene eliminato anche Container Insights for Windows.

```
aws eks delete-addon –cluster-name my-cluster –addon-name amazon-cloudwatch-
observability
```

Altrimenti, per eliminare tutte le risorse relative all' CloudWatch agente e a Fluent Bit, inserisci il seguente comando. In questo comando, *My_Cluster_Name è il nome del tuo cluster* Amazon EKS o Kubernetes e *My_Region è il nome della regione* in cui vengono pubblicati i log.

```
ClusterName=My_Cluster_Name
RegionName=My-Region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Visualizzazione dei parametri di Container Insights

Dopo aver configurato Container Insights e aver raccolto le metriche, puoi visualizzarle nella console. CloudWatch

Per visualizzare i parametri Container Insights nel pannello di controllo, è necessario completare la configurazione di Container Insights. Per ulteriori informazioni, consulta la pagina Configurazione di Container Insights.

In questa procedura viene descritto come visualizzare i parametri generati automaticamente da Container Insights dai dati di log raccolti. Il resto di questa sezione spiega come approfondire ulteriormente i dati e utilizzare CloudWatch Logs Insights per visualizzare più metriche a più livelli di granularità.

Per visualizzare i parametri di Container Insights

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nel pannello di navigazione scegli Approfondimenti, quindi Approfondimenti sui container.
3. Nella casella a discesa sotto Approfondimenti sui container, scegli Monitoraggio delle prestazioni.
4. Utilizzare le caselle a discesa nella parte superiore per selezionare il tipo di risorsa da visualizzare e la risorsa specifica.

Puoi impostare un CloudWatch allarme su qualsiasi metrica raccolta da Container Insights. Per ulteriori informazioni, consulta Utilizzo degli CloudWatch allarmi Amazon

> **Note**
>
> Se hai già configurato CloudWatch Application Insights per monitorare le tue applicazioni containerizzate, la dashboard di Application Insights viene visualizzata sotto la dashboard di Container Insights. Se non hai già abilitato Application Insights, puoi farlo scegliendo Configurazione automatica di Application Insights sotto la vista delle prestazioni nel pannello di controllo di Container Insights.
> Per ulteriori informazioni su Application Insights e applicazioni containerizzate, consulta Abilitazione di monitoraggio delle risorse di Application Insights per Amazon ECS e Amazon EKS.

Visualizzazione dei fattori determinanti principali

Per alcune visualizzazioni nel monitoraggio delle prestazioni di Container Insights, è inoltre possibile visualizzare i fattori determinanti principali in base alla memoria, alla CPU o alle risorse attive più di recente. Questo valore è disponibile quando selezioni uno dei seguenti pannelli di controllo nella casella a discesa nella parte superiore della pagina:

- Servizi ECS
- Attività di ECS
- Spazi dei nomi EKS
- Servizi EKS
- Pod EKS

Quando visualizzi uno di questi tipi di risorse, nella parte inferiore della pagina viene visualizzata una tabella ordinata inizialmente in base all'utilizzo della CPU. Puoi modificarla per ordinare gli elementi in base all'utilizzo della memoria o all'attività recente. Per visualizzare ulteriori informazioni su una delle righe della tabella, puoi selezionare la casella di controllo accanto a tale riga e scegliere Actions (Operazioni) e scegliere una delle opzioni nel menu Actions (Operazioni).

Utilizzo di CloudWatch Logs Insights per visualizzare i dati di Container Insights

Container Insights raccoglie parametri usando eventi di log delle prestazioni Embedded Metric Format. I log vengono archiviati in CloudWatch Logs. CloudWatch genera automaticamente diverse metriche dai log che è possibile visualizzare nella console. CloudWatch È inoltre possibile eseguire un'analisi più approfondita dei dati sulle prestazioni raccolti utilizzando le query di CloudWatch Logs Insights.

Per ulteriori informazioni su CloudWatch Logs Insights, consulta Analizzare i dati di registro con CloudWatch Logs Insights. Per ulteriori informazioni sui campi di log che puoi utilizzare nelle query, consulta Eventi di log delle prestazioni di Container Insights per Amazon EKS e Kubernetes.

Per utilizzare CloudWatch Logs Insights per interrogare i dati metrici del contenitore

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Informazioni dettagliate.

   Nella parte superiore della schermata si trova l'editor di query. Quando apri CloudWatch Logs Insights per la prima volta, questa casella contiene una query predefinita che restituisce i 20 eventi di registro più recenti.
3. Nella casella sopra l'editor di query, seleziona uno dei gruppi di log Container Insights su cui eseguire query. Perché le seguenti query di esempio funzionino, il nome del gruppo di log deve terminare con performance.

   Quando si seleziona un gruppo di log, CloudWatch Logs Insights rileva automaticamente i campi nei dati del gruppo di log e li visualizza nei Campi rilevati nel riquadro destro. Inoltre, visualizza un grafico a barre di eventi di log in questo gruppo di log nel tempo. Questo grafico a barre mostra l'implementazione di eventi nel gruppo di log che corrisponde alla query e all'intervallo di tempo, non solo gli eventi visualizzati nella tabella.
4. Nell'editor di query, sostituisci la query predefinita con la seguente query e scegli Run query (Esegui query).

```
STATS avg(node_cpu_utilization) as avg_node_cpu_utilization by NodeName
```

```
| SORT avg_node_cpu_utilization DESC
```

Questa query mostra un elenco di nodi ordinati in base all'utilizzo medio della CPU del nodo.

5. Per provare un altro esempio, sostituisci tale query con un'altra query e scegli Run query (Esegui query). Altre query di esempio sono elencate in seguito in questa pagina.

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by
 PodName
| SORT avg_number_of_container_restarts DESC
```

Questa query visualizza un elenco dei pod ordinati in base al numero medio di riavvii di container.

6. Se si desidera provare un'altra query, puoi utilizzare i campi di inclusione nell'elenco a destra dello schermo. Per ulteriori informazioni sulla sintassi delle query, vedere CloudWatch Logs Insights Query Syntax.

Per visualizzare gli elenchi delle risorse

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di spostamento seleziona Resources (Risorse).
3. La visualizzazione predefinita è un elenco delle risorse monitorate da Container Insights e degli avvisi impostati su queste risorse. Per visualizzare una mappa visiva delle risorse, scegli Visualizzazione mappa.
4. Dalla visualizzazione mappa, è possibile mettere il puntatore su qualsiasi risorsa nella mappa per visualizzare i parametri di base relative a tale risorsa. È possibile scegliere qualsiasi risorsa per visualizzare grafici più dettagliati sulla risorsa.

Caso d'uso: visualizzazione dei parametri a livello di attività nei container Amazon ECS

L'esempio seguente illustra come utilizzare CloudWatch Logs Insights per approfondire i log di Container Insights. Per altri esempi, consulta il blog Introduzione ad Amazon CloudWatch Container Insights per Amazon ECS.

Container Insights non genera automaticamente parametri a livello di granularità dell'attività. Nella query seguente vengono visualizzati i parametri a livello di attività per l'utilizzo della CPU e della memoria.

```
stats avg(CpuUtilized) as CPU, avg(MemoryUtilized) as Mem by TaskId, ContainerName
| sort Mem, CPU desc
```

Altre query di esempio per Container Insights

Elenco dei pod, ordinati in base al numero medio di riavvii del container

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by PodName
| SORT avg_number_of_container_restarts DESC
```

Pod richiesti e pod in esecuzione

```
fields @timestamp, @message
| sort @timestamp desc
| filter Type="Pod"
| stats min(pod_number_of_containers) as requested,
 min(pod_number_of_running_containers) as running, ceil(avg(pod_number_of_containers-
pod_number_of_running_containers)) as pods_missing by kubernetes.pod_name
| sort pods_missing desc
```

Numero di errori dei nodi cluster

```
stats avg(cluster_failed_node_count) as CountOfNodeFailures
| filter Type="Cluster"
| sort @timestamp desc
```

Errori di log dell'applicazione in base al nome del container

```
stats count() as countoferrors by kubernetes.container_name
| filter stream="stderr"
| sort countoferrors desc
```

Parametri raccolti da Container Insights

Container Insights raccoglie un set di parametri per Amazon ECS e AWS Fargate Amazon ECS e un set diverso per Amazon EKS, AWS Fargate Amazon EKS e Kubernetes.

I parametri non sono visibili fino a quando le attività del container non sono in esecuzione da qualche tempo.

Argomenti

- Parametri di Amazon ECS Container Insights
- Parametri di Container Insights per Amazon EKS e Kubernetes

Parametri di Amazon ECS Container Insights

La tabella seguente elenca i parametri e le dimensioni raccolti da Container Insights per Amazon ECS. Tali parametri si trovano nello spazio dei nomi `ECS/ContainerInsights`. Per ulteriori informazioni, consulta la pagina Metriche.

Se nella console non viene visualizzato alcun parametro di Container Insights, assicurati di aver completato la configurazione di Container Insights. I parametri vengono visualizzati solo dopo aver completato la configurazione di Container Insights. Per ulteriori informazioni, consulta la pagina Configurazione di Container Insights.

I seguenti parametri sono disponibili al termine delle procedure in Configurazione di Container Insights su Amazon ECS per parametri a livello di cluster e di servizio

Nome parametro	Dimensioni	Descrizione
`ContainerInstanceCount`	ClusterName	Il numero di istanze EC2 che eseguono l'agente Amazon ECS e che sono registrate con un cluster. Questo parametro viene raccolto solo per le istanze di container che eseguono attività Amazon ECS nel cluster. Non viene raccolto per le istanze di container vuote che non hanno alcuna attività Amazon ECS. Unità: numero
`CpuUtilized`	`TaskDefinitionFamily` , `ClusterName`	Le unità CPU utilizzate dalle attività nella risorsa

Nome parametro	Dimensioni	Descrizione

	`ServiceName` , `ClusterName`
	
	`ClusterName`
		specificata dal set di dimensioni in uso.
		
		Questo parametro viene raccolto solo per attività che dispongono di una prenotazione CPU definita nella loro definizione di attività.
		
		Unità: nessuna

`CpuReserved`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Le unità di CPU riservate dalle attività nella risorsa specificata dal set di dimensioni in uso.
		
		Questo parametro viene raccolto solo per attività che dispongono di una prenotazione CPU definita nella loro definizione di attività.
		
		Unità: nessuna

`DeploymentCount`	`ServiceName` , `ClusterName`	Il numero di implement azioni in un servizio Amazon ECS. Unità: numero
`DesiredTaskCount`	`ServiceName` , `ClusterName`	Il numero desiderato di attività per un servizio Amazon ECS. Unità: numero

Nome parametro	Dimensioni	Descrizione

`EBSFilesystemSize`	`VolumeName` , `TaskDefinitionFamily` , `ClusterName`
	
	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
		La quantità totale, in gigabyte (GB), di storage del file system Amazon EBS allocata alle risorse specificate dalle dimensioni utilizzate.
		
		Questa metrica è disponibile solo per le attività eseguite sull'infrastruttura Amazon ECS in esecuzione su Fargate utilizzando la versione della piattaforma o per le istanze Amazon `1.4.0` EC2 che utilizzano la versione dell'agente container o successiva. `1.79.0`
		
		Unità: Gigabyte (GB)

Nome parametro	Dimensioni	Descrizione
EBSFilesystemUtilized	VolumeName , TaskDefinitionFamily , ClusterName TaskDefinitionFamily , ClusterName ServiceName , ClusterName	La quantità totale, in gigabyte (GB), di storage del file system Amazon EBS utilizzata dalle risorse specificate dalle dimensioni che stai utilizzando. Questa metrica è disponibile solo per le attività eseguite sull'infrastruttura Amazon ECS in esecuzione su Fargate utilizzando la versione della piattaforma o per le istanze Amazon 1.4.0 EC2 che utilizzano la versione dell'agente container o successiva. 1.79.0 Per le attività eseguite su Fargate, Fargate riserva spazio sul disco utilizzato solo da Fargate. Non ci sono costi associati allo spazio utilizzato da Fargate, ma vedrai questo spazio di archiviazione aggiuntivo utilizzando strumenti come. df Unità: Gigabyte (GB)

Nome parametro	Dimensioni	Descrizione

`EphemeralStorageReserved` [1]	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Numero di byte riservati dall'archiviazione temporanea nella risorsa specificata dalle dimensioni in uso. L'archiviazione temporanea viene utilizzata per il filesystem root del container e per qualsiasi volume host a montaggio vincolato definito nell'immagine del container e nella definizione dell'attività. La quantità di spazio di archiviazione temporanea non può essere modificata in un'attività in esecuzione.
		
		Questo parametro è disponibile solo per i processi che utilizzano la piattaforma Fargate Linux versione 1.4.0 o successive.
		
		Unità: Gigabyte (GB)

Nome parametro	Dimensioni	Descrizione

`EphemeralStorageUtilized` [1]	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Numero di byte usati dall'archiviazione temporanea nella risorsa specificata dalle dimensioni in uso. L'archiviazione temporanea viene utilizzata per il filesystem root del container e per qualsiasi volume host a montaggio vincolato definito nell'immagine del container e nella definizione dell'attività. La quantità di spazio di archiviazione temporanea non può essere modificata in un'attività in esecuzione.
		
		Questo parametro è disponibile solo per i processi che utilizzano la piattaforma Fargate Linux versione 1.4.0 o successive.
		
		Unità: Gigabyte (GB)

Nome parametro	Dimensioni	Descrizione

`MemoryUtilized`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		La memoria utilizzata dalle attività nella risorsa specificata dal set di dimensioni in uso.
		
		Questo parametro viene raccolto solo per attività che dispongono di una prenotazione di memoria definita nella loro definizione di attività.
		
		Unità: megabyte

`MemoryReserved`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		La memoria riservata dalle attività nella risorsa specificata dal set di dimensioni in uso.
		
		Questo parametro viene raccolto solo per attività che dispongono di una prenotazione di memoria definita nella loro definizione di attività.
		
		Unità: megabyte

Nome parametro	Dimensioni	Descrizione
NetworkRxBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Il numero di byte ricevuti dalla risorsa specificata dalle dimensioni in uso. Questo parametro è ottenuto dal runtime Docker. Questo parametro è disponibile solo per i container che usano le modalità di rete awsvpc e bridge. Unità: byte/secondo
NetworkTxBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Il numero di byte trasmessi dalla risorsa specificata dalle dimensioni in uso. Questo parametro è ottenuto dal runtime Docker. Questo parametro è disponibile solo per i container che usano le modalità di rete awsvpc e bridge. Unità: byte/secondo
PendingTaskCount	ServiceName , ClusterName	Il numero di attività attualmente con stato PENDING. Unità: numero

Nome parametro	Dimensioni	Descrizione
RunningTaskCount	ServiceName , ClusterName	Il numero di attività attualmente con stato RUNNING. Unità: numero
ServiceCount	ClusterName	Il numero di servizi nel cluster. Unità: numero
StorageReadBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Il numero di byte letti dall'archiviazione sull'istanza nella risorsa specificata dalle dimensioni in uso. Questo non include i byte letti per i dispositivi di archiviazione. Questo parametro è ottenuto dal runtime Docker. Unità: byte
StorageWriteBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Il numero di byte scritti nello storage nella risorsa specificata dalle dimensioni in uso. Questo parametro è ottenuto dal runtime Docker. Unità: byte
TaskCount	ClusterName	Il numero di attività in esecuzione nel cluster. Unità: numero

Nome parametro	Dimensioni	Descrizione
`TaskSetCount`	`ServiceName` , `ClusterName`	Il numero di set di attività nel servizio. Unità: numero

> **Note**
>
> Le metriche `EphemeralStorageReserved` e `EphemeralStorageUtilized` sono disponibili solo per i processi che utilizzano la piattaforma Fargate Linux versione 1.4.0 o successive.
> Fargate riserva spazio su disco destinato unicamente a questo motore di calcolo. Non ti viene addebitato alcun costo. Sebbene non sia mostrato in queste metriche, puoi visualizzare questo spazio di archiviazione aggiuntivo in altri strumenti, come `df`.

I seguenti parametri sono disponibili al termine delle procedure in Implementazione dell' CloudWatch agente per raccogliere parametri a livello di istanza EC2 su Amazon ECS

Nome parametro	Dimensioni	Descrizione
`instance_cpu_limit`	`ClusterName`	Il numero massimo di unità di CPU che è possibile assegnare a una singola istanza EC2 nel cluster corrente. Unità: nessuna

`instance_cpu_reserved_capacity`	`ClusterName`
	
	`InstanceId` , `ContainerInstanceId` , `ClusterName`
		La percentuale di CPU attualmente prenotata su una singola istanza EC2 nel cluster.

Nome parametro	Dimensioni	Descrizione
		Unità: percentuale
instance_cpu_usage_total	ClusterName	Il numero di unità CPU utilizzate su una singola istanza EC2 nel cluster. Unità: nessuna
instance_cpu_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	La percentuale totale di unità CPU utilizzate su una singola istanza EC2 nel cluster. Unità: percentuale
instance_filesystem_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	La percentuale totale di capacità del file system utilizzata su una singola istanza EC2 nel cluster. Unità: percentuale
instance_memory_limit	ClusterName	La quantità massima di memoria, espressa in byte, che può essere assegnata a una singola istanza EC2 nel cluster corrente. Unità: byte

Nome parametro	Dimensioni	Descrizione
instance_memory_reserved_capacity	ClusterName InstanceId , ContainerInstanceId , ClusterName	La percentuale di memoria attualmente prenotata su una singola istanza EC2 nel cluster. Unità: percentuale
instance_memory_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	La percentuale totale di memoria utilizzata su una singola istanza EC2 nel cluster. Unità: percentuale
instance_memory_working_set	ClusterName	La quantità di memoria, in byte, utilizzata su una singola istanza EC2 nel cluster. Unità: byte
instance_network_total_bytes	ClusterName	Il numero totale di byte al secondo trasmessi e ricevuti tramite la rete su una singola istanza EC2 nel cluster. Unità: byte/secondo

Nome parametro	Dimensioni	Descrizione
`instance_number_of_running_tasks`	`ClusterName`	Il numero di attività in esecuzione su una singola istanza EC2 nel cluster. Unità: numero

Parametri di Container Insights per Amazon EKS e Kubernetes

Le tabelle seguenti elencano le metriche e le dimensioni raccolte da Container Insights per Amazon EKS e Kubernetes. Tali parametri si trovano nello spazio dei nomi `ContainerInsights`. Per ulteriori informazioni, consulta la pagina Metriche.

Se nella console non viene visualizzato alcun parametro di Container Insights, assicurati di aver completato la configurazione di Container Insights. I parametri vengono visualizzati solo dopo aver completato la configurazione di Container Insights. Per ulteriori informazioni, consulta la pagina Configurazione di Container Insights.

Se utilizzi la versione 1.5.0 o successiva del componente aggiuntivo Amazon EKS o la versione 1.300035.0 dell' CloudWatch agente, la maggior parte delle metriche elencate nella tabella seguente viene raccolta per i nodi Linux e Windows. Consulta la colonna Metric Name della tabella per vedere quali metriche non vengono raccolte per Windows.

Con la versione originale di Approfondimenti sui container, i parametri vengono addebitati come parametri personalizzati. Con Approfondimenti sui container con osservabilità migliorata per Amazon EKS, i parametri di Approfondimenti sui container vengono addebitati per osservazione anziché per parametro archiviato o log importato. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

> **Note**
>
> In Windows, i parametri di rete come `pod_network_rx_bytes` e non `pod_network_tx_bytes` vengono raccolti per i contenitori dei processi host.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
`cluster_failed_node_count`	`ClusterName`		Il numero di nodi di lavoro non riusciti nel cluster. Un nodo è considerato non riuscito se risente delle condizioni del nodo. Per ulteriori informazioni, consulta Conditions (Condizioni) nella documentazione Kubernetes.
`cluster_node_count`	`ClusterName`		Il numero totale di nodi di lavoro nel cluster.

`namespace_number_of_running_pods`	`Namespace` `ClusterName`
	
	`ClusterName`
			Il numero di pod in esecuzione per spazio dei nomi nella risorsa specificata dalle dimensioni in uso.

`node_cpu_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	Il numero massimo di unità di CPU che può essere assegnato a un singolo nodo nel cluster corrente.
`node_cpu_reserved_capacity`	`NodeName`, `ClusterName` , `InstanceId`		La percentuale di unità CPU riservate per i componenti del

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
	`ClusterName`		nodo, ad esempio kubelet, kube-proxy e Docker. Formula: `node_cpu_request / node_cpu_limit` **Note** `node_cpu_request` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			EKS e Kubernetes.
node_cpu_usage_total	ClusterName	ClusterName , InstanceId , NodeName	Il numero di unità di CPU in uso sui nodi del cluster.
node_cpu_utilization	NodeName, ClusterName , InstanceId ClusterName		La percentuale totale delle unità di CPU in uso sui nodi del cluster. Formula: node_cpu_usage_total / node_cpu_limit

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_filesystem_utilization`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			La percentuale totale della capacità del file system in uso sui nodi del cluster.
			
			Formula: `node_filesystem_usage / node_filesystem_capacity`
			
			Note
			`node_filesystem_usage` e `node_filesystem_capacity` non vengono riportati direttamente come parametri, ma sono campi nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes.
`node_memory_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	La quantità massima di memoria, espressa in byte, che può essere assegnata a un singolo nodo nel cluster corrente.

`node_filesystem_inodes`
 	
Questa metrica è disponibile solo con Container Insights con osservabilità migliorata per Amazon EKS. Non è disponibile su Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Il numero totale di inode (utilizzati e inutilizzati) su un nodo.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_filesystem_inodes_free`
 	
Questa metrica è disponibile solo con Container Insights con osservabilità migliorata per Amazon EKS. Non è disponibile su Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Il numero di inode inutilizzati su un nodo.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_memory_reserved_capacity`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			La percentuale di memoria attualmente utilizzata sui nodi del cluster.
			
			Formula: `node_memory_request / node_memory_limit`
			
			Note
			`node_memory_request` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			EKS e Kubernetes.
node_memory_utilization	NodeName, ClusterName , InstanceId ClusterName		La percentuale di memoria attualmente utilizzata dal nodo o dai nodi. È la percentuale di utilizzo della memoria del nodo rispetto alla limitazione di memoria del nodo. Formula: node_memory_working_set / node_memory_limit .
node_memory_working_set	ClusterName	ClusterName , InstanceId , NodeName	La quantità di memoria, espressa in byte, in uso nel working set dei nodi del cluster.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_network_total_bytes`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Il numero totale di byte al secondo trasmessi e ricevuti tramite la rete per nodo in un cluster.
			
			Formula: `node_network_rx_bytes + node_network_tx_bytes`
			
			Note
			`node_network_rx_bytes` e `node_network_tx_bytes` non vengono riportati direttamente come parametri, ma sono campi nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes.
node_number_of_running_containers	NodeName, ClusterName , InstanceId ClusterName		Il numero di container in esecuzione per nodo in un cluster.
node_number_of_running_pods	NodeName, ClusterName , InstanceId ClusterName		Il numero di pod in esecuzione per nodo in un cluster.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_status_allocatable_pods`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Il numero di pod che è possibile assegnare a un nodo in base alle relative risorse allocabili, definito come la parte restante della capacità di un nodo dopo aver tenuto conto delle prenotazioni dei daemon di sistema e delle soglie di espulsione forzata.

`node_status_capacity_pods`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Il numero di pod che possono essere assegnati a un nodo in base alla sua capacità.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_status_condition_ready`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indica se la condizione dello stato del nodo `Ready` è vera.

`node_status_condition_memory_pressure`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indica se la condizione dello stato del nodo `MemoryPressure` è vera.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_status_condition_pid_pressure`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indica se la condizione dello stato del nodo `PIDPressure` è vera.

`node_status_condition_disk_pressure`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indica se la condizione dello stato del nodo `OutOfDisk` è vera.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_status_condition_unknown`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Indica se una qualsiasi delle condizioni di stato del nodo è sconosciuta.

`node_interface_network_rx_dropped`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Il numero di pacchetti ricevuti e successivamente annullati da questa interfaccia di rete sul nodo.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_interface_network_tx_dropped`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Il numero di pacchetti che dovevano essere trasmessi ma che sono stati annullati da un'interfaccia di rete sul nodo.

`node_diskio_io_service_bytes_total`
 	
Questa metrica è disponibile solo con Container Insights con osservabilità migliorata per Amazon EKS. Non è disponibile su Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Il numero totale di byte trasferiti da tutte le operazioni di I/O sul nodo.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`node_diskio_io_serviced_total`
 	
Questa metrica è disponibile solo con Container Insights con osservabilità migliorata per Amazon EKS. Non è disponibile su Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Il numero totale di operazioni di I/O sul nodo.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_cpu_reserved_capacity`	PodName, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , PodName, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			La capacità della CPU riservata per pod in un cluster.
			
			Formula: `pod_cpu_request / node_cpu_limit`
			
			Note
			`pod_cpu_request` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_cpu_utilization`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Service, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	La percentuale di unità CPU utilizzate dai pod.
			
			Formula: `pod_cpu_usage_total / node_cpu_limit`
			
			Note
			`pod_cpu_usage_total` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
`pod_cpu_utilization_over_pod_limit`	PodName, Namespace, `ClusterName` Namespace, `ClusterName` Service, Namespace, `ClusterName` `ClusterName`	`ClusterName` , `Namespace` , PodName, `FullPodName`	La percentuale di unità CPU utilizzata dai pod relativa al limite di pod. Formula: `pod_cpu_usage_total / pod_cpu_limit` **Note** `pod_cpu_usage_total` e `pod_cpu_limit` non vengono riportati direttamente come parametri, ma sono campi nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazio

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			ni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_memory_reserved_capacity`	`PodName`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			La percentuale di memoria riservata per i pod.
			
			Formula: `pod_memory_request / node_memory_limit`
			
			Note
			`pod_memory_request` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_memory_utilization`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	`Service`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`	La percentuale di memoria attualmente utilizzata dal pod o dai pod.
			
			Formula: `pod_memory_working_set / node_memory_limit`
			
			Note
			`pod_memory_working_set` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazio

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			ni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_memory_utilization_over_pod_limit`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	Service, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	La percentuale di memoria utilizzata dai pod relativa al limite di pod. Se uno qualsiasi dei container nel pod non ha un limite di memoria definito, questo parametro non viene visualizzato.
			
			Formula: `pod_memory_working_set / pod_memory_limit`
			
			Note
			`pod_memory_working_set` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazi

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			oni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_network_rx_bytes`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	`Service`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Il numero di byte al secondo ricevuti sulla rete dal pod.
			
			Formula: `sum(pod_interface_network_rx_bytes)`
			
			Note
			`pod_interface_network_rx_bytes` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_network_tx_bytes`	`PodName`, `Namespace`, `ClusterName`
	
	`Namespace`, `ClusterName`
	
	`Service`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Il numero di byte al secondo trasmessi sulla rete dal pod.
			
			Formula: `sum(pod_interface_network_tx_bytes)`
			
			Note
			`pod_interface_network_tx_bytes` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_cpu_request`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Le richieste della CPU per il pod.
			
			Formula: `sum(container_cpu_request)`
			
			Note
			`pod_cpu_request` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_memory_request`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Le richieste di memoria per il pod.
			
			Formula: `sum(container_memory_request)`
			
			Note
			`pod_memory_request` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_cpu_limit`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Il limite di CPU definito per i container nel pod. Se uno qualsiasi dei container nel pod non ha un limite di CPU definito, questo parametro non viene visualizzato.
			
			Formula: `sum(container_cpu_limit)`
			
			Note
			`pod_cpu_limit` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			delle prestazioni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_memory_limit`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Il limite di memoria definito per i container nel pod. Se uno qualsiasi dei container nel pod non ha un limite di memoria definito, questo parametro non viene visualizzato.
			
			Formula: `sum(container_memory_limit)`
			
			Note
			`pod_cpu_limit` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			eventi di log delle prestazioni per Amazon EKS e Kubernetes.

`pod_status_failed`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica che tutti i container nel pod sono terminati e che almeno un container è terminato con uno stato diverso da zero o è stato terminato dal sistema.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_status_ready`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica che tutti i container nel pod sono pronti, dopo aver raggiunto la condizione `Container Ready` .

`pod_status_running`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica che tutti i container nel pod sono in esecuzione.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_status_scheduled`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica che il pod è stato pianificato su un nodo.

`pod_status_unknown`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica che lo stato del pod non può essere ottenuto.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_status_pending`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica che il pod è stato accettato dal cluster ma uno o più container non sono ancora pronti.

`pod_status_succeeded`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indica che tutti i container nel pod sono stati terminati correttamente e non verranno riavviati.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_number_of_containers`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Riporta il numero di container definito nella specifica del pod.

`pod_number_of_running_containers`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Riporta il numero di container nel pod che si trovano attualmente nello stato `Running`.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_container_status_terminated`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Riporta il numero di container nel pod che si trovano nello stato `Terminated` .

`pod_container_status_running`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Riporta il numero di container nel pod che si trovano nello stato `Running`.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_container_status_waiting`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Riporta il numero di container nel pod che si trovano nello stato `Waiting`.

`pod_interface_network_rx_dropped`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Il numero di pacchetti ricevuti e successivamente annullati da un'interfaccia di rete per il pod.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`pod_interface_network_tx_dropped`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Il numero di pacchetti che dovevano essere trasmessi ma che sono stati annullati per il pod.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`container_cpu_utilization`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			La percentuale di unità CPU utilizzate dal container.
			
			Formula: `container_cpu_usage_total / node_cpu_limit`
			
			Note
			`container_cpu_utilization` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`container_cpu_utilization_over_container_limit`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			La percentuale di unità CPU utilizzate dal container in rapporto al limite del container. Se il container non ha un limite di CPU definito, questo parametro non viene visualizzato.
			
			Formula: `container_cpu_usage_total / container_cpu_limit`
			
			Note
			`container_cpu_utilization_over_container_limit` non viene riportato direttamente come parametro, ma è un campo nei log eventi delle

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`container_memory_utilization`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			La percentuale di unità di memoria utilizzate dal container .
			
			Formula: `container_memory_working_set / node_memory_limit`
			
			Note
			`container_memory_utilization` non viene riportato direttamente come parametro , ma è un campo nei log eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			delle prestazioni per Amazon EKS e Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`container_memory_utilization_over_container_limit`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			La percentuale di unità di memoria utilizzate dal container in rapporto al limite del container. Se il container non ha un limite di memoria definito, questo parametro non viene visualizzato.
			
			Formula: `container_memory_working_set / container_memory_limit`
			
			Note
			`container_memory_utilization_over_container_limit` non viene riportato direttamente come parametro, ma è un campo nei log

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
			eventi delle prestazioni. Per ulteriori informazioni, consulta Campi rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes.

`container_memory_failures_total`
 	
Questa metrica è disponibile solo con Container Insights con osservabilità migliorata per Amazon EKS. Non è disponibile su Windows.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Il numero di errori di allocazione della memoria riscontrati dal container.

`pod_number_of_container_restarts`	PodName, `Namespace` , `ClusterName`		Il numero totale di riavvii del container in un pod.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione
`service_number_of_running_pods`	Service, `Namespace`, `ClusterName` `ClusterName`		Il numero di pod che eseguono il servizio o i servizi nel cluster.

`replicas_desired`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
			Il numero di pod desiderato per un carico di lavoro come definito nella specifica del carico di lavoro.

`replicas_ready`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
			Il numero di pod per un carico di lavoro che hanno raggiunto lo stato pronto.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`status_replicas_available`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			Il numero di pod disponibili per un carico di lavoro. Un pod è disponibile quando è pronto per il tempo `minReadySeconds` definito nella specifica del carico di lavoro.

`status_replicas_unavailable`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			Il numero di pod non disponibili per un carico di lavoro. Un pod è disponibile quando è pronto per il tempo `minReadySeconds` definito nella specifica del carico di lavoro. I pod non sono disponibili se non soddisfano questo criterio.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`apiserver_storage_objects`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `resource`
			Il numero di oggetti memorizzati in etcd al momento dell'ultimo controllo.

`apiserver_request_total`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			Il numero totale di richieste API al server API Kubernetes.

`apiserver_request_duration_seconds`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `verb`
			Latenza di risposta per le richieste API al server API Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`apiserver_admission_controller_admission_duration_seconds`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			Latenza del controller di ammissione in secondi. Un controller di ammissione è un codice che intercetta le richieste al server API Kubernetes.

`rest_client_request_duration_seconds`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			Latenza di risposta riscontrata dai client che chiamano il server API Kubernetes. Questo parametro è sperimentale e potrebbe cambiare nelle future versioni di Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`rest_client_requests_total`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `code`, `method`
			Il numero totale di richieste API al server API Kubernetes effettuate dai client. Questo parametro è sperimentale e potrebbe cambiare nelle future versioni di Kubernetes.

`etcd_request_duration_seconds`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `operation`
			Latenza di risposta delle chiamate API a Etcd. Questo parametro è sperimentale e potrebbe cambiare nelle future versioni di Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`apiserver_storage_size_bytes`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , endpoint`
			Dimensione del file del database di archiviazione allocato fisicamente, espressa in byte. Questo parametro è sperimentale e potrebbe cambiare nelle future versioni di Kubernetes.

`apiserver_longrunning_requests`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , resource`
			Il numero di richieste attive di lunga durata al server API Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`apiserver_current_inflight_requests`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , request_kind`
			Il numero di richieste che il server API Kubernetes sta elaborando.

`apiserver_admission_webhook_admission_duration_seconds`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , name`
			Latenza del webhook di ammissione in secondi. I webhook di ammissione sono callback HTTP che ricevono le richieste di ammissione e le utilizzano a uno scopo.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`apiserver_admission_step_admission_duration_seconds`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , operation`
			Latenza delle fasi secondarie di ammissione in secondi.

`apiserver_requested_deprecated_apis`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , group`
			Il numero di richieste API obsolete al server API Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`apiserver_request_total_5XX`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			Il numero di richieste al server API Kubernetes a cui è stata data risposta con un codice di risposta HTTP 5XX.

`apiserver_storage_list_duration_seconds`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName`, `resource`
			Latenza di risposta dell'elencazione degli oggetti da Etcd. Questo parametro è sperimentale e potrebbe cambiare nelle future versioni di Kubernetes.

Nome parametro	Dimensioni con qualsiasi versione di Approfondimenti sui container	Dimensioni aggiuntive con Approfondimenti sui container con osservabilità migliorata per Amazon EKS	Descrizione

`apiserver_current_inqueue_requests`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , request_kind`
			Il numero di richieste in coda messe in coda dal server API Kubernetes. Questo parametro è sperimentale e potrebbe cambiare nelle future versioni di Kubernetes.

`apiserver_flowcontrol_rejected_requests_total`
 	
Questo parametro è disponibile solo con Approfondimenti sui container con osservabilità migliorata per Amazon EKS
		`ClusterName`
		
		`ClusterName , reason`
			Il numero di richieste rifiutate dal sottosistema API Priority and Fairness. Questo parametro è sperimentale e potrebbe cambiare nelle future versioni di Kubernetes.

Metriche della GPU NVIDIA

A partire dalla versione `1.300034.0` dell' CloudWatch agente, Container Insights con osservabilità migliorata per Amazon EKS raccoglie per impostazione predefinita le metriche delle GPU NVIDIA dai carichi di lavoro EKS. L' CloudWatch agente deve essere installato utilizzando la versione aggiuntiva CloudWatch Observability EKS o successiva. `v1.3.0-eksbuild.1` Per ulteriori informazioni,

consulta Installa l' CloudWatch agente utilizzando il componente aggiuntivo Amazon CloudWatch Observability EKS. Le metriche della GPU NVIDIA raccolte sono elencate nella tabella di questa sezione.

Affinché Container Insights raccolga i parametri della GPU NVIDIA, è necessario soddisfare i seguenti prerequisiti:

- Devi utilizzare Container Insights con osservabilità migliorata per Amazon EKS, con la versione `v1.3.0-eksbuild.1` aggiuntiva Amazon CloudWatch Observability EKS o successiva.
- Il plug-in del dispositivo NVIDIA per Kubernetes deve essere installato nel cluster.
- Il toolkit NVIDIA Container deve essere installato sui nodi del cluster. Ad esempio, le AMI accelerate ottimizzate per Amazon EKS sono costruite con i componenti necessari.

Puoi scegliere di non raccogliere i parametri della GPU NVIDIA impostando l'`accelerated_compute_metrics`opzione nel file di configurazione dell'agente CloudWatch beginn su. `false` Per ulteriori informazioni e un esempio di configurazione di opt-out, consulta. (Facoltativo) Configurazione aggiuntiva

Nome parametro	Dimensioni	Descrizione

`container_gpu_memory_total`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		La dimensione totale del frame buffer, in byte, sulle GPU allocate al contenitore.

`container_gpu_memory_used`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
		I byte del frame buffer utilizzati sulle GPU allocate al contenitore.

Nome parametro	Dimensioni	Descrizione

	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`	

`container_gpu_memory_utilization`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		La percentuale di frame buffer utilizzata delle GPU allocate al contenitore.

`container_gpu_power_draw`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		Il consumo energetico in watt della o delle GPU allocate al contenitore.

Nome parametro	Dimensioni	Descrizione
container_gpu_temperature	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	La temperatura in gradi Celsius delle GPU allocate al contenitore.
container_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	La percentuale di utilizzo delle GPU allocate al contenitore.
node_gpu_memory_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	La dimensione totale del frame buffer, in byte, sulle GPU allocate al nodo.

Nome parametro	Dimensioni	Descrizione
node_gpu_memory_used	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	I byte del frame buffer utilizzati sulle GPU allocate al nodo.
node_gpu_memory_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	La percentuale di frame buffer utilizzata sulle GPU allocate al nodo.
node_gpu_power_draw	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Il consumo energetico in watt della o delle GPU allocate al nodo.
node_gpu_temperature	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	La temperatura in gradi Celsius delle GPU allocate al nodo.

Nome parametro	Dimensioni	Descrizione
node_gpu_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	La percentuale di utilizzo delle GPU allocate al nodo.
pod_gpu_memory_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	La dimensione totale del frame buffer, in byte, sulle GPU allocate al pod.

Nome parametro	Dimensioni	Descrizione
pod_gpu_memory_used	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	I byte del frame buffer utilizzati sulle GPU allocate al pod.
pod_gpu_memory_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	La percentuale di frame buffer utilizzata delle GPU allocate al pod.

Nome parametro	Dimensioni	Descrizione
pod_gpu_power_draw	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Il consumo energetico in watt della o delle GPU allocate al pod.
pod_gpu_temperature	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	La temperatura in gradi Celsius delle GPU assegnate al pod.

Nome parametro	Dimensioni	Descrizione
pod_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	La percentuale di utilizzo delle GPU allocate al pod.

AWS Metriche dei neuroni per Trainium e Inferentia AWSAWS

A partire dalla versione 1.300036.0 dell' CloudWatch agente, Container Insights con osservabilità migliorata per Amazon EKS raccoglie per impostazione predefinita i parametri di elaborazione accelerata dagli AWS acceleratori Trainium e AWS Inferentia. L' CloudWatch agente deve essere installato utilizzando la versione aggiuntiva Observability EKS o successiva. CloudWatch v1.5.0-eksbuild.1 Per ulteriori informazioni sul componente aggiuntivo, vedere. Installa l' CloudWatch agente utilizzando il componente aggiuntivo Amazon CloudWatch Observability EKS Per ulteriori informazioni su AWS Trainium, vedere AWS Trainium. Per ulteriori informazioni su AWS Inferentia, vedere Inferentia.AWS

Affinché Container Insights raccolga le metriche di AWS Neuron, è necessario soddisfare i seguenti prerequisiti:

- Devi utilizzare Container Insights con osservabilità migliorata per Amazon EKS, con la versione v1.5.0-eksbuild.1 aggiuntiva Amazon CloudWatch Observability EKS o successiva.
- Il driver Neuron deve essere installato sui nodi del cluster.
- Il plug-in del dispositivo Neuron deve essere installato sul cluster. Ad esempio, le AMI accelerate ottimizzate per Amazon EKS sono costruite con i componenti necessari.

Le metriche raccolte sono elencate nella tabella di questa sezione. Le metriche vengono raccolte per AWS Trainium, AWS Inferentia e Inferentia2. AWS

L' CloudWatch agente raccoglie queste metriche dal monitor Neuron ed esegue la necessaria correlazione delle risorse Kubernetes per fornire le metriche a livello di pod e container

Nome parametro	Dimensioni	Descrizione
container_neuroncore_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	NeuronCore utilizzo, durante il periodo di acquisizione, del materiale allocato al contenitore. NeuronCore Unità: percentuale
container_neuroncore_memory_usage_constants	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per le costanti durante l'addestramento da parte del NeuronCore che viene allocata al contenitore (o i pesi durante l'inferenza). Unità: byte
container_neuroncore_memory_usage_model_code	ClusterName ClusterName , Namespace , PodName, ContainerName	La quantità di memoria del dispositivo utilizzata per il codice eseguibile dei modelli da NeuronCore che viene allocata al contenitore. Unità: byte

Nome parametro	Dimensioni	Descrizione

	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `NeuronDevice` , `NeuronCore`	

`container_neuroncore_memory_usage_model_shared_scratchpad`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `NeuronDevice` , `NeuronCore`
		La quantità di memoria del dispositivo utilizzata per lo scratchpad condiviso dai modelli dal NeuronCore che viene allocata al contenitore. Questa area di memoria è riservata ai modelli.
		
		Unità: byte

`container_neuroncore_memory_usage_runtime_memory`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `NeuronDevice` , `NeuronCore`
		La quantità di memoria del dispositivo utilizzata per il runtime Neuron da quella NeuronCore allocata al contenitore.
		
		Unità: byte

Nome parametro	Dimensioni	Descrizione
container_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per i tensori da quella NeuronCore allocata al contenitore. Unità: byte
container_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La quantità totale di memoria utilizzata dall' NeuronCore allocato al contenitore. Unità: byte

Nome parametro	Dimensioni	Descrizione
container_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice	Il numero di eventi ECC corretti e non corretti per la SRAM sul chip e la memoria del dispositivo Neuron sul nodo. Unità: numero
pod_neuroncore_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	L' NeuronCore utilizzo durante il periodo di acquisizione del materiale allocato al pod. NeuronCore Unità: percentuale

Nome parametro	Dimensioni	Descrizione
pod_neuroncore_memory_usage_constants	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per le costanti durante l'addestramento da the NeuronCore che viene allocata al pod (o i pesi durante l'inferenza). Unità: byte
pod_neuroncore_memory_usage_model_code	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per il codice eseguibile dei modelli da NeuronCore che viene allocata al pod. Unità: byte

Nome parametro	Dimensioni	Descrizione
pod_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per lo scratchpad condiviso dai modelli dal NeuronCore che viene allocata al pod. Questa area di memoria è riservata ai modelli. Unità: byte
pod_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per il runtime di Neuron dall'area NeuronCore allocata al pod. Unità: byte

Nome parametro	Dimensioni	Descrizione
pod_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per i tensori da quella NeuronCore allocata al pod. Unità: byte
pod_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantità totale di memoria utilizzata dall' NeuronCore allocazione al pod. Unità: byte

Nome parametro	Dimensioni	Descrizione
pod_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice	Il numero di eventi ECC corretti e non corretti per la SRAM sul chip e la memoria del dispositivo Neuron allocati a un pod. Unità: byte
node_neuroncore_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	L' NeuronCore utilizzo durante il periodo di acquisizione dell'allocazione al nodo. NeuronCore Unità: percentuale
node_neuroncore_memory_usage_constants	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per le costanti durante l'addestramento da the NeuronCore che viene allocata al nodo (o i pesi durante l'inferenza). Unità: byte

Nome parametro	Dimensioni	Descrizione
node_neuroncore_memory_usage_model_code	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per il codice eseguibile dei modelli da NeuronCore che viene allocata al nodo. Unità: byte
node_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per lo scratchpad condiviso dai modelli dal NeuronCore che viene allocata al nodo. Si tratta di un'area di memoria riservata ai modelli. Unità: byte
node_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per il runtime di Neuron da NeuronCore che viene allocata al nodo. Unità: byte
node_neuroncore_memory_usage_tensors	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantità di memoria del dispositivo utilizzata per i tensori da NeuronCore that viene allocata al nodo. Unità: byte

Nome parametro	Dimensioni	Descrizione
node_neuroncore_memory_usage_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantità totale di memoria utilizzata da NeuronCore that viene allocata al nodo. Unità: byte
node_neuron_execution_errors_total	ClusterName ClusterName , InstanceId , NodeName	Il numero totale di errori di esecuzione sul nodo. Viene calcolato dall' CloudWatch agente aggregando gli errori dei seguenti tipi:generic,, numerical transient , modelruntime, e hardware Unità: numero
node_neurondevice_runtime_memory_used_bytes	ClusterName ClusterName , InstanceId , NodeName	L'utilizzo totale della memoria del dispositivo Neuron in byte sul nodo. Unità: byte
node_neuron_execution_latency	ClusterName ClusterName , InstanceId , NodeName	In secondi, la latenza per un'esecuzione sul nodo misurata dal runtime di Neuron. Unità: secondi
node_neurondevice_hw_ecc_events_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , NodeName, NeuronDevice	Il numero di eventi ECC corretti e non corretti per la SRAM su chip e la memoria del dispositivo Neuron sul nodo. Unità: numero

AWS Metriche Elastic Fabric Adapter (EFA)

A partire dalla versione `1.300037.0` dell' CloudWatch agente, Container Insights con osservabilità migliorata per Amazon EKS raccoglie i parametri AWS Elastic Fabric Adapter (EFA) dai cluster Amazon EKS su istanze Linux. L' CloudWatch agente deve essere installato utilizzando la versione aggiuntiva CloudWatch Observability EKS o successiva. `v1.5.2-eksbuild.1` Per ulteriori informazioni sul componente aggiuntivo, vedere. Installa l' CloudWatch agente utilizzando il componente aggiuntivo Amazon CloudWatch Observability EKS Per ulteriori informazioni su AWS Elastic Fabric Adapter, consulta Elastic Fabric Adapter.

Affinché Container Insights raccolga le metriche dell'adattatore AWS Elastic Fabric, devi soddisfare i seguenti prerequisiti:

- Devi utilizzare Container Insights con osservabilità migliorata per Amazon EKS, con la versione `v1.5.2-eksbuild.1` aggiuntiva Amazon CloudWatch Observability EKS o successiva.
- Il plug-in del dispositivo EFA deve essere installato nel cluster. Per ulteriori informazioni, vedere aws-efa-k8 s-device-plugin su. GitHub

Le metriche raccolte sono elencate nella tabella seguente.

Nome parametro	Dimensioni	Descrizione

`container_efa_rx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Il numero di byte al secondo ricevuti dai dispositivi EFA allocati al contenitore.
		
		Unità: byte/secondo

`container_efa_tx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
		Il numero di byte al secondo trasmessi dai dispositivi EFA allocati al contenitore.

Nome parametro	Dimensioni	Descrizione

	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Unità: byte/secondo

`container_efa_rx_dropped`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Il numero di pacchetti ricevuti e poi rilasciati dai dispositivi EFA assegnati al contenitore.
		
		Unità: conteggio/secondo

`container_efa_rdma_read_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Il numero di byte al secondo ricevuti utilizzando le operazioni di lettura con accesso diretto alla memoria remota dai dispositivi EFA allocati al contenitore.
		
		Unità: byte/secondo

Nome parametro	Dimensioni	Descrizione
container_efa_rdma_write_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Il numero di byte al secondo trasmessi utilizzando le operazioni di lettura con accesso diretto alla memoria remota dai dispositivi EFA allocati al contenitore. Unità: byte/secondo
container_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Il numero di byte al secondo ricevuti durante le operazioni di scrittura con accesso diretto alla memoria remota dai dispositivi EFA allocati al contenitore. Unità: byte/secondo

Nome parametro	Dimensioni	Descrizione
pod_efa_rx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Il numero di byte al secondo ricevuti dai dispositivi EFA allocati al pod. Unità: byte/secondo
pod_efa_tx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Il numero di byte al secondo trasmessi dai dispositivi EFA assegnati al pod. Unità: byte/secondo

Nome parametro	Dimensioni	Descrizione
pod_efa_rx_dropped	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Il numero di pacchetti ricevuti e poi rilasciati dai dispositivi EFA assegnati al pod. Unità: conteggio/secondo
pod_efa_rdma_read_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Il numero di byte al secondo ricevuti utilizzando le operazioni di lettura con accesso diretto alla memoria remota dai dispositivi EFA allocati al pod. Unità: byte/secondo

Nome parametro	Dimensioni	Descrizione
pod_efa_rdma_write_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Il numero di byte al secondo trasmessi utilizzando le operazioni di lettura con accesso diretto alla memoria remota dai dispositivi EFA assegnati al pod. Unità: byte/secondo
pod_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Il numero di byte al secondo ricevuti durante le operazioni di scrittura con accesso diretto alla memoria remota dai dispositivi EFA allocati al pod. Unità: byte/secondo

Nome parametro	Dimensioni	Descrizione
node_efa_rx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Il numero di byte al secondo ricevuti dai dispositivi EFA allocati al nodo. Unità: byte/secondo
node_efa_tx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Il numero di byte al secondo trasmessi dai dispositivi EFA assegnati al nodo. Unità: byte/secondo
node_efa_rx_dropped	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Il numero di pacchetti ricevuti e poi rilasciati dai dispositivi EFA assegnati al nodo. Unità: conteggio/secondo
node_efa_rdma_read_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Il numero di byte al secondo ricevuti utilizzando le operazioni di lettura di accesso diretto alla memoria remota dai dispositivi EFA allocati al nodo. Unità: byte/secondo

Nome parametro	Dimensioni	Descrizione
pod_efa_rdma_write_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Il numero di byte al secondo trasmessi utilizzando le operazioni di lettura con accesso diretto alla memoria remota dai dispositivi EFA assegnati al pod. Unità: byte/secondo
node_efa_rdma_write_recv_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Il numero di byte al secondo ricevuti durante le operazioni di scrittura con accesso diretto alla memoria remota dai dispositivi EFA allocati al nodo. Unità: byte/secondo

Documentazione di riferimento dei log delle prestazioni di Container Insights

Questa sezione include informazioni di riferimento su come Container Insights utilizza gli eventi di log delle prestazioni per raccogliere i parametri. Quando si implementa Approfondimenti sui container, viene creato automaticamente un gruppo di log per gli eventi del log delle prestazioni. Non è necessario creare questo gruppo di log da soli.

Argomenti

- Eventi di log delle prestazioni di Container Insights per Amazon ECS
- Eventi di log delle prestazioni di Container Insights per Amazon EKS e Kubernetes
- Campi rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes

Eventi di log delle prestazioni di Container Insights per Amazon ECS

Di seguito sono elencati alcuni esempi di eventi di log delle prestazioni che Container Insights raccoglie da Amazon ECS.

Questi registri si trovano in CloudWatch Logs, in un gruppo di log denominato. `/aws/ecs/containerinsights/`*`CLUSTER_NAME`*`/performance` All'interno di quel gruppo di log, ogni istanza del container avrà un flusso di log denominato `AgentTelemetry-`*`CONTAINER_INSTANCE_ID`*.

È possibile interrogare questi log utilizzando query come `{ $.Type = "Container" }` per visualizzare tutti i log eventi del container.

Tipo: Container

```
{
 "Version":"0",
 "Type":"Container",
 "ContainerName":"sleep",
 "TaskId":"7ac4dfba69214411b4783a3b8189c9ba",
 "TaskDefinitionFamily":"sleep360",
 "TaskDefinitionRevision":"1",
 "ContainerInstanceId":"0d7650e6dec34c1a9200f72098071e8f",
 "EC2InstanceId":"i-0c470579dbcdbd2f3",
 "ClusterName":"MyCluster",
 "Image":"busybox",
 "ContainerKnownStatus":"RUNNING",
 "Timestamp":1623963900000,
 "CpuUtilized":0.0,
 "CpuReserved":10.0,
 "MemoryUtilized":0,
 "MemoryReserved":10,
 "StorageReadBytes":0,
 "StorageWriteBytes":0,
 "NetworkRxBytes":0,
 "NetworkRxDropped":0,
 "NetworkRxErrors":0,
 "NetworkRxPackets":14,
 "NetworkTxBytes":0,
 "NetworkTxDropped":0,
 "NetworkTxErrors":0,
 "NetworkTxPackets":0
}
```

Tipo: Task

```
{
    "Version": "0",
    "Type": "Task",
```

```
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "TaskDefinitionFamily": "sleep360",
    "TaskDefinitionRevision": "1",
    "ContainerInstanceId": "0d7650e6dec34c1a9200f72098071e8f",
    "EC2InstanceId": "i-0c470579dbcdbd2f3",
    "ClusterName": "MyCluster",
    "AccountID": "637146863587",
    "Region": "us-west-2",
    "AvailabilityZone": "us-west-2b",
    "KnownStatus": "RUNNING",
    "LaunchType": "EC2",
    "PullStartedAt": 1623963608201,
    "PullStoppedAt": 1623963610065,
    "CreatedAt": 1623963607094,
    "StartedAt": 1623963610382,
    "Timestamp": 1623963900000,
    "CpuUtilized": 0.0,
    "CpuReserved": 10.0,
    "MemoryUtilized": 0,
    "MemoryReserved": 10,
    "StorageReadBytes": 0,
    "StorageWriteBytes": 0,
    "NetworkRxBytes": 0,
    "NetworkRxDropped": 0,
    "NetworkRxErrors": 0,
    "NetworkRxPackets": 14,
    "NetworkTxBytes": 0,
    "NetworkTxDropped": 0,
    "NetworkTxErrors": 0,
    "NetworkTxPackets": 0,
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "CpuUtilized",
                    "Unit": "None"
                },
                {
                    "Name": "CpuReserved",
                    "Unit": "None"
                },
```

```
                {
                    "Name": "MemoryUtilized",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "MemoryReserved",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "StorageReadBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "StorageWriteBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkRxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkTxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
                [
                    "ClusterName",
                    "TaskDefinitionFamily"
                ]
            ]
        }
    ]
}
```

Tipo: Service

```
{
    "Version": "0",
    "Type": "Service",
    "ServiceName": "myCIService",
    "ClusterName": "myCICluster",
    "Timestamp": 1561586460000,
    "DesiredTaskCount": 2,
    "RunningTaskCount": 2,
    "PendingTaskCount": 0,
    "DeploymentCount": 1,
    "TaskSetCount": 0,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "DesiredTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "RunningTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "PendingTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "DeploymentCount",
                    "Unit": "Count"
                },
                {
                    "Name": "TaskSetCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
```

```
        }
    ]
}
```

Tipo: Volume

```
{
    "Version": "0",
    "Type": "Volume",
    "TaskDefinitionFamily": "myCITaskDef",
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "ClusterName": "myCICluster",
    "ServiceName": "myCIService",
    "VolumeId": "vol-1233436545ff708cb",
    "InstanceId": "i-0c470579dbcdbd2f3",
    "LaunchType": "EC2",
    "VolumeName": "MyVolumeName",
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
                [
                    "VolumeName",
                    "TaskDefinitionFamily",
                    "ClusterName"
                ],
                [
                    "ServiceName",
                    "ClusterName"
                ]
```

```
            ]
        }
    ]
}
```

Tipo: Cluster

```
{
    "Version": "0",
    "Type": "Cluster",
    "ClusterName": "myCICluster",
    "Timestamp": 1561587300000,
    "TaskCount": 5,
    "ContainerInstanceCount": 5,
    "ServiceCount": 2,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "TaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ContainerInstanceCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ServiceCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Eventi di log delle prestazioni di Container Insights per Amazon EKS e Kubernetes

Di seguito sono elencati esempi di eventi di log delle prestazioni raccolti da Container Insights da cluster Amazon EKS e Kubernetes.

Tipo: nodo


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        }
      ],
      "Dimensions": [
```

```
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        },
        {
          "Name": "node_cpu_usage_total"
        },
        {
          "Name": "node_cpu_limit"
        },
```

```
        {
          "Unit": "Bytes",
          "Name": "node_memory_working_set"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_limit"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "/proc",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567096682364",
  "Type": "Node",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_cpu_limit": 4000,
  "node_cpu_request": 1130,
  "node_cpu_reserved_capacity": 28.249999999999996,
  "node_cpu_usage_system": 33.794636630852764,
  "node_cpu_usage_total": 136.47852169244098,
  "node_cpu_usage_user": 71.67075111567326,
  "node_cpu_utilization": 3.4119630423110245,
  "node_memory_cache": 3103297536,
  "node_memory_failcnt": 0,
  "node_memory_hierarchical_pgfault": 0,
  "node_memory_hierarchical_pgmajfault": 0,
```

```
  "node_memory_limit": 16624865280,
  "node_memory_mapped_file": 406646784,
  "node_memory_max_usage": 4230746112,
  "node_memory_pgfault": 0,
  "node_memory_pgmajfault": 0,
  "node_memory_request": 1115684864,
  "node_memory_reserved_capacity": 6.7109407818311055,
  "node_memory_rss": 798146560,
  "node_memory_swap": 0,
  "node_memory_usage": 3901444096,
  "node_memory_utilization": 6.601302600149552,
  "node_memory_working_set": 1097457664,
  "node_network_rx_bytes": 35918.392817386324,
  "node_network_rx_dropped": 0,
  "node_network_rx_errors": 0,
  "node_network_rx_packets": 157.67565245448117,
  "node_network_total_bytes": 68264.20276554905,
  "node_network_tx_bytes": 32345.80994816272,
  "node_network_tx_dropped": 0,
  "node_network_tx_errors": 0,
  "node_network_tx_packets": 154.21455923431654,
  "node_number_of_running_containers": 16,
  "node_number_of_running_pods": 13
}
```

Tipo: NodeFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_filesystem_utilization"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
```

```
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097939726",
  "Type": "NodeFS",
  "Version": "0",
  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_filesystem_available": 17298395136,
  "node_filesystem_capacity": 21462233088,
  "node_filesystem_inodes": 10484720,
  "node_filesystem_inodes_free": 10367158,
  "node_filesystem_usage": 4163837952,
  "node_filesystem_utilization": 19.400767547940255
}
```

Tipo: NodeDisk IO

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
```

```
  "Sources": [
    "cadvisor"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeDiskIO",
  "Version": "0",
  "device": "/dev/nvme0n1",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_diskio_io_service_bytes_async": 9750.505814277016,
  "node_diskio_io_service_bytes_read": 0,
  "node_diskio_io_service_bytes_sync": 230.6174506688036,
  "node_diskio_io_service_bytes_total": 9981.123264945818,
  "node_diskio_io_service_bytes_write": 9981.123264945818,
  "node_diskio_io_serviced_async": 1.153087253344018,
  "node_diskio_io_serviced_read": 0,
  "node_diskio_io_serviced_sync": 0.03603397666700056,
  "node_diskio_io_serviced_total": 1.1891212300110185,
  "node_diskio_io_serviced_write": 1.1891212300110185
}
```

Tipo: NodeNet

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeNet",
  "Version": "0",
  "interface": "eni972f6bfa9a0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_interface_network_rx_bytes": 3163.008420864309,
```

```
  "node_interface_network_rx_dropped": 0,
  "node_interface_network_rx_errors": 0,
  "node_interface_network_rx_packets": 16.575629266820258,
  "node_interface_network_total_bytes": 3518.3935157426017,
  "node_interface_network_tx_bytes": 355.385094878293,
  "node_interface_network_tx_dropped": 0,
  "node_interface_network_tx_errors": 0,
  "node_interface_network_tx_packets": 3.9997714100370625
}
```

Tipo: Pod

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_rx_bytes"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_tx_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization_over_pod_limit"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization_over_pod_limit"
        }
      ],
```

```
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_reserved_capacity"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
```

```
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "pod_number_of_container_restarts"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "Pod",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
```

```
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_cpu_limit": 200,
  "pod_cpu_request": 200,
  "pod_cpu_reserved_capacity": 5,
  "pod_cpu_usage_system": 1.4504841104992765,
  "pod_cpu_usage_total": 5.817016867430125,
  "pod_cpu_usage_user": 1.1281543081661038,
  "pod_cpu_utilization": 0.14542542168575312,
  "pod_cpu_utilization_over_pod_limit": 2.9085084337150624,
  "pod_memory_cache": 8192,
  "pod_memory_failcnt": 0,
  "pod_memory_hierarchical_pgfault": 0,
  "pod_memory_hierarchical_pgmajfault": 0,
  "pod_memory_limit": 104857600,
  "pod_memory_mapped_file": 0,
  "pod_memory_max_usage": 25268224,
  "pod_memory_pgfault": 0,
  "pod_memory_pgmajfault": 0,
  "pod_memory_request": 104857600,
  "pod_memory_reserved_capacity": 0.6307275170893897,
  "pod_memory_rss": 22777856,
  "pod_memory_swap": 0,
  "pod_memory_usage": 25141248,
  "pod_memory_utilization": 0.10988455961791709,
  "pod_memory_utilization_over_pod_limit": 17.421875,
  "pod_memory_working_set": 18268160,
  "pod_network_rx_bytes": 9880.697124714186,
  "pod_network_rx_dropped": 0,
  "pod_network_rx_errors": 0,
  "pod_network_rx_packets": 107.80005532263283,
  "pod_network_total_bytes": 10158.829201483635,
  "pod_network_tx_bytes": 278.13207676944796,
  "pod_network_tx_dropped": 0,
  "pod_network_tx_errors": 0,
  "pod_network_tx_packets": 1.146027574644318,
  "pod_number_of_container_restarts": 0,
  "pod_number_of_containers": 1,
  "pod_number_of_running_containers": 1,
  "pod_status": "Running"
```

```
}
```

Tipo: PodNet

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "PodNet",
  "Version": "0",
  "interface": "eth0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_interface_network_rx_bytes": 9880.697124714186,
  "pod_interface_network_rx_dropped": 0,
  "pod_interface_network_rx_errors": 0,
  "pod_interface_network_rx_packets": 107.80005532263283,
```

```
  "pod_interface_network_total_bytes": 10158.829201483635,
  "pod_interface_network_tx_bytes": 278.13207676944796,
  "pod_interface_network_tx_dropped": 0,
  "pod_interface_network_tx_errors": 0,
  "pod_interface_network_tx_packets": 1.146027574644318
}
```

Tipo: Container

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-NodeGroup-
sample",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "Container",
  "Version": "0",
  "container_cpu_limit": 200,
  "container_cpu_request": 200,
  "container_cpu_usage_system": 1.87958283771964,
  "container_cpu_usage_total": 6.159993652997942,
  "container_cpu_usage_user": 1.6707403001952357,
  "container_cpu_utilization": 0.15399984132494854,
  "container_memory_cache": 8192,
  "container_memory_failcnt": 0,
  "container_memory_hierarchical_pgfault": 0,
  "container_memory_hierarchical_pgmajfault": 0,
  "container_memory_limit": 104857600,
  "container_memory_mapped_file": 0,
  "container_memory_max_usage": 24580096,
  "container_memory_pgfault": 0,
  "container_memory_pgmajfault": 0,
  "container_memory_request": 104857600,
```

```
  "container_memory_rss": 22736896,
  "container_memory_swap": 0,
  "container_memory_usage": 24453120,
  "container_memory_utilization": 0.10574541028701798,
  "container_memory_working_set": 17580032,
  "container_status": "Running",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "number_of_container_restarts": 0
}
```

Tipo: ContainerFS

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
```

```
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "ContainerFS",
  "Version": "0",

  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  }
}
```

Tipo: Cluster

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
```

```
          "Name": "cluster_node_count"
        },
        {
          "Unit": "Count",
          "Name": "cluster_failed_node_count"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "Cluster",
  "Version": "0",
  "cluster_failed_node_count": 0,
  "cluster_node_count": 3
}
```

Tipo: ClusterService

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "service_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
```

```
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "ClusterService",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch",
    "service_name": "cloudwatch-agent-statsd"
  },
  "service_number_of_running_pods": 1
}
```

Tipo: ClusterNamespace

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "namespace_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
```

```
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097594160",
  "Type": "ClusterNamespace",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch"
  },
  "namespace_number_of_running_pods": 7
}
```

Campi rilevanti negli eventi di log delle prestazioni per Amazon EKS e Kubernetes

Per Amazon EKS e Kubernetes, l' CloudWatch agente containerizzato emette dati come eventi di registro delle prestazioni. Ciò consente di importare e CloudWatch archiviare dati ad alta cardinalità. CloudWatch utilizza i dati negli eventi del registro delle prestazioni per creare CloudWatch metriche aggregate a livello di cluster, nodo e pod senza la necessità di perdere dettagli granulari.

La tabella seguente elenca i campi di tali eventi di log delle prestazioni rilevanti per la raccolta dei dati di parametri di Container Insights. È possibile utilizzare CloudWatch Logs Insights per eseguire query su uno di questi campi per raccogliere dati o esaminare problemi. Per ulteriori informazioni, consulta Analizza i dati di registro con CloudWatch Logs Insights.

Type	Campo di log	Origine	Formula o note
Pod	`pod_cpu_utilization`	Calcolato	Formula: `pod_cpu_usage_total / node_cpu_limit`
Pod	`pod_cpu_usage_total`	cadvisor	

Type	Campo di log	Origine	Formula o note
	`pod_cpu_usage_total` è riportato in millicore.		
Pod	`pod_cpu_limit`	Calcolato	Formula: `sum(container_cpu_limit)` `sum(container_cpu_limit)` include pod già completati. Se uno qualsiasi dei contenitori nel pod non dispone di un limite di CPU definito, questo campo non viene visualizzato nel log di eventi. Ciò include i container init.

Type	Campo di log	Origine	Formula o note
Pod	`pod_cpu_request`	Calcolato	Formula: `sum(container_cpu_request)` `container_cpu_request` non è garantito per essere impostato. Solo quelli che sono impostati sono inclusi nella somma.
Pod	`pod_cpu_utilization_over_pod_limit`	Calcolato	Formula: `pod_cpu_usage_total / pod_cpu_limit`
Pod	`pod_cpu_reserved_capacity`	Calcolato	Formula: `pod_cpu_request / node_cpu_limit`

Type	Campo di log	Origine	Formula o note
Pod	`pod_memory_utilization`	Calcolato	Formula: `pod_memory_working_set / node_memory_limit` È la percentuale di utilizzo della memoria pod rispetto alla limitazione di memoria del nodo.
Pod	`pod_memory_working_set`	cadvisor	

Type	Campo di log	Origine	Formula o note
Pod	`pod_memory_limit`	Calcolato	Formula: `sum(container_memory_limit)` Se uno qualsiasi dei contenitori nel pod non dispone di un limite di memoria definito, questo campo non viene visualizzato nel log di eventi. Ciò include i container init.

Type	Campo di log	Origine	Formula o note
Pod	`pod_memory_request`	Calcolato	Formula: `sum(container_memory_request)` `container_memory_request` non è garantito per essere impostato. Solo quelli che sono impostati sono inclusi nella somma.

Type	Campo di log	Origine	Formula o note
Pod	`pod_memory_utilization_over_pod_limit`	Calcolato	Formula: `pod_memory_working_set / pod_memory_limit` Se uno qualsiasi dei contenitori nel pod non dispone di un limite di memoria definito, questo campo non viene visualizzato nel log di eventi. Ciò include i container init.
Pod	`pod_memory_reserved_capacity`	Calcolato	Formula: `pod_memory_request / node_memory_limit`

Type	Campo di log	Origine	Formula o note
Pod	`pod_network_tx_bytes`	Calcolato	Formula: `sum(pod_interface_network_tx_bytes)` Questi dati sono disponibili per tutte le interfacce di rete per pod. L' CloudWatch agente calcola il totale e aggiunge le regole di estrazione metriche.
Pod	`pod_network_rx_bytes`	Calcolato	Formula: `sum(pod_interface_network_rx_bytes)`
Pod	`pod_network_total_bytes`	Calcolato	Formula: `pod_network_rx_bytes + pod_network_tx_bytes`

Type	Campo di log	Origine	Formula o note
PodNet	`pod_interface_network_rx_bytes`	cadvisor	Questi dati sono byte RX di rete al secondo di un'interfaccia di rete pod.
PodNet	`pod_interface_network_tx_bytes`	cadvisor	Questi dati sono byte TX di rete al secondo di un'interfaccia di rete pod.
Container	`container_cpu_usage_total`	cadvisor	
Container	`container_cpu_limit`	cadvisor	Non garantito come impostato. Non è emesso se non è impostato.
Container	`container_cpu_request`	cadvisor	Non garantito come impostato. Non è emesso se non è impostato.
Container	`container_memory_working_set`	cadvisor	

Type	Campo di log	Origine	Formula o note
Container	`container_memory_limit`	pod	Non garantito come impostato. Non è emesso se non è impostato.
Container	`container_memory_request`	pod	Non garantito come impostato. Non è emesso se non è impostato.
Nodo	`node_cpu_utilization`	Calcolato	Formula: `node_cpu_usage_total / node_cpu_limit`
Nodo	`node_cpu_usage_total`	cadvisor	
Nodo	`node_cpu_limit`	/proc	

Type	Campo di log	Origine	Formula o note
Nodo	node_cpu_request	Calcolato	Formula: sum(pod_cpu_request) Per cronjobs, node_cpu_request include anche le richieste provenienti dai pod completati. Ciò può portare a un valore elevato per node_cpu_reserved_capacity .
Nodo	node_cpu_reserved_capacity	Calcolato	Formula: node_cpu_request / node_cpu_limit
Nodo	node_memory_utilization	Calcolato	Formula: node_memory_working_set / node_memory_limit
Nodo	node_memory_working_set	cadvisor	
Nodo	node_memory_limit	/proc	

Type	Campo di log	Origine	Formula o note
Nodo	node_memory_request	Calcolato	Formula: sum(pod_memory_request)
Nodo	node_memory_reserved_capacity	Calcolato	Formula: node_memory_request / node_memory_limit
Nodo	node_network_rx_bytes	Calcolato	Formula: sum(node_interface_network_rx_bytes)
Nodo	node_network_tx_bytes	Calcolato	Formula: sum(node_interface_network_tx_bytes)
Nodo	node_network_total_bytes	Calcolato	Formula: node_network_rx_bytes + node_network_tx_bytes
Nodo	node_number_of_running_pods	Elenco pod	
Nodo	node_number_of_running_containers	Elenco pod	

Type	Campo di log	Origine	Formula o note
NodeNet	`node_interface_network_rx_bytes`	cadvisor	Questi dati sono byte RX di rete al secondo di un'interfaccia di rete nodo di lavoro.
NodeNet	`node_interface_network_tx_bytes`	cadvisor	Questi dati sono byte TX di rete al secondo di un'interfaccia di rete nodo di lavoro.
NodeFS	`node_filesystem_capacity`	cadvisor	
NodeFS	`node_filesystem_usage`	cadvisor	
NodeFS	`node_filesystem_utilization`	Calcolato	Formula: `node_filesystem_usage / node_filesystem_capacity` Questi dati sono disponibili per nome dispositivo.
Cluster	`cluster_failed_node_count`	Server API	
Cluster	`cluster_node_count`	Server API	

Type	Campo di log	Origine	Formula o note
Servizio	`service_number_of_running_pods`	Server API	
`Namespace`	`namespace_number_of_running_pods`	Server API	

Esempi di calcolo dei parametri

Questa sezione include esempi che mostrano il modo in cui alcuni dei valori riportati nella tabella precedente sono calcolati.

Supponiamo che si disponga di un cluster nel seguente stato.

```
Node1
   node_cpu_limit = 4
   node_cpu_usage_total = 3

   Pod1
     pod_cpu_usage_total = 2

     Container1
        container_cpu_limit = 1
        container_cpu_request = 1
        container_cpu_usage_total = 0.8

     Container2
        container_cpu_limit = null
        container_cpu_request = null
        container_cpu_usage_total = 1.2

   Pod2
     pod_cpu_usage_total = 0.4

     Container3
        container_cpu_limit = 1
        container_cpu_request = 0.5
        container_cpu_usage_total = 0.4

Node2
   node_cpu_limit = 8
```

```
   node_cpu_usage_total = 1.5

   Pod3
     pod_cpu_usage_total = 1

     Container4
        container_cpu_limit = 2
        container_cpu_request = 2
        container_cpu_usage_total = 1
```

La tabella riportata di seguito mostra il modo in cui vengono calcolati i parametri della CPU del pod utilizzando questi dati.

Parametro	Formula	Pod1	Pod2	Pod3
`pod_cpu_utilization`	`pod_cpu_usage_total / node_cpu_limit`	2 / 4 = 50%	0,4 / 4 = 10%	1 / 8 = 12,5%
`pod_cpu_utilization_over_pod_limit`	`pod_cpu_usage_total / sum(container_cpu_limit)`	N/A perché il limite di CPU per `Container 2` non è definito	0,4 / 1 = 40%	1 / 2 = 50%
`pod_cpu_reserved_capacity`	`sum(container_cpu_request) / node_cpu_limit`	(1 + 0) / 4 = 25%	0,5 / 4 = 12,5%	2 / 8 = 25%

La tabella riportata di seguito mostra il modo in cui vengono calcolati i parametri della CPU del nodo utilizzando questi dati.

Parametro	Formula	Node1	Node2
`node_cpu_utilization`	`node_cpu_usage_total / node_cpu_limit`	3 / 4 = 75%	1,5 / 8 = 18,75%

Parametro	Formula	Node1	Node2
`node_cpu_reserved_capacity`	`sum(pod_cpu_request) / node_cpu_limit`	1,5 / 4 = 37,5%	2 / 8 = 25%

Monitoraggio dei parametri di Container Insights Prometheus

CloudWatch Il monitoraggio di Container Insights per Prometheus automatizza la scoperta delle metriche di Prometheus da sistemi e carichi di lavoro containerizzati. Prometheus è un kit di strumenti di monitoraggio e avvisi di sistema open source. Per ulteriori informazioni, vedere Che cos'è Prometheus? nella documentazione di Prometheus.

L'individuazione dei parametri Prometheus è supportata per cluster Amazon Elastic Container Service, Amazon Elastic Kubernetes Service e Kubernetes in esecuzione su istanze Amazon EC2. Vengono raccolti i tipi di parametri misuratore, contatore e riepilogo di Prometheus. Il supporto per i parametri dell'istogramma e è pianificato per una release imminente.

Per i cluster Amazon ECS e Amazon EKS, sono supportati sia i tipi di avvio EC2 che Fargate. Container Insights raccoglie automaticamente i parametri da diversi carichi di lavoro ed è possibile configurarlo per raccogliere parametir da qualsiasi carico di lavoro.

Puoi adottare Prometheus come metodo open source e open standard per inserire metriche personalizzate. CloudWatch L' CloudWatch agente con supporto Prometheus rileva e raccoglie le metriche di Prometheus per monitorare, risolvere i problemi e segnalare più rapidamente il degrado delle prestazioni e i guasti delle applicazioni. Ciò riduce anche il numero di strumenti di monitoraggio necessari per migliorare l'osservabilità.

Il supporto di Container Insights Prometheus pay-per-use include metriche e log, tra cui la raccolta, l'archiviazione e l'analisi. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

Pannelli di controllo predefiniti per alcuni carichi di lavoro

La soluzione Container Insights Prometheus include pannelli di controllo predefiniti per i carichi di lavoro più diffusi elencati in questa sezione. Per configurazioni di esempio per questi carichi di lavoro, consulta (Facoltativo) Impostazione dei carichi di lavoro Amazon ECS containerizzati di esempio per i test dei parametri di Prometheus e (Facoltativo) Impostazione dei carichi di lavoro Amazon EKS containerizzati di esempio per i test dei parametri di Prometheus.

È inoltre possibile configurare Container Insights per raccogliere parametri Prometheus da altri servizi e applicazioni containerizzati modificando il file di configurazione dell'agente.

Carichi di lavoro con pannelli di controllo predefiniti per cluster Amazon EKS e cluster Kubernetes in esecuzione su istanze Amazon EC2:

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy

Carichi di lavoro con pannelli di controllo predefiniti per cluster Amazon ECS:

- AWS App Mesh
- Java/JMX
- NGINX
- NGINX Plus

Configurare e configurare la raccolta dei parametri Prometheus su cluster Amazon ECS

Per raccogliere i parametri di Prometheus dai cluster Amazon ECS, puoi CloudWatch utilizzare l'agente come raccoglitore o utilizzare Distro for collector. AWS OpenTelemetry Per informazioni sull'utilizzo di Distro for collector, consulta https://aws-otel.github.io/docs/getting-started/container-insights/ecs-prometheus. AWS OpenTelemetry

Le sezioni seguenti spiegano come utilizzare l' CloudWatch agente come raccoglitore per recuperare le metriche di Prometheus. Installa l' CloudWatch agente con il monitoraggio Prometheus su cluster che eseguono Amazon ECS e, facoltativamente, puoi configurare l'agente per acquisire obiettivi aggiuntivi. Queste sezioni offrono inoltre esercitazioni facoltative per l'impostazione di carichi di lavoro di esempio da utilizzare per i test con il monitoraggio Prometheus.

Container Insights su Amazon ECS supporta le seguenti combinazioni di modalità di avvio e modalità di rete per i parametri Prometheus:

Tipo di avvio di Amazon ECS	Modalità di rete supportate
EC2 (Linux)	bridge, host e awsvpc
Fargate	awsvpc

Requisiti del gruppo di sicurezza VPC

Le regole di ingresso dei gruppi di sicurezza per i carichi di lavoro Prometheus devono aprire le porte CloudWatch Prometheus all'agente per lo scraping delle metriche di Prometheus tramite l'IP privato.

Le regole di uscita del gruppo di sicurezza per l' CloudWatch agente devono consentire all'agente di connettersi alla CloudWatch porta dei carichi di lavoro Prometheus tramite IP privato.

Argomenti

- Installa l' CloudWatch agente con la raccolta di metriche Prometheus sui cluster Amazon ECS
- Scraping di ulteriori origini Prometheus e importazione di tali parametri
- (Facoltativo) Impostazione dei carichi di lavoro Amazon ECS containerizzati di esempio per i test dei parametri di Prometheus

Installa l' CloudWatch agente con la raccolta di metriche Prometheus sui cluster Amazon ECS

Questa sezione spiega come configurare l' CloudWatch agente con il monitoraggio Prometheus in un cluster che esegue Amazon ECS. Dopo aver eseguito questa operazione, l'agente esegue automaticamente lo scraping e l'importazione dei parametri per i seguenti carichi di lavoro in esecuzione in quel cluster.

- AWS App Mesh
- Java/JMX

È inoltre possibile configurare l'agente per recuperare e importare parametri da altri carichi di lavoro e origini Prometheus.

Configurazione di ruoli IAM

Sono necessari due ruoli IAM per la definizione delle attività dell'agente. CloudWatch Se specifichi **`CreateIAMRoles=True`** nello AWS CloudFormation stack che Container Insights crei questi ruoli

per te, i ruoli verranno creati con le autorizzazioni corrette. Se si desidera crearli personalmente o utilizzare ruoli esistenti, sono necessari i seguenti ruoli e autorizzazioni.

- CloudWatch ruolo dell'agente ECS: il contenitore dell' CloudWatch agente utilizza questo ruolo. Deve includere la CloudWatchAgentServerPolicypolicy e una policy gestita dal cliente che contenga le seguenti autorizzazioni di sola lettura:
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:ListServices`
  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeServices`
  - `ecs:DescribeTasks`
  - `ecs:DescribeTaskDefinition`
- CloudWatch ruolo di esecuzione delle attività ECS dell'agente: questo è il ruolo richiesto da Amazon ECS per avviare ed eseguire i container. Assicurati che al tuo ruolo di esecuzione delle attività siano associati AmazonSSM, ReadOnlyAccess AmazonECS e le policy. TaskExecutionRolePolicy CloudWatchAgentServerPolicy Se si desidera archiviare dati più riservati utilizzabili da Amazon ECS, consulta Specifica dei dati sensibili per ulteriori informazioni.

Installa l' CloudWatch agente con il monitoraggio Prometheus utilizzando AWS CloudFormation

Lo usi AWS CloudFormation per installare l' CloudWatch agente con il monitoraggio Prometheus per i cluster Amazon ECS. L'elenco seguente mostra i parametri che verranno utilizzati nel modello AWS CloudFormation .

- ECS ClusterName: specifica il cluster Amazon ECS di destinazione.
- CreateIAMRoles: specifica **`True`** per creare nuovi ruoli per il ruolo dell'attività Amazon ECS e il ruolo di esecuzione dell'attività Amazon ECS. Specifica **`False`** per riutilizzare i ruoli esistenti.
- TaskRoleName— Se hai specificato **`True`** CreateIAMRoles, questo specifica il nome da utilizzare per il nuovo ruolo task di Amazon ECS. Se hai specificato **`False`** per CreateIAMRoles, questa opzione specifica il ruolo esistente da utilizzare come ruolo dell'attività Amazon ECS.
- ExecutionRoleName— Se hai specificato **`True`** CreateIAMRoles, questo specifica il nome da utilizzare per il nuovo ruolo di esecuzione delle attività di Amazon ECS. Se hai specificato **`False`** per CreateIAMRoles, questa opzione specifica il ruolo esistente da utilizzare come ruolo di esecuzione dell'attività Amazon ECS.

- ECS NetworkMode: se utilizzi il tipo di avvio EC2, specifica qui la modalità di rete. Deve essere **`bridge`** o **`host`**.
- ECS LaunchType: specifica o. **`fargate`** **`EC2`**
- SecurityGroupID: se l'ECS NetworkMode è**`awsvpc`**, specifica qui l'ID del gruppo di sicurezza.
- SubnetID: se l'ECS lo NetworkMode è**`awsvpc`**, specifica qui l'ID della sottorete.

Comandi di esempio

Questa sezione include AWS CloudFormation comandi di esempio per installare Container Insights con il monitoraggio Prometheus in vari scenari.

Crea uno AWS CloudFormation stack per un cluster Amazon ECS in modalità di rete bridge

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=bridge
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Crea uno AWS CloudFormation stack per un cluster Amazon ECS in modalità rete host

```
export AWS_PROFILE=your_aws_config_profile_eg_default
```

```
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=host
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name


curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Crea uno AWS CloudFormation stack per un cluster Amazon ECS in modalità di rete awsvpc

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=EC2
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
```

```
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
                 ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                 ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Crea uno AWS CloudFormation stack per un cluster Fargate in modalità di rete awsvpc

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=FARGATE
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
                 ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                 ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
```

```
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

AWS risorse create dallo stack AWS CloudFormation

La tabella seguente elenca le AWS risorse che vengono create quando si utilizza AWS CloudFormation per configurare il monitoraggio di Container Insights con Prometheus su un cluster Amazon ECS.

Tipo di risorsa	Nome risorsa	Commenti
AWS::SSM::Parameter	AmazonCloudWatch-*CW -$ ECS_CLUSTER_NAME -$ ECS_LAUNCH_TYPE -$ ECS_NETWORK_MODE AgentConfig*	Questo è l' CloudWatch agente con la definizione predefinita del formato metrico incorporato App Mesh e Java/JMX.
AWS::SSM::Parameter	AmazonCloudWatch-*$ ECS_CLUSTER_NAME -$ PrometheusConfigName ECS_LAUNCH_TYPE -$ ECS_NETWORK_MODE*	Questa è la configurazione di scraping di Prometheus.
AWS::IAM::Role	$ECS_TASK_ROLE_NAME.	Il ruolo dell'attività di Amazon ECS. Questo viene creato solo se è stato specificato **`True`** per `CREATE_IAM_ROLES` .
AWS::IAM::Role	${ECS_EXECUTION_ROLE_NAME}	Ruolo per l'esecuzione dell'attività Amazon ECS. Questo viene creato solo se è stato specificato **`True`** per `CREATE_IAM_ROLES` .
AWS::ECS::TaskDefinition	cwagent-prometheus-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	

Tipo di risorsa	Nome risorsa	Commenti
AWS::ECS::Service	cwagent-prometheus-replica-service-*$ TIPO_ECS_LAUNCH_-$* ECS_NETWORK_MODE	

Eliminazione dello AWS CloudFormation stack per l' CloudWatch agente con il monitoraggio di Prometheus

Per eliminare l' CloudWatch agente da un cluster Amazon ECS, inserisci questi comandi.

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export CLOUDFORMATION_STACK_NAME=your_cloudformation_stack_name

aws cloudformation delete-stack \
--stack-name ${CLOUDFORMATION_STACK_NAME} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Scraping di ulteriori origini Prometheus e importazione di tali parametri

L' CloudWatch agente con monitoraggio Prometheus necessita di due configurazioni per analizzare le metriche di Prometheus. Una è per le configurazioni standard Prometheus come documentato in <scrape_config> nella documentazione di Prometheus. L'altra è per la configurazione dell'agente. CloudWatch

Per i cluster Amazon ECS, le configurazioni sono integrate con il Parameter Store di AWS Systems Manager dai segreti nella definizione dell'attività Amazon ECS:

- Il segreto `PROMETHEUS_CONFIG_CONTENT` è per la configurazione di scraping di Prometheus.
- Il segreto `CW_CONFIG_CONTENT` è la configurazione CloudWatch dell'agente.

Per acquisire ulteriori fonti di metriche Prometheus e importare tali metriche in, è necessario modificare sia la configurazione dello scrape di Prometheus che la configurazione dell'agente, quindi ridistribuire l'agente con la configurazione aggiornata. CloudWatch CloudWatch

Requisiti del gruppo di sicurezza VPC

Le regole di ingresso dei gruppi di sicurezza per i carichi di lavoro Prometheus devono aprire le porte CloudWatch Prometheus all'agente per lo scraping delle metriche di Prometheus tramite l'IP privato.

Le regole di uscita del gruppo di sicurezza per l' CloudWatch agente devono consentire all'agente di connettersi alla CloudWatch porta dei carichi di lavoro Prometheus tramite IP privato.

Configurazione di Prometheus Scrape

<scrape_config>L' CloudWatch agente supporta le configurazioni scrape standard di Prometheus come documentato nella documentazione di Prometheus.https://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config È possibile modificare questa sezione per aggiornare le configurazioni già presenti in questo file e aggiungere ulteriori destinazioni di scraping Prometheus. Per impostazione predefinita, il file di configurazione campione contiene le seguenti righe di configurazione globali:

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval: definisce con quale frequenza recuperare le destinazioni.
- scrape_timeout: definisce quanto tempo attendere prima che una richiesta di scrape scada.

È inoltre possibile definire valori diversi per queste impostazioni a livello di processo, per ignorare le configurazioni globali.

Attività di scraping di Prometheus

Nei file YAML dell' CloudWatch agente sono già configurati alcuni processi di scraping predefiniti. Ad esempio, nei file YAML per Amazon ECS ad esempio `cwagent-ecs-prometheus-metric-for-bridge-host.yaml`, i processi di scraping predefiniti sono configurati nella sezione `ecs_service_discovery`.

```
"ecs_service_discovery": {
                  "sd_frequency": "1m",
                  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
                  "docker_label": {
                  },
                  "task_definition_list": [
                    {
                      "sd_job_name": "ecs-appmesh-colors",
```

```
                    "sd_metrics_ports": "9901",
                    "sd_task_definition_arn_pattern": ".*:task-definition\/.*-
ColorTeller-(white):[0-9]+",
                    "sd_metrics_path": "/stats/prometheus"
                  },
                  {
                    "sd_job_name": "ecs-appmesh-gateway",
                    "sd_metrics_ports": "9901",
                    "sd_task_definition_arn_pattern": ".*:task-definition/.*-
ColorGateway:[0-9]+",
                    "sd_metrics_path": "/stats/prometheus"
                  }
                ]
              }
```

Ciascuno di questi obiettivi predefiniti viene eliminato e le metriche vengono inviate agli eventi di registro utilizzando il formato CloudWatch metrico incorporato. Per ulteriori informazioni, consulta la pagina Incorporamento dei parametri nei log.

Gli eventi di log dei cluster Amazon ECS sono archiviati nel gruppo di log /aws/ecs/containerinsights/***cluster_name***/prometheus.

Ogni processo di scraping è contenuto in un flusso di log diverso in questo gruppo di log.

Per aggiungere una nuova destinazione di scraping, aggiungi una nuova voce nella sezione `task_definition_list` alla sezione `ecs_service_discovery` del file YAML e riavvia l'agente. Per un esempio di questo processo, vedere Esercitazione per l'aggiunta di nuove destinazioni di scraping di Prometheus: parametri del server API Prometheus.

CloudWatch configurazione dell'agente per Prometheus

Il file di configurazione CloudWatch dell'agente ha una `prometheus` sezione sotto `metrics_collected` per la configurazione dello scraping di Prometheus. Include le opzioni di configurazione seguenti:

- cluster_name: specifica il nome del cluster da aggiungere come etichetta nell'evento log. Questo campo è facoltativo. Se lo ometti, l'agente può rilevare il nome del cluster Amazon ECS.
- log_group_name: specifica il nome del gruppo di log per i parametri Prometheus sottoposti a scraping. Questo campo è facoltativo. *Se lo CloudWatch ometti, usa **/aws/ecs/containerinsights/ cluster_name /prometheus per i log dei cluster Amazon ECS.***

- prometheus_config_path: specifica il percorso del file di configurazione di scraping di Prometheus. Se il valore di questo campo inizia con `env:`, il contenuto del file di configurazione di scraping di Prometheus verrà recuperato dalla variabile di ambiente del container. Non modificare questo campo.
- ecs_service_discovery: è la sezione per specificare le configurazioni delle funzioni di individuazione automatica delle destinazioni di Amazon ECS Prometheus. Sono supportate due modalità per individuare le destinazioni Prometheus: individuazione basata sull'etichetta Docker del container o individuazione basata sull'espressione regolare dell'ARN della definizione dell'attività di Amazon ECS. Puoi utilizzare le due modalità insieme e l'agente deduplicherà gli obiettivi rilevati in base a: {private_ip}: {port}/{metrics_path CloudWatch }.

  La sezione `ecs_service_discovery` può contenere i seguenti campi:

  - `sd_frequency` è la frequenza di individuazione degli elementi di esportazione di Prometheus. Specifica un numero e un suffisso di unità. Ad esempio, `1m` per una volta al minuto o `30s` per una volta ogni 30 secondi. I suffissi di unità validi sono `ns`, `us`, `ms`, `s`, `m` e `h`.

    Questo campo è facoltativo. Il valore predefinito è 60 secondi (1 minuto).
  - `sd_target_cluster` è il nome del cluster Amazon ECS di destinazione per l'individuazione automatica. Questo campo è facoltativo. L'impostazione predefinita è il nome del cluster Amazon ECS in cui è installato l' CloudWatch agente.
  - `sd_cluster_region` è la regione del cluster Amazon ECS di destinazione. Questo campo è facoltativo. L'impostazione predefinita è la regione del cluster Amazon ECS in cui è installato l' CloudWatchagente.
  - `sd_result_file` è il percorso del file YAML per i risultati di destinazione di Prometheus. La configurazione di scraping di Prometheus farà riferimento a questo file.
  - `docker_label` è una sezione facoltativa che è possibile utilizzare per specificare la configurazione per l'individuazione dei servizi basati su etichette Docker. Se ometti questa sezione, l'individuazione basata sull'etichetta Docker non viene utilizzata. Questa sezione può contenere i seguenti campi:
    - `sd_port_label` è il nome dell'etichetta Docker del container che specifica la porta del container per i parametri Prometheus. Il valore predefinito è `ECS_PROMETHEUS_EXPORTER_PORT`. Se il contenitore non ha questa etichetta docker, l' CloudWatch agente la salterà.
    - `sd_metrics_path_label` è il nome dell'etichetta Docker del container che specifica il percorso dei parametri di Prometheus. Il valore predefinito è

ECS_PROMETHEUS_METRICS_PATH. Se il container non dispone di questa etichetta Docker, l'agente assume il percorso predefinito /metrics.

- sd_job_name_label è il nome dell'etichetta Docker del container che specifica il nome del processo di scraping di Prometheus. Il valore predefinito è job. Se il contenitore non ha questa etichetta docker, l' CloudWatch agente utilizza il nome del lavoro nella configurazione dello scrape di Prometheus.

- task_definition_list è una sezione facoltativa che è possibile utilizzare per specificare la configurazione per l'individuazione dei servizi basati sulla definizione dell'attività. Se ometti questa sezione, l'individuazione basata sulla definizione dell'attività non viene utilizzata. Questa sezione può contenere i seguenti campi:
  - sd_task_definition_arn_pattern è il modello da utilizzare per specificare le definizioni delle attività Amazon ECS da individuare. Questa è un'espressione regolare.
  - sd_metrics_ports elenca gli elementi containerPort per i parametri di Prometheus. Separa gli elementi containerPort con il punto e virgola.
  - sd_container_name_pattern specifica i nomi dei container dell'attività di Amazon ECS. Questa è un'espressione regolare.
  - sd_metrics_path specifica il percorso del parametro Prometheus. Se ometti questa opzione, l'agente assume il percorso predefinito /metrics
  - sd_job_name specifica il nome del processo di scraping di Prometheus. Se si omette questo campo, l' CloudWatch agente utilizza il nome del lavoro nella configurazione dello scrape di Prometheus.
- service_name_list_for_tasks è una sezione facoltativa che puoi utilizzare per specificare la configurazione per l'individuazione basata sul nome del servizio. Se ometti questa sezione, l'individuazione basata sul nome del servizio non viene utilizzata. Questa sezione può contenere i seguenti campi:
  - sd_service_name_pattern è il modello da utilizzare per specificare il servizio Amazon ECS in cui si trovano le attività da individuare. Questa è un'espressione regolare.
  - sd_metrics_ports Elenca gli elementi containerPort per i parametri Prometheus. Separa più elementi containerPorts con punto e virgola.
  - sd_container_name_pattern specifica i nomi dei container dell'attività di Amazon ECS. Questa è un'espressione regolare.
  - sd_metrics_path specifica il percorso dei parametri Prometheus. Se ometti questa opzione, l'agente presume che il percorso predefinito è /metrics.

    - `sd_job_name` specifica il nome del processo di scraping di Prometheus. Se si omette questo campo, l' CloudWatch agente utilizza il nome del lavoro nella configurazione dello scrape di Prometheus.

- metric_declaration: sono sezioni che specificano la matrice di log con formato metrico incorporato da generare. Esistono `metric_declaration` sezioni per ogni sorgente Prometheus da cui l'agente importa per impostazione predefinita CloudWatch. Ciascuna di queste sezioni include i seguenti campi:
    - `label_matcher` è un'espressione regolare che controlla il valore delle etichette elencate in `source_labels`. Le metriche corrispondenti sono abilitate per l'inclusione nel formato metrico incorporato inviato a. CloudWatch

      Se sono state specificate più etichette in `source_labels`, ti consigliamo di non utilizzare ^ o caratteri $ nell'espressione regolare per `label_matcher`.
    - `source_labels` specifica il valore delle etichette controllate dalla riga `label_matcher`.
    - `label_separator` specifica il separatore da utilizzare nella riga `label_matcher` se sono specificati `source_labels` multipli. Il valore predefinito è `;`. È possibile visualizzare questo valore predefinito utilizzato nella riga `label_matcher` nell'esempio seguente.
    - `metric_selectors`è un'espressione regolare che specifica le metriche da raccogliere e a cui inviare. CloudWatch
    - `dimensions`è l'elenco di etichette da utilizzare come CloudWatch dimensioni per ogni metrica selezionata.

Guarda l'esempio `metric_declaration` che segue.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system$",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

In questo esempio viene configurata una sezione di formato metrica incorporata da inviare come evento di log se sono soddisfatte le seguenti condizioni:

- Il valore di `Service` contiene `node-exporter` o `kube-dns`.
- Il valore di `Namespace` è `kube-system`.
- La metrica Prometheus `coredns_dns_request_type_count_total` contiene le etichette sia `Service` che `Namespace`.

L'evento di log inviato include la seguente sezione evidenziata:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "Namespace":"kube-system",
   "Service":"kube-dns",
   "coredns_dns_request_type_count_total":2562,
   "eks_amazonaws_com_component":"kube-dns",
   "instance":"192.168.61.254:9153",
   "job":"kubernetes-service-endpoints",
   ...
}
```

Guida dettagliata per l'individuazione automatica dei cluster Amazon ECS

Prometheus offre dozzine di meccanismi dinamici di individuazione dei servizi, come descritto in <scrape_config>. Tuttavia, non esiste un rilevamento dei servizi integrato per Amazon ECS. L' CloudWatch agente aggiunge questo meccanismo.

Quando il rilevamento del servizio Amazon ECS Prometheus è abilitato, l'agente effettua periodicamente CloudWatch le seguenti chiamate API ai frontend Amazon ECS e Amazon EC2 per recuperare i metadati delle attività ECS in esecuzione nel cluster ECS di destinazione.

```
EC2:DescribeInstances
ECS:ListTasks
ECS:ListServices
ECS:DescribeContainerInstances
ECS:DescribeServices
ECS:DescribeTasks
ECS:DescribeTaskDefinition
```

I metadati vengono utilizzati dall' CloudWatch agente per scansionare i target Prometheus all'interno del cluster ECS. L' CloudWatch agente supporta tre modalità di rilevamento dei servizi:

- Individuazione del servizio basata sull'etichetta Docker del container
- Individuazione del servizio basata su espressioni regolari dell'ARN della definizione dell'attività ECS
- Individuazione del servizio basato su espressioni regolari del nome del servizio ECS

Tutte le modalità possono essere utilizzate insieme. CloudWatch l'agente deduplica gli obiettivi rilevati in base a:. `{private_ip}:{port}/{metrics_path}`

Tutte le destinazioni rilevate vengono scritte in un file di risultati specificato dal campo di `sd_result_file` configurazione all'interno del contenitore dell' CloudWatch agente. Di seguito è riportato un file di esempio:

```
- targets:
  - 10.6.1.95:32785
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT: "9406"
    ECS_PROMETHEUS_JOB_NAME: demo-jar-ec2-bridge-dynamic
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-jar-ec2-bridge-dynamic-port
    TaskGroup: family:demo-jar-ec2-bridge-dynamic-port
    TaskRevision: "7"
```

```
    VpcId: vpc-01234567890
    container_name: demo-jar-ec2-bridge-dynamic-port
    job: demo-jar-ec2-bridge-dynamic
- targets:
  - 10.6.3.193:9404
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_B: "9404"
    ECS_PROMETHEUS_JOB_NAME: demo-tomcat-ec2-bridge-mapped-port
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-tomcat-ec2-bridge-mapped-port
    TaskGroup: family:demo-jar-tomcat-bridge-mapped-port
    TaskRevision: "12"
    VpcId: vpc-01234567890
    container_name: demo-tomcat-ec2-bridge-mapped-port
    job: demo-tomcat-ec2-bridge-mapped-port
```

Puoi integrare direttamente questo file di risultati con l'individuazione dei servizi basata su file Prometheus. Per ulteriori informazioni sull'individuazione dei servizi basata su file Prometheus, consulta <file_sd_config>.

Supponiamo che il file dei risultati sia scritto su `/tmp/cwagent_ecs_auto_sd.yaml`. La seguente configurazione di scraping Prometheus lo consumerà.

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: cwagent-ecs-file-sd-config
    sample_limit: 10000
    file_sd_configs:
      - files: [ "/tmp/cwagent_ecs_auto_sd.yaml" ]
```

L' CloudWatch agente aggiunge anche le seguenti etichette aggiuntive per le destinazioni scoperte.

- `container_name`
- `TaskDefinitionFamily`
- `TaskRevision`

- `TaskGroup`
- `StartedBy`
- `LaunchType`
- `job`
- `__metrics_path__`
- Etichette Docker

Quando il cluster ha il tipo di avvio EC2, vengono aggiunte le seguenti tre etichette.

- `InstanceType`
- `VpcId`
- `SubnetId`

> **Note**
>
> Le etichette Docker che non corrispondono all'espressione regolare `[a-zA-Z_][a-zA-Z0-9_]*` vengono filtrate. Questo corrisponde alle convenzioni Prometheus elencate in `label_name` in Configuration file (File di configurazione) nella documentazione di Prometheus.

Esempi di configurazione dell'individuazione dei servizi ECS

In questa sezione sono inclusi esempi che illustrano l'individuazione dei servizi ECS.

Esempio 1

```
"ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
  }
}
```

Questo esempio abilita l'individuazione dei servizi basata su etichette Docker. L' CloudWatch agente interrogherà i metadati delle attività ECS una volta al minuto e scriverà le destinazioni rilevate nel `/tmp/cwagent_ecs_auto_sd.yaml` file all'interno del contenitore dell'agente. CloudWatch

Il valore predefinito di `sd_port_label` nella sezione `docker_label` è `ECS_PROMETHEUS_EXPORTER_PORT`. Se un contenitore in esecuzione nelle attività ECS ha un'etichetta `ECS_PROMETHEUS_EXPORTER_PORT` docker, CloudWatch l'agente utilizza il suo valore `container port` per scansionare tutte le porte esposte del contenitore. Se esiste una corrispondenza, la porta host mappata più l'IP privato del container vengono utilizzati per costruire la destinazione dell'esportatore Prometheus nel seguente formato: `private_ip:host_port`.

Il valore predefinito di `sd_metrics_path_label` nella sezione `docker_label` è `ECS_PROMETHEUS_METRICS_PATH`. Se il container ha questa etichetta Docker, il suo valore verrà utilizzato come `__metrics_path__`. Se il container non ha questa etichetta, viene utilizzato il valore predefinito `/metrics`.

Il valore predefinito di `sd_job_name_label` nella sezione `docker_label` è `job`. Se il container ha questa etichetta Docker, il suo valore verrà aggiunto come una delle etichette per la destinazione per sostituire il nome del processo predefinito specificato nella configurazione Prometheus. Il valore di questa etichetta docker viene utilizzato come nome del flusso di log nel gruppo CloudWatch Logs log.

Esempio 2

```
"ecs_service_discovery": {
  "sd_frequency": "15s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A",
    "sd_job_name_label": "ECS_PROMETHEUS_JOB_NAME"
  }
}
```

Questo esempio abilita l'individuazione dei servizi basata su etichette Docker. L' CloudWatch agente interrogherà i metadati delle attività ECS ogni 15 secondi e scriverà le destinazioni rilevate nel file all'interno del contenitore dell'`/tmp/cwagent_ecs_auto_sd.yaml`agente. CloudWatch I container con un'etichetta Docker di `ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A` verranno scansionati. Il valore dell'etichetta Docker `ECS_PROMETHEUS_JOB_NAME` viene usato come nome del processo.

Esempio 3

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
```

```
  "task_definition_list": [
    {
      "sd_job_name": "java-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9404; 9406",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*javajmx.*:[0-9]+"
    },
    {
      "sd_job_name": "envoy-prometheus",
      "sd_metrics_path": "/stats/prometheus",
      "sd_container_name_pattern": "^envoy$",
      "sd_metrics_ports": "9901",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*appmesh.*:23"
    }
  ]
}
```

Questo esempio abilita l'individuazione del servizio basata su espressioni regolari dell'ARN della definizione dell'attività ECS. L' CloudWatch agente interrogherà i metadati delle attività ECS ogni cinque minuti e scriverà le destinazioni rilevate nel file all'interno del contenitore dell'agente. `/tmp/cwagent_ecs_auto_sd.yaml` CloudWatch

Sono definite due sezioni di espressione regolare dell'ARN di definizione dell'attività:

- Per la prima sezione, le attività ECS con `javajmx` nel loro ARN di definizione dell'attività ECS vengono filtrati per la scansione della porta del container. Se i contenitori all'interno di queste attività ECS espongono la porta del container su 9404 o 9406, la porta host mappata insieme all'IP privato del container vengono utilizzati per creare le destinazioni dell'esportatore Prometheus. Il valore di `sd_metrics_path` è impostato da `__metrics_path__` a `/metrics`. Quindi l' CloudWatch agente estrarrà le metriche di Prometheus, le metriche eliminate verranno inviate al flusso `private_ip:host_port/metrics` di log in Logs nel gruppo di `java-prometheus` log. CloudWatch `/aws/ecs/containerinsights/cluster_name/prometheus`
- Per la seconda sezione, le attività ECS con `appmesh` nel loro ARN di definizione dell'attività ECS e con `version` di `:23` vengono filtrati per la scansione della porta del container. Per i contenitori con il nome `envoy` che espongono la porta del container su `9901`, la porta host mappata insieme all'IP privato del container vengono utilizzati per creare i target dell'esportatore Prometheus. Il valore all'interno di queste attività ECS espongono la porta del container su 9404 o 9406, la porta host mappata insieme all'IP privato del container vengono utilizzati per creare le destinazioni dell'esportatore Prometheus. Il valore di `sd_metrics_path` è impostato da `__metrics_path__` a `/stats/prometheus`. Quindi l' CloudWatchagente raccoglierà le metriche di Prometheus e

le invierà al `private_ip:host_port/stats/prometheus` flusso di log in Logs nel gruppo di `envoy-prometheus` log. CloudWatch `/aws/ecs/containerinsights/cluster_name/prometheus`

Esempio 4

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "service_name_list_for_tasks": [
    {
      "sd_job_name": "nginx-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9113",
      "sd_service_name_pattern": "^nginx-.*"
    },
    {
      "sd_job_name": "haproxy-prometheus",
      "sd_metrics_path": "/stats/metrics",
      "sd_container_name_pattern": "^haproxy$",
      "sd_metrics_ports": "8404",
      "sd_service_name_pattern": ".*haproxy-service.*"
    }
  ]
}
```

Questo esempio abilita l'individuazione del servizio basata su espressioni regolari del nome del servizio ECS. L' CloudWatch agente interrogherà i metadati dei servizi ECS ogni cinque minuti e scriverà le destinazioni rilevate nel file all'interno del contenitore dell'agente. `/tmp/cwagent_ecs_auto_sd.yaml` CloudWatch

Sono definite due sezioni di espressioni regolari del nome del servizio:

- Per la prima sezione, le attività ECS associate ai servizi ECS con nomi corrispondenti all'espressione regolare `^nginx-.*` vengono filtrati per la scansione della porta del container. Se i contenitori all'interno di queste attività ECS espongono la porta del container su 9113, la porta host mappata insieme all'IP privato del container vengono utilizzati per creare le destinazioni dell'esportatore Prometheus. Il valore di `sd_metrics_path` è impostato da `__metrics_path__` a `/metrics`. Quindi l' CloudWatch agente estrarrà le metriche di Prometheus e le metriche eliminate `private_ip:host_port/metrics` verranno inviate al flusso di log in Logs nel gruppo

di `nginx-prometheus` log. CloudWatch `/aws/ecs/containerinsights/cluster_name/prometheus`

- o per la seconda sezione, le attività ECS associate ai servizi ECS con nomi corrispondenti all'espressione regolare `.*haproxy-service.*` vengono filtrati per la scansione della porta del container. Per i contenitori con il nome `haproxy` che espongono la porta del container su 8404, la porta host mappata insieme all'IP privato del container vengono utilizzati per creare i target dell'esportatore Prometheus. Il valore di `sd_metrics_path` è impostato da `__metrics_path__` a `/stats/metrics`. Quindi l' CloudWatch agente estrarrà le metriche di Prometheus e le metriche eliminate `private_ip:host_port/stats/metrics` verranno inviate al flusso di log in Logs nel gruppo di `haproxy-prometheus` log. CloudWatch `/aws/ecs/containerinsights/cluster_name/prometheus`

Esempio 5

```
"ecs_service_discovery": {
  "sd_frequency": "1m30s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "MY_PROMETHEUS_EXPORTER_PORT_LABEL",
    "sd_metrics_path_label": "MY_PROMETHEUS_METRICS_PATH_LABEL",
    "sd_job_name_label": "MY_PROMETHEUS_METRICS_NAME_LABEL"
  }
  "task_definition_list": [
    {
      "sd_metrics_ports": "9150",
      "sd_task_definition_arn_pattern": "*memcached.*"
    }
  ]
}
```

In questo esempio vengono abilitate entrambe le modalità di individuazione dei servizi ECS. L' CloudWatch agente interrogherà i metadati delle attività ECS ogni 90 secondi e scriverà le destinazioni rilevate nel file all'interno del contenitore dell'agente. `/tmp/cwagent_ecs_auto_sd.yaml` CloudWatch

Per la configurazione dell'individuazione del servizio basata su Docker:

- Le attività ECS con etichetta Docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` verranno filtrate per la scansione della porta Prometheus. La porta del container Prometheus di destinazione è specificata dal valore dell'etichetta `MY_PROMETHEUS_EXPORTER_PORT_LABEL`.
- Il valore dell'etichetta Docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` viene utilizzato per `__metrics_path__`. Se il container non ha questa etichetta Docker, viene utilizzato il valore predefinito `/metrics`.
- Il valore dell'etichetta Docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` viene usato come etichetta del processo. Se il container non dispone di questa etichetta Docker, viene utilizzato il nome del processo definito nella configurazione Prometheus.

Per la configurazione dell'individuazione del servizio basata su espressioni regolari dell'ARN della definizione dell'attività ECS:

- Le attività ECS con `memcached` nell'ARN di definizione dell'attività ECS vengono filtrati per la scansione della porta del container. La porta del container Prometheus di destinazione è 9150 come definito da `sd_metrics_ports`. Viene utilizzato il percorso dei parametri predefinito `/metrics`. Viene utilizzato il nome del processo definito nella configurazione Prometheus.

(Facoltativo) Impostazione dei carichi di lavoro Amazon ECS containerizzati di esempio per i test dei parametri di Prometheus

Per testare il supporto delle metriche Prometheus CloudWatch in Container Insights, puoi configurare uno o più dei seguenti carichi di lavoro containerizzati. L' CloudWatch agente con supporto Prometheus raccoglie automaticamente le metriche da ciascuno di questi carichi di lavoro. Per visualizzare le metriche raccolte per impostazione predefinita, vedere Metriche di Prometheus raccolte dall'agente CloudWatch .

Argomenti

- Carico di lavoro App Mesh di esempio per cluster Amazon ECS
- Carico di lavoro Java/JMX di esempio per cluster Amazon ECS
- Installazione del carico di lavoro di esempio del proxy inverso NGINX per cluster Amazon ECS
- Esempio di carico di lavoro NGINX Plus per cluster Amazon ECS
- Esercitazione per l'aggiunta di una nuova destinazione di scraping Prometheus: Memcached su Amazon ECS
- Esercitazione per eseguire lo scraping dei parametri Prometheus di Redis su Amazon ECS Fargate

Carico di lavoro App Mesh di esempio per cluster Amazon ECS

Per raccogliere parametri da un carico di lavoro Prometheus di esempio per Amazon ECS, devi eseguire Container Insights nel cluster. Per informazioni sull'installazione di Container Insights, consulta Configurazione di Container Insights su Amazon ECS.

Per prima cosa, segui questa spiegazione passo per passo per implementare l'app a colori di esempio sul cluster Amazon ECS. Dopo aver terminato, avrai i parametri di App Mesh Prometheus esposti sulla porta 9901.

Quindi, segui questi passaggi per installare l' CloudWatch agente con il monitoraggio Prometheus sullo stesso cluster Amazon ECS in cui hai installato l'app a colori. La procedura descritta in questa sezione consente di installare l' CloudWatch agente in modalità di rete bridge.

Le variabili di ambiente `ENVIRONMENT_NAME`, `AWS_PROFILE` e `AWS_DEFAULT_REGION` impostate nella spiegazione passo per passo verranno utilizzate anche nei passaggi seguenti.

Per installare l' CloudWatch agente con Prometheus Monitoring for Testing

1. Scarica il AWS CloudFormation modello inserendo il seguente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
bridge-host.yaml
```

2. Imposta la modalità di rete immettendo i seguenti comandi.

```
export ECS_CLUSTER_NAME=${ENVIRONMENT_NAME}
export ECS_NETWORK_MODE=bridge
```

3. Crea lo AWS CloudFormation stack inserendo i seguenti comandi.

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=CWAgent-Prometheus-
TaskRole-${ECS_CLUSTER_NAME} \
```

```
                  ParameterKey=ExecutionRoleName,ParameterValue=CWAgent-Prometheus-
ExecutionRole-${ECS_CLUSTER_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

4. (Facoltativo) Quando viene creato lo AWS CloudFormation stack, viene visualizzato un `CREATE_COMPLETE` messaggio. Per verificare lo stato prima di visualizzare il messaggio, inserisci il seguente comando.

```
aws cloudformation describe-stacks \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--query 'Stacks[0].StackStatus' \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Risoluzione dei problemi

I passaggi della spiegazione passo per passo usano jq per analizzare il risultato di output della AWS CLI. Per ulteriori informazioni sull'installazione di jq, consulta jq. Usa il seguente comando per impostare il formato di output predefinito della tua AWS CLI su JSON in modo che jq possa analizzarlo correttamente.

```
$ aws configure
```

Quando la risposta arriva a `Default output format`, inserisci **`json`**.

Disinstalla l' CloudWatch agente con il monitoraggio Prometheus

Al termine del test, immettete il seguente comando per disinstallare l' CloudWatchagente eliminando lo stack. AWS CloudFormation

```
aws cloudformation delete-stack \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Carico di lavoro Java/JMX di esempio per cluster Amazon ECS

JMX Exporter è un esportatore ufficiale di Prometheus che può recuperare ed esporre JMX mBeans JMX come metriche Prometheus. Per ulteriori informazioni, vedere prometheus/jmx_exporter.

L' CloudWatch agente con supporto Prometheus analizza i parametri Java/JMX Prometheus in base alla configurazione del service discovery nel cluster Amazon ECS. È possibile configurare JMX Exporter per esporre le metriche su una porta o metrics_path diverso. Se modifichi la porta o il percorso, aggiorna la sezione predefinita nella configurazione dell'agente. `ecs_service_discovery` CloudWatch

Per raccogliere parametri da un carico di lavoro Prometheus di esempio per Amazon ECS, devi eseguire Container Insights nel cluster. Per informazioni sull'installazione di Container Insights, consulta Configurazione di Container Insights su Amazon ECS.

Per installare il carico di lavoro di esempio Java/JMX per cluster Amazon ECS

1. Procedi come descritto in queste sezioni per creare le immagini Docker.
   - Esempio: immagine Docker dell'applicazione Java Jar con parametri Prometheus
   - Esempio: immagine Docker Apache Tomcat con parametri Prometheus
2. Specifica le due etichette Docker seguenti nel file di definizione dell'attività Amazon ECS. Puoi quindi eseguire la definizione dell'attività come servizio Amazon ECS o attività Amazon ECS nel cluster.
   - Imposta `ECS_PROMETHEUS_EXPORTER_PORT` per puntare all'elemento containerPort in cui sono esposti i parametri Prometheus.
   - Imposta `Java_EMF_Metrics` su `true`. L' CloudWatch agente utilizza questo flag per generare il formato metrico incorporato nell'evento di registro.

   Di seguito è riportato un esempio:

```
{
  "family": "workload-java-ec2-bridge",
  "taskRoleArn": "{{task-role-arn}}",
  "executionRoleArn": "{{execution-role-arn}}",
  "networkMode": "bridge",
  "containerDefinitions": [
    {
      "name": "tomcat-prometheus-workload-java-ec2-bridge-dynamic-port",
```

```
      "image": "your_docker_image_tag_for_tomcat_with_prometheus_metrics",
      "portMappings": [
        {
          "hostPort": 0,
          "protocol": "tcp",
          "containerPort": 9404
        }
      ],
      "dockerLabels": {
        "ECS_PROMETHEUS_EXPORTER_PORT": "9404",
        "Java_EMF_Metrics": "true"
      }
    }
  ],
  "requiresCompatibilities": [
    "EC2"  ],
  "cpu": "256",
  "memory": "512"
  }
```

L'impostazione predefinita dell' CloudWatch agente nel AWS CloudFormation modello consente sia l'individuazione dei servizi basata su etichette docker sia l'individuazione dei servizi basata sulla definizione delle attività ARN. Per visualizzare queste impostazioni predefinite, vedere la riga 65 del file di configurazione YAML dell' CloudWatch agente. I container con l'etichetta `ECS_PROMETHEUS_EXPORTER_PORT` verranno individuati automaticamente in base alla porta del container specificata per lo scraping di Prometheus.

L'impostazione predefinita dell' CloudWatch agente contiene anche l'`metric_declaration`impostazione per Java/JMX alla riga 112 dello stesso file. Tutte le etichette docker dei contenitori di destinazione verranno aggiunte come etichette aggiuntive nelle metriche di Prometheus e inviate a Logs. CloudWatch Per i container Java/JMX con etichetta Docker `Java_EMF_Metrics="true"`, verrà generato il formato della metrica incorporata.

Installazione del carico di lavoro di esempio del proxy inverso NGINX per cluster Amazon ECS

L'esportatore NGINX Prometheus può sottoporre a scraping ed esporre i dati NGINX come parametri Prometheus. Questo esempio utilizza l'esportatore in tandem con il servizio proxy inverso NGINX per Amazon ECS.

Per ulteriori informazioni sull'esportatore NGINX Prometheus, vedi su Github. nginx-prometheus-exporter Per ulteriori informazioni sul reverse proxy NGINX, vedi su Github. ecs-nginx-reverse-proxy

L' CloudWatch agente con supporto Prometheus analizza i parametri di NGINX Prometheus in base alla configurazione del service discovery nel cluster Amazon ECS. È possibile configurare NGINX dell'esportatore Prometheus per esporre i parametri su una porta o percorso diverso. Se modifichi la porta o il percorso, aggiorna la sezione nel file di configurazione dell'agente. `ecs_service_discovery` CloudWatch

Installazione del carico di lavoro di esempio del proxy inverso NGINX per cluster Amazon ECS

Procedi come segue per installare il carico di lavoro di esempio del proxy inverso NGINX.

Creazione delle immagini Docker

Per creare le immagini Docker per il carico di lavoro di esempio del proxy inverso NGINX

1. Scarica la seguente cartella dal repository del reverse proxy di NGINX: https://github.com/awslabs/ /tree/master/reverse-proxy/. ecs-nginx-reverse-proxy
2. Trova la directory `app` e crea un'immagine da quella directory:

   ```
   docker build -t web-server-app ./path-to-app-directory
   ```

3. Crea un'immagine personalizzata per NGINX. Innanzitutto, crea una directory con i due file seguenti:

   - Un file Docker di esempio:

     ```
     FROM nginx
     COPY nginx.conf /etc/nginx/nginx.conf
     ```

   - Qualsiasi file`nginx.conf`, modificato da ecs-nginx-reverse-proxyhttps://github.com/awslabs/ /tree/master/reverse-proxy/:

     ```
     events {
       worker_connections 768;
     }

     http {
       # Nginx will handle gzip compression of responses from the app server
       gzip on;
       gzip_proxied any;
       gzip_types text/plain application/json;
       gzip_min_length 1000;
     ```

```
  server{
    listen 8080;
    location /stub_status {
        stub_status   on;
    }
  }

  server {
    listen 80;

    # Nginx will reject anything not matching /api
    location /api {
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }

      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://app:3000;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

> **Note**
>
> `stub_status` deve essere abilitato nella stessa porta da cui `nginx-prometheus-exporter` è configurato per eseguire lo scraping dei parametri. Nella nostra definizione di attività di esempio, `nginx-prometheus-exporter` è configurato per eseguire lo scraping dei parametri dalla porta 8080.

4. Crea un'immagine dai file nella tua nuova directory:

```
docker build -t nginx-reverse-proxy ./path-to-your-directory
```

5. Carica le nuove immagini in un repository di immagini per utilizzarle in un secondo momento.

Creazione della definizione dell'attività per eseguire NGINX e l'app del server Web in Amazon ECS

Successivamente, imposta la definizione dell'attività.

Questa definizione dell'attività consente la raccolta e l'esportazione deli parametri Prometheus NGINX. Il container NGINX tiene traccia dell'input dall'app ed espone tali dati alla porta 8080, come impostato in `nginx.conf`. Il contenitore NGINX prometheus exporter analizza queste metriche e le invia alla porta 9113, per utilizzarle in. CloudWatch

Per impostare la definizione dell'attività per il carico di lavoro Amazon ECS di NGINX di esempio

1. Crea un file JSON di definizione dell'attività con il seguente contenuto. Sostituisci *`your-customized-nginx-iamge`*con l'URI dell'immagine per la tua immagine NGINX personalizzata e sostituisci *`your-web-server-app-image`* con l'URI dell'immagine per l'immagine dell'app del tuo server web.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 256,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
```

```
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
        "-nginx.scrape-uri",
        "http://nginx:8080/stub_status"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-sample-stack"
}
```

2. Registra la definizione dell'attività inserendo il seguente comando.

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Crea un servizio per eseguire l'attività inserendo il seguente comando:

   Assicurati di non modificare il nome del servizio. Eseguiremo un servizio CloudWatch agente utilizzando una configurazione che cerca le attività utilizzando i modelli di nomi dei servizi che le hanno avviate. Ad esempio, per consentire all' CloudWatch agente di trovare l'attività avviata da questo comando, è possibile specificare il valore di `sd_service_name_pattern` to be`^nginx-service$`. La sezione successiva offre ulteriori dettagli.

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-service \
 --task-definition nginx-sample-stack:1 \
 --desired-count 1
```

Configura l' CloudWatch agente per acquisire le metriche di NGINX Prometheus

Il passaggio finale consiste nel configurare l'agente per l'analisi delle metriche NGINX CloudWatch . In questo esempio, l' CloudWatch agente rileva l'attività tramite il modello del nome del servizio e la porta 9113, dove l'esportatore espone le metriche di Prometheus per NGINX. Una volta individuata l'attività e disponibili le metriche, l'agente inizia a pubblicare le metriche raccolte nel CloudWatch flusso di log. nginx-prometheus-exporter

Per configurare l' CloudWatch agente per l'analisi delle metriche NGINX

1. Scarica la versione più recente del file YAML necessario immettendo il seguente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml
```

2. Apri il file con un editor di testo e trova la configurazione completa dell' CloudWatch agente nella chiave della sezione. `value` `resource:CWAgentConfigSSMParameter` Quindi, nella sezione `ecs_service_discovery` aggiungi la seguente sezione `service_name_list_for_tasks`.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-service$"
   }
],
```

3. Nello stesso file aggiungi la sezione seguente nella sezione `metric_declaration` per consentire i parametri NGINX. Assicurati di seguire il modello di rientro esistente.

```
{
  "source_labels": ["job"],
  "label_matcher": ".*nginx.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginx_.*$"
  ]
},
```

4. Se l' CloudWatch agente non è già distribuito in questo cluster, vai al passaggio 8.

   Se hai già distribuito l' CloudWatch agente nel cluster Amazon ECS utilizzando AWS CloudFormation, puoi creare un set di modifiche inserendo i seguenti comandi:

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-scraping-support
```

5. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.

6. Rivedi il changeset appena creato. nginx-scraping-support Dovresti vedere una modifica applicata alla risorsa CW SSMParameter. AgentConfig Esegui il changeset e riavvia l'attività dell' CloudWatch agente inserendo il seguente comando:

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Attendi circa 10 secondi e inserisci il comando seguente.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
```

```
--region $AWS_REGION
```

8. Se stai installando l' CloudWatch agente con la raccolta delle metriche di Prometheus sul cluster per la prima volta, inserisci i seguenti comandi.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Visualizzazione dei parametri e dei log di NGINX

Ora puoi visualizzare i parametri NGINX raccolti.

Per visualizzare i parametri per il carico di lavoro NGINX

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nella regione in cui è in esecuzione il cluster, scegli Metrics (Parametri) nel pannello di navigazione sinistro. Trova lo spazio dei nomi ContainerInsights/Prometheus per visualizzare le metriche.
3. Per visualizzare gli eventi di CloudWatch Logs, scegli Log groups nel riquadro di navigazione. *Gli eventi si trovano nel gruppo di log* ***/aws/containerinsights/your_cluster_name /prometheus****, nel flusso di log. nginx-prometheus-exporter*

Esempio di carico di lavoro NGINX Plus per cluster Amazon ECS

NGINX Plus è la versione commerciale di NGINX. Per utilizzarla, è necessario disporre di una licenza. Per ulteriori informazioni, consulta NGINX Plus

L'esportatore NGINX Prometheus può sottoporre a scraping ed esporre i dati NGINX come parametri Prometheus. Questo esempio utilizza l'esportatore in tandem con il servizio proxy inverso NGINX Plus per Amazon ECS.

Per ulteriori informazioni sull'esportatore NGINX Prometheus, vedi su Github. nginx-prometheus-exporter Per ulteriori informazioni sul reverse proxy NGINX, vedi su Github. ecs-nginx-reverse-proxy

L' CloudWatch agente con supporto Prometheus analizza i parametri di NGINX Plus Prometheus in base alla configurazione del service discovery nel cluster Amazon ECS. È possibile configurare NGINX dell'esportatore Prometheus per esporre i parametri su una porta o percorso diverso. Se modifichi la porta o il percorso, aggiorna la sezione nel file di configurazione dell'agente. `ecs_service_discovery` CloudWatch

Installazione del carico di lavoro di esempio del proxy inverso NGINX Plus per cluster Amazon ECS

Procedi come segue per installare il carico di lavoro di esempio del proxy inverso NGINX.

Creazione delle immagini Docker

Per creare le immagini Docker per il carico di lavoro di esempio del proxy inverso NGINX Plus

1. Scarica la seguente cartella dal repository del reverse proxy di NGINX: https://github.com/awslabs/ /tree/master/reverse-proxy/. ecs-nginx-reverse-proxy
2. Trova la directory `app` e crea un'immagine da quella directory:

   ```
   docker build -t web-server-app ./path-to-app-directory
   ```

3. Creazione di un'immagine personalizzata per NGINX Plus. Prima di poter creare l'immagine per NGINX Plus, è necessario ottenere la chiave denominata `nginx-repo.key` e il certificato SSL `nginx-repo.crt` per la tua licenza NGINX Plus. Crea una directory e archivia in essa i tuoi file `nginx-repo.key` e `nginx-repo.crt`.

   Nella directory appena creata, crea i due file seguenti:

   - Un Dockerfile di esempio con il seguente contenuto. Questo file docker è adottato da un file di esempio fornito all'indirizzo https://docs.nginx.com/nginx/admin-guide/installing-nginx/

installing-nginx-docker/#docker_plus_image. La modifica importante che apportiamo è che carichiamo un file separato, chiamato `nginx.conf`, che verrà creato nel passaggio successivo.

```
FROM debian:buster-slim

LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.com>“

# Define NGINX versions for NGINX Plus and NGINX Plus modules
# Uncomment this block and the versioned nginxPackages block in the main RUN
# instruction to install a specific release
# ENV NGINX_VERSION 21
# ENV NJS_VERSION 0.3.9
# ENV PKG_RELEASE 1~buster

# Download certificate and key from the customer portal (https://cs.nginx.com
 (https://cs.nginx.com/))
# and copy to the build context
COPY nginx-repo.crt /etc/ssl/nginx/
COPY nginx-repo.key /etc/ssl/nginx/
# COPY nginx.conf /etc/ssl/nginx/nginx.conf

RUN set -x \
# Create nginx user/group first, to be consistent throughout Docker variants
&& addgroup --system --gid 101 nginx \
&& adduser --system --disabled-login --ingroup nginx --no-create-home --home /
nonexistent --gecos "nginx user" --shell /bin/false --uid 101 nginx \
&& apt-get update \
&& apt-get install --no-install-recommends --no-install-suggests -y ca-
certificates gnupg1 \
&& \
NGINX_GPGKEY=573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62; \
found=''; \
for server in \
ha.pool.sks-keyservers.net (http://ha.pool.sks-keyservers.net/) \
hkp://keyserver.ubuntu.com:80 \
hkp://p80.pool.sks-keyservers.net:80 \
pgp.mit.edu (http://pgp.mit.edu/) \
; do \
echo "Fetching GPG key $NGINX_GPGKEY from $server"; \
apt-key adv --keyserver "$server" --keyserver-options timeout=10 --recv-keys
 "$NGINX_GPGKEY" && found=yes && break; \
done; \
```

```
test -z "$found" && echo >&2 "error: failed to fetch GPG key $NGINX_GPGKEY" &&
 exit 1; \
apt-get remove --purge --auto-remove -y gnupg1 && rm -rf /var/lib/apt/lists/* \
# Install the latest release of NGINX Plus and/or NGINX Plus modules
# Uncomment individual modules if necessary
# Use versioned packages over defaults to specify a release
&& nginxPackages=" \
nginx-plus \
# nginx-plus=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-xslt \
# nginx-plus-module-xslt=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-geoip \
# nginx-plus-module-geoip=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-image-filter \
# nginx-plus-module-image-filter=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-perl \
# nginx-plus-module-perl=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-njs \
# nginx-plus-module-njs=${NGINX_VERSION}+${NJS_VERSION}-${PKG_RELEASE} \
" \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Peer \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Host \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslCert \"/etc/ssl/nginx/nginx-
repo.crt\";" >> /etc/apt/apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslKey \"/etc/ssl/nginx/nginx-
repo.key\";" >> /etc/apt/apt.conf.d/90nginx \
&& printf "deb https://plus-pkgs.nginx.com/debian buster nginx-plus\n" > /etc/
apt/sources.list.d/nginx-plus.list \
&& apt-get update \
&& apt-get install --no-install-recommends --no-install-suggests -y \
$nginxPackages \
gettext-base \
curl \
&& apt-get remove --purge --auto-remove -y && rm -rf /var/lib/apt/lists/* /etc/
apt/sources.list.d/nginx-plus.list \
&& rm -rf /etc/apt/apt.conf.d/90nginx /etc/ssl/nginx

# Forward request logs to Docker log collector
RUN ln -sf /dev/stdout /var/log/nginx/access.log \
&& ln -sf /dev/stderr /var/log/nginx/error.log

COPY nginx.conf /etc/nginx/nginx.conf
```

```
EXPOSE 80

STOPSIGNAL SIGTERM

CMD ["nginx", "-g", "daemon off;"]
```

- Un `nginx.conf` file, modificato da https://github.com/awslabs/ ecs-nginx-reverse-proxy /tree/master/reverse-proxy/nginx.

```
events {
  worker_connections 768;
}

http {
  # Nginx will handle gzip compression of responses from the app server
  gzip on;
  gzip_proxied any;
  gzip_types text/plain application/json;
  gzip_min_length 1000;

  upstream backend {
    zone name 10m;
    server app:3000    weight=2;
    server app2:3000    weight=1;
  }

  server{
    listen 8080;
    location /api {
      api write=on;
    }
  }

  match server_ok {
    status 100-599;
  }

  server {
    listen 80;
    status_zone zone;
    # Nginx will reject anything not matching /api
    location /api {
```

```
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }

      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://backend;
      health_check uri=/lorem-ipsum match=server_ok;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

4. Crea un'immagine dai file nella tua nuova directory:

```
docker build -t nginx-plus-reverse-proxy ./path-to-your-directory
```

5. Carica le nuove immagini in un repository di immagini per utilizzarle in un secondo momento.

Creazione della definizione dell'attività per eseguire NGINX Plus e l'app del server Web in Amazon ECS

Successivamente, imposta la definizione dell'attività.

Questa definizione dell'attività consente la raccolta e l'esportazione deli parametri Prometheus NGINX Plus. Il container NGINX tiene traccia dell'input dall'app ed espone tali dati alla porta 8080, come impostato in `nginx.conf`. Il contenitore NGINX prometheus exporter analizza queste metriche e le invia alla porta 9113, per utilizzarle in. CloudWatch

Per impostare la definizione dell'attività per il carico di lavoro Amazon ECS di NGINX di esempio

1. Crea un file JSON di definizione dell'attività con il seguente contenuto. *Sostituisci your-customized-nginx-plus-image con l'URI dell'immagine per la tua immagine NGINX Plus personalizzata e sostituisci your-web-server-app -image con l'URI dell'immagine per l'immagine dell'app del server web.*

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-plus-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app",
        "app2"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "app2",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
        "-nginx.plus",
        "-nginx.scrape-uri",
         "http://nginx:8080/api"
```

```
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-plus-sample-stack"
}
```

2. Registra la definizione dell'attività:

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Crea un servizio per eseguire l'attività inserendo il seguente comando:

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-plus-service \
 --task-definition nginx-plus-sample-stack:1 \
 --desired-count 1
```

Assicurati di non modificare il nome del servizio. Eseguiremo un servizio CloudWatch agente utilizzando una configurazione che cerca le attività utilizzando i modelli di nomi dei servizi che le hanno avviate. Ad esempio, per consentire all' CloudWatch agente di trovare l'attività avviata da questo comando, è possibile specificare il valore di `sd_service_name_pattern` to be`^nginx-plus-service$`. La sezione successiva offre ulteriori dettagli.

Configura l' CloudWatch agente per acquisire le metriche di NGINX Plus Prometheus

Il passaggio finale consiste nel configurare l'agente per l'analisi delle metriche NGINX CloudWatch . In questo esempio, l' CloudWatch agente rileva l'attività tramite il modello del nome del servizio e la porta 9113, dove l'esportatore espone le metriche di Prometheus per NGINX. Una volta individuata

l'attività e disponibili le metriche, l'agente inizia a pubblicare le metriche raccolte nel CloudWatch flusso di log. nginx-prometheus-exporter

Per configurare l' CloudWatch agente per l'analisi delle metriche NGINX

1. Scarica la versione più recente del file YAML necessario immettendo il seguente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml
```

2. Apri il file con un editor di testo e trova la configurazione completa dell' CloudWatch agente nella chiave della sezione. `value` `resource:CWAgentConfigSSMParameter` Quindi, nella sezione `ecs_service_discovery` aggiungi la seguente sezione `service_name_list_for_tasks`.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-plus-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-plus.*"
   }
],
```

3. Nello stesso file aggiungi la sezione seguente nella sezione `metric_declaration` per consentire i parametri NGINX Plus. Assicurati di seguire il modello di rientro esistente.

```
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginxplus_connections_accepted$",
    "^nginxplus_connections_active$",
    "^nginxplus_connections_dropped$",
    "^nginxplus_connections_idle$",
    "^nginxplus_http_requests_total$",
    "^nginxplus_ssl_handshakes$",
    "^nginxplus_ssl_handshakes_failed$",
    "^nginxplus_up$",
    "^nginxplus_upstream_server_health_checks_fails$"
```

```
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName",
 "upstream"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_response_time$"
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName", "code"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_responses$",
    "^nginxplus_server_zone_responses$"
  ]
},
```

4. Se l' CloudWatch agente non è già distribuito in questo cluster, vai al passaggio 8.

   Se hai già distribuito l' CloudWatch agente nel cluster Amazon ECS utilizzando AWS CloudFormation, puoi creare un set di modifiche inserendo i seguenti comandi:

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
```

```
    --region $AWS_REGION \
    --change-set-name nginx-plus-scraping-support
```

5. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.

6. Rivedi il changeset appena creato. nginx-plus-scraping-support Dovresti vedere una modifica applicata alla risorsa CW SSMParameter. AgentConfig Eseguite il changeset e riavviate l'attività dell' CloudWatch agente immettendo il seguente comando:

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Attendi circa 10 secondi e inserisci il comando seguente.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

8. Se stai installando l' CloudWatch agente con la raccolta delle metriche di Prometheus sul cluster per la prima volta, inserisci i seguenti comandi.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
```

```
    --region $AWS_REGION
```

Visualizzazione deli parametri e dei log di NGINX Plus

Ora puoi visualizzare i parametri NGINX Plus raccolti.

Per visualizzare i parametri per il carico di lavoro NGINX

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nella regione in cui è in esecuzione il cluster, scegli Metrics (Parametri) nel pannello di navigazione sinistro. Trova lo spazio dei nomi ContainerInsights/Prometheus per visualizzare le metriche.
3. Per visualizzare gli eventi di CloudWatch Logs, scegli Log groups nel riquadro di navigazione. *Gli eventi si trovano nel gruppo di log* ***/aws/containerinsights/your_cluster_name /prometheus****, nel flusso di log. nginx-plus-prometheus-exporter*

Esercitazione per l'aggiunta di una nuova destinazione di scraping Prometheus: Memcached su Amazon ECS

Questa esercitazione offre un'introduzione pratica per eseguire lo scraping dei parametri Prometheus di un esempio di applicazione Memcached su un cluster Amazon ECS con il tipo di avvio EC2.
Il target dell'esportatore Memcached Prometheus verrà scoperto automaticamente dall'agente mediante il rilevamento del servizio basato sulla definizione delle attività ECS. CloudWatch

Memcached è un sistema di caching in memoria implementato per scopo generico. Viene spesso utilizzato per accelerare i siti Web dinamici basati su database, memorizzando nella cache dati e oggetti nella RAM per ridurre il numero di volte in cui un'origine dati esterna (ad esempio un database o un'API) deve essere letta. Per ulteriori informazioni, consulta la pagina Cos'è Memcached?

memchached_exporter (Apache License 2.0) è uno degli esportatori Prometheus ufficiali. Per impostazione predefinita, memcache_exporter serve sulla porta 0.0.0.0:9150 in `/metrics`.

In questa esercitazione vengono utilizzate le immagini Docker nei due repository Docker Hub seguenti:

- Memcached
- prom/memcached-exporter

Prerequisito

Per raccogliere parametri da un carico di lavoro Prometheus di esempio per Amazon ECS, devi eseguire Container Insights nel cluster. Per informazioni sull'installazione di Container Insights, consulta Configurazione di Container Insights su Amazon ECS.

Argomenti

- Impostazione delle variabili di ambiente del cluster EC2 di Amazon ECS
- Installazione del carico di lavoro Memcached di esempio
- Configurare l' CloudWatch agente per acquisire le metriche di Memcached Prometheus
- Visualizzazione dei parametri Memcached

Impostazione delle variabili di ambiente del cluster EC2 di Amazon ECS

Per impostare le variabili di ambiente del cluster EC2 di Amazon ECS

1. Installa la CLI di Amazon ECS, se non l'hai già fatto. Per ulteriori informazioni, consulta Installazione della CLI di Amazon ECS.
2. Imposta il nuovo nome del cluster Amazon ECS e la nuova regione. Per esempio:

```
ECS_CLUSTER_NAME=ecs-ec2-memcached-tutorial
AWS_DEFAULT_REGION=ca-central-1
```

3. (Facoltativo) Se non disponi già di un cluster Amazon ECS con il tipo di avvio EC2 in cui desideri installare il carico di lavoro e l' CloudWatch agente Memcached di esempio, puoi crearne uno inserendo il seguente comando.

```
ecs-cli up --capability-iam --size 1 \
--instance-type t3.medium \
--cluster $ECS_CLUSTER_NAME \
--region $AWS_REGION
```

   Il risultato previsto di questo comando è il seguente:

```
WARN[0000] You will not be able to SSH into your EC2 instances without a key pair.
INFO[0000] Using recommended Amazon Linux 2 AMI with ECS Agent 1.44.4 and Docker
 version 19.03.6-ce
INFO[0001] Created cluster                               cluster=ecs-ec2-memcached-
tutorial region=ca-central-1
```

```
INFO[0002] Waiting for your cluster resources to be created...
INFO[0002] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0063] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0124] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Security Group created: sg-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Installazione del carico di lavoro Memcached di esempio

Per installare il carico di lavoro Memcached di esempio che espone i parametri Prometheus

1. Scarica il modello Memcached AWS CloudFormation inserendo il seguente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/memcached/memcached-traffic-sample.yaml
```

2. Imposta i nomi dei ruoli IAM da creare per Memcached inserendo i comandi riportati di seguito.

```
MEMCACHED_ECS_TASK_ROLE_NAME=memcached-prometheus-demo-ecs-task-role-name
MEMCACHED_ECS_EXECUTION_ROLE_NAME=memcached-prometheus-demo-ecs-execution-role-name
```

3. Installa il carico di lavoro Memcached di esempio inserendo il seguente comando. Questo esempio installa il carico di lavoro in modalità di rete `host`.

```
MEMCACHED_ECS_NETWORK_MODE=host

aws cloudformation create-stack --stack-name Memcached-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-EC2-$MEMCACHED_ECS_NETWORK_MODE \
    --template-body file://memcached-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=ECSNetworkMode,ParameterValue=
$MEMCACHED_ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=
$MEMCACHED_ECS_TASK_ROLE_NAME \
```

```
                  ParameterKey=ExecutionRoleName,ParameterValue=
$MEMCACHED_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Lo AWS CloudFormation stack crea quattro risorse:

- Un ruolo dell'attività ECS
- Un ruolo di esecuzione dell'attività ECS
- Una definizione dell'attività Memcached
- Un servizio di Memcached

Nella definizione dell'attività Memcached vengono definiti due container:

- Il container primario esegue una semplice applicazione Memcached e apre la porta 11211 per l'accesso.
- L'altro container esegue il processo dell'esportatore Redis per esporre i parametri Prometheus sulla porta 9150. Questo è il contenitore che deve essere scoperto e raschiato dall' CloudWatch agente.

Configurare l' CloudWatch agente per acquisire le metriche di Memcached Prometheus

Per configurare l' CloudWatch agente per l'acquisizione delle metriche di Memcached Prometheus

1. Scarica la versione più recente di `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` inserendo il seguente comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
   cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
   awsvpc.yaml
   ```

2. Apri il file con un editor di testo e trova la configurazione completa dell' CloudWatch agente dietro la chiave nella sezione. `value` `resource:CWAgentConfigSSMParameter`

   Quindi, nella sezione `ecs_service_discovery`, aggiungi la configurazione seguente nella sezione `task_definition_list`.

   ```
   {
   ```

```
    "sd_job_name": "ecs-memcached",
    "sd_metrics_ports": "9150",
    "sd_task_definition_arn_pattern": ".*:task-definition/memcached-prometheus-
demo.*:[0-9]+"
},
```

Per la sezione `metric_declaration`, l'impostazione predefinita non consente alcun parametro Memcached. Aggiungi la sezione seguente per consentire i parametri Memcached. Assicurati di seguire il modello di rientro esistente.

```
{
  "source_labels": ["container_name"],
  "label_matcher": "memcached-exporter-.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily"]],
  "metric_selectors": [
    "^memcached_current_(bytes|items|connections)$",
    "^memcached_items_(reclaimed|evicted)_total$",
    "^memcached_(written|read)_bytes_total$",
    "^memcached_limit_bytes$",
    "^memcached_commands_total$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "memcached-exporter-.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily","status","command"],
 ["ClusterName", "TaskDefinitionFamily","command"]],
  "metric_selectors": [
    "^memcached_commands_total$"
  ]
},
```

3. Se hai già distribuito l' CloudWatch agente nel cluster Amazon ECS da AWS CloudFormation, puoi creare un set di modifiche inserendo i seguenti comandi.

```
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
```

```
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name memcached-scraping-support
```

4. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.

5. Esamina il set di modifiche `memcached-scraping-support` appena creato. Dovresti vedere una modifica applicata alla risorsa `CWAgentConfigSSMParameter`. Esegui il changeset e riavvia l'attività dell' CloudWatch agente inserendo i seguenti comandi.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

6. Attendi circa 10 secondi e inserisci il comando seguente.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Se stai installando l' CloudWatch agente con la raccolta delle metriche Prometheus per il cluster per la prima volta, inserisci i seguenti comandi:

```
ECS_NETWORK_MODEE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
```

```
                ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Visualizzazione dei parametri Memcached

Questo tutorial invia le seguenti metriche allo spazio dei nomi ECS/ ContainerInsights /Prometheus in. CloudWatch Puoi usare la console per visualizzare le metriche in quel namespace CloudWatch .

Nome parametro	Dimensioni
memcached_current_items	ClusterName , TaskDefinitionFamily
memcached_current_connections	ClusterName , TaskDefinitionFamily
memcached_limit_bytes	ClusterName , TaskDefinitionFamily
memcached_current_bytes	ClusterName , TaskDefinitionFamily
memcached_written_bytes_total	ClusterName , TaskDefinitionFamily
memcached_read_bytes_total	ClusterName , TaskDefinitionFamily
memcached_items_evicted_total	ClusterName , TaskDefinitionFamily

Nome parametro	Dimensioni	
memcached_items_reclaimed_total	ClusterName , TaskDefinitionFamily	
memcached_commands_total	ClusterName , TaskDefinitionFamily ClusterName , comando TaskDefinitionFamily ClusterName , TaskDefinitionFamily, status, comando	

> **Note**
>
> I valori della dimensione command (comando) possono essere: `delete`, `get`, `cas`, `set`, `decr`, `touch`, `incr` o `flush`.
> I valori della dimensione status (stato) possono essere `hit`, `miss` o `badval`.

Puoi anche creare una CloudWatch dashboard per le metriche di Memcached Prometheus.

Per creare un pannello di controllo per i parametri Prometheus di Memcached

1. Crea variabili di ambiente, sostituendo i valori sotto in modo che corrispondano all'implementazione.

```
DASHBOARD_NAME=your_memcached_cw_dashboard_name
ECS_TASK_DEF_FAMILY=memcached-prometheus-demo-$ECS_CLUSTER_NAME-EC2-
$MEMCACHED_ECS_NETWORK_MOD
```

2. Inserisci il seguente comando per creare il pannello di controllo.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/ecs-task-definition-templates/deployment-mode/
replica-service/cwagent-prometheus/sample_cloudwatch_dashboards/memcached/
cw_dashboard_memcached.json \
| sed "s/{{YOUR_AWS_REGION}}/$AWS_REGION/g" \
```

```
| sed "s/{{YOUR_CLUSTER_NAME}}/$ECS_CLUSTER_NAME/g" \
| sed "s/{{YOUR_TASK_DEF_FAMILY}}/$ECS_TASK_DEF_FAMILY/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --region
 $AWS_REGION --dashboard-body
```

Esercitazione per eseguire lo scraping dei parametri Prometheus di Redis su Amazon ECS Fargate

Questa esercitazione offre un'introduzione pratica per recuperare i parametri Prometheus di un'applicazione Redis di esempio in un cluster Amazon ECS Fargate. Il target dell'esportatore Redis Prometheus verrà scoperto automaticamente dall' CloudWatch agente con il supporto metrico Prometheus basato sulle etichette docker del contenitore.

Redis (https://redis.io/) è uno store di strutture dati open source (con licenza BSD), in memoria, utilizzato come database, cache e broker di messaggi. Per ulteriori informazioni, consulta la pagina redis.

redis_exporter (con licenza MIT Licence) viene utilizzato per esporre i parametri Prometheus Redis sulla porta specificata (predefinita: 0.0.0.0:9121). Per ulteriori informazioni, consulta la pagina redis_exporter.

In questa esercitazione vengono utilizzate le immagini Docker nei due repository Docker Hub seguenti:

- redis
- redis_exporter

Prerequisito

Per raccogliere parametri da un carico di lavoro Prometheus di esempio per Amazon ECS, devi eseguire Container Insights nel cluster. Per informazioni sull'installazione di Container Insights, consulta Configurazione di Container Insights su Amazon ECS.

Argomenti

- Impostazione della variabile di ambiente del cluster Fargate di Amazon ECS
- Impostazione delle variabili di ambiente di rete per il cluster Fargate di Amazon ECS
- Installazione del carico di lavoro Redis di esempio
- Configura l' CloudWatch agente per acquisire le metriche di Redis Prometheus

- Visualizzazione dei parametri Redis

Impostazione della variabile di ambiente del cluster Fargate di Amazon ECS

Per impostare la variabile di ambiente del cluster Fargate di Amazon ECS

1. Installa la CLI di Amazon ECS, se non l'hai già fatto. Per ulteriori informazioni, consulta Installazione della CLI di Amazon ECS.
2. Imposta il nuovo nome del cluster Amazon ECS e la nuova regione. Per esempio:

```
ECS_CLUSTER_NAME=ecs-fargate-redis-tutorial
AWS_DEFAULT_REGION=ca-central-1
```

3. (Facoltativo) Se non disponi già di un cluster Amazon ECS Fargate in cui desideri installare il carico di lavoro CloudWatch e l'agente Redis di esempio, puoi crearne uno inserendo il seguente comando.

```
ecs-cli up --capability-iam \
--cluster $ECS_CLUSTER_NAME \
--launch-type FARGATE \
--region $AWS_DEFAULT_REGION
```

Il risultato previsto di questo comando è il seguente:

```
INFO[0000] Created cluster   cluster=ecs-fargate-redis-tutorial region=ca-central-1
INFO[0001] Waiting for your cluster resources to be created...
INFO[0001] Cloudformation stack status   stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Impostazione delle variabili di ambiente di rete per il cluster Fargate di Amazon ECS

Per impostare le variabili di ambiente di rete per il cluster Fargate di Amazon ECS

1. Imposta il VPC e l'ID della sottorete del cluster Amazon ECS. Se hai stato creato un nuovo cluster nella procedura precedente, questi valori verranno visualizzati nel risultato del comando finale. In caso contrario, utilizza gli ID del cluster esistente che intendi utilizzare con Redis.

```
ECS_CLUSTER_VPC=vpc-xxxxxxxxxxxxxxxxx
ECS_CLUSTER_SUBNET_1=subnet-xxxxxxxxxxxxxxxxx
ECS_CLUSTER_SUBNET_2=subnet-xxxxxxxxxxxxxxxxx
```

2. In questo tutorial, installeremo l'applicazione Redis e l' CloudWatchagente nel gruppo di sicurezza predefinito del VPC del cluster Amazon ECS. Il gruppo di sicurezza predefinito consente tutte le connessioni di rete all'interno dello stesso gruppo di sicurezza in modo che l' CloudWatch agente possa acquisire le metriche Prometheus esposte sui contenitori Redis. In un ambiente di produzione reale, potresti voler creare gruppi di sicurezza dedicati per l'applicazione e l' CloudWatch agente Redis e impostare autorizzazioni personalizzate per tali gruppi.

   Per ottenere l'ID del gruppo di sicurezza predefinito, inserisci il comando seguente.

   ```
   aws ec2 describe-security-groups \
   --filters Name=vpc-id,Values=$ECS_CLUSTER_VPC  \
   --region $AWS_DEFAULT_REGION
   ```

   Quindi imposta la variabile del gruppo di sicurezza predefinito del cluster Fargate inserendo il seguente comando, sostituendolo *my-default-security-group*con il valore trovato dal comando precedente.

   ```
   ECS_CLUSTER_SECURITY_GROUP=my-default-security-group
   ```

Installazione del carico di lavoro Redis di esempio

Per installare il carico di lavoro Redis di esempio che espone i parametri Prometheus

1. Scarica il AWS CloudFormation modello Redis inserendo il seguente comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
   cwagent-prometheus/sample_traffic/redis/redis-traffic-sample.yaml
   ```

2. Imposta i nomi dei ruoli IAM da creare per Redis inserendo i comandi riportati di seguito.

   ```
   REDIS_ECS_TASK_ROLE_NAME=redis-prometheus-demo-ecs-task-role-name
   REDIS_ECS_EXECUTION_ROLE_NAME=redis-prometheus-demo-ecs-execution-role-name
   ```

3. Installa il carico di lavoro Redis di esempio inserendo il seguente comando.

```
aws cloudformation create-stack --stack-name Redis-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-fargate-awsvpc \
    --template-body file://redis-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET_1 \
                 ParameterKey=TaskRoleName,ParameterValue=$REDIS_ECS_TASK_ROLE_NAME
 \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$REDIS_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_DEFAULT_REGION
```

Lo AWS CloudFormation stack crea quattro risorse:

- Un ruolo dell'attività ECS
- Un ruolo di esecuzione dell'attività ECS
- Una definizione dell'attività Redis
- Un servizio Redis

Nella definizione dell'attività Redis vengono definiti due container:

- Il container primario esegue una semplice applicazione Redis e apre la porta 6379 per l'accesso.
- L'altro container esegue il processo dell'esportatore Redis per esporre i parametri Prometheus sulla porta 9121. Questo è il contenitore che deve essere scoperto e raschiato dall' CloudWatch agente. La seguente etichetta docker è definita in modo che l' CloudWatch agente possa scoprire questo contenitore in base ad essa.

```
ECS_PROMETHEUS_EXPORTER_PORT: 9121
```

Configura l' CloudWatch agente per acquisire le metriche di Redis Prometheus

Per configurare l' CloudWatch agente per l'acquisizione delle metriche di Redis Prometheus

1. Scarica la versione più recente di `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` inserendo il seguente comando.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml
```

2. Apri il file con un editor di testo e trova la configurazione completa dell' CloudWatch agente dietro la chiave nella sezione. `value` `resource:CWAgentConfigSSMParameter`

   Poi, nella sezione `ecs_service_discovery` mostrata qui, l'individuazione del servizio basata su `docker_label` è abilitata con le impostazioni predefinite che sono basate su `ECS_PROMETHEUS_EXPORTER_PORT`, che corrisponde all'etichetta Docker che abbiamo definito nella definizione dell'attività Redis ECS. Quindi non abbiamo bisogno di apportare modifiche in questa sezione:

```
ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
*  "docker_label": {
  },*
  ...
```

   Per la sezione `metric_declaration`, l'impostazione predefinita non consente alcun parametro Redis. Aggiungi la sezione seguente per consentire i parametri Redis. Assicurati di seguire il modello di rientro esistente.

```
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily"]],
  "metric_selectors": [
    "^redis_net_(in|out)put_bytes_total$",
    "^redis_(expired|evicted)_keys_total$",
    "^redis_keyspace_(hits|misses)_total$",
    "^redis_memory_used_bytes$",
```

```
    "^redis_connected_clients$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","cmd"]],
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","db"]],
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

3. Se hai già distribuito l' CloudWatch agente nel cluster Amazon ECS da AWS CloudFormation, puoi creare un set di modifiche inserendo i seguenti comandi.

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
```

```
    --change-set-name redis-scraping-support
```

4. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.

5. Esamina il set di modifiche `redis-scraping-support` appena creato. Dovresti vedere una modifica applicata alla risorsa `CWAgentConfigSSMParameter`. Esegui il changeset e riavvia l'attività dell' CloudWatch agente inserendo i seguenti comandi.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

6. Attendi circa 10 secondi e inserisci il comando seguente.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

7. Se stai installando l' CloudWatch agente con la raccolta delle metriche Prometheus per il cluster per la prima volta, inserisci i seguenti comandi:

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
```

```
    --region ${AWS_DEFAULT_REGION}
```

Visualizzazione dei parametri Redis

Questo tutorial invia le seguenti metriche allo spazio dei nomi ECS/ ContainerInsights /Prometheus in. CloudWatch Puoi usare la console per visualizzare le metriche in quel namespace CloudWatch .

Nome parametro	Dimensioni	
`redis_net_input_bytes_total`	ClusterName, `TaskDefinitionFamily`	
`redis_net_output_bytes_total`	ClusterName, `TaskDefinitionFamily`	
`redis_expired_keys_total`	ClusterName, `TaskDefinitionFamily`	
`redis_evicted_keys_total`	ClusterName, `TaskDefinitionFamily`	
`redis_keyspace_hits_total`	ClusterName, `TaskDefinitionFamily`	
`redis_keyspace_misses_total`	ClusterName, `TaskDefinitionFamily`	
`redis_memory_used_bytes`	ClusterName, `TaskDefinitionFamily`	
`redis_connected_clients`	ClusterName, `TaskDefinitionFamily`	

Nome parametro	Dimensioni	
`redis_commands_total`	`ClusterName` , `TaskDefinitionFamily` , `cmd`	
`redis_db_keys`	`ClusterName` , `TaskDefinitionFamily` , `db`	

> **Note**
>
> Il valore della dimensione cmd può essere `append`, `client`, `command`, `config`, `dbsize`, `flushall`, `get`, `incr`, `info`, `latency` o `slowlog`.
> Il valore della dimensione db può essere da `db0` a `db15`.

Puoi anche creare una CloudWatch dashboard per le metriche di Redis Prometheus.

Per creare un pannello di controllo per i parametri Prometheus di Redis

1. Crea variabili di ambiente, sostituendo i valori sotto in modo che corrispondano all'implementazione.

```
DASHBOARD_NAME=your_cw_dashboard_name
ECS_TASK_DEF_FAMILY=redis-prometheus-demo-$ECS_CLUSTER_NAME-fargate-awsvpc
```

2. Inserisci il seguente comando per creare il pannello di controllo.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Impostazione e configurazione della raccolta dei parametri Prometheus su cluster Amazon EKS e Kubernetes

Per raccogliere i parametri di Prometheus dai cluster che eseguono Amazon EKS o Kubernetes, puoi utilizzare l'agente come raccoglitore o utilizzare CloudWatch Distro for collector. AWS OpenTelemetry Per informazioni sull'utilizzo di Distro for collector, consulta https://aws-otel.github.io/docs/getting-started/container-insights/eks-prometheus. AWS OpenTelemetry

Le sezioni seguenti spiegano come raccogliere le metriche di Prometheus utilizzando l'agente. CloudWatch Spiegano come installare l' CloudWatch agente con il monitoraggio Prometheus su cluster che eseguono Amazon EKS o Kubernetes e come configurare l'agente per acquisire obiettivi aggiuntivi. Offrono inoltre esercitazioni facoltative per l'impostazione di carichi di lavoro di esempio da utilizzare per i test con il monitoraggio Prometheus.

Argomenti

Installa l' CloudWatch agente con la raccolta di metriche Prometheus sui cluster Amazon EKS e Kubernetes

Questa sezione spiega come configurare l' CloudWatch agente con il monitoraggio Prometheus in un cluster che esegue Amazon EKS o Kubernetes. Dopo aver eseguito questa operazione, l'agente esegue automaticamente lo scraping e l'importazione dei parametri per i seguenti carichi di lavoro in esecuzione in quel cluster.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

È inoltre possibile configurare l'agente per recuperare e importare carichi di lavoro e origini Prometheus.

Prima di seguire questi passaggi per installare l' CloudWatch agente per la raccolta di metriche Prometheus, devi avere un cluster in esecuzione su Amazon EKS o un cluster Kubernetes in esecuzione su un'istanza Amazon EC2.

Requisiti del gruppo di sicurezza VPC

Le regole di ingresso dei gruppi di sicurezza per i carichi di lavoro Prometheus devono aprire le porte CloudWatch Prometheus all'agente per lo scraping delle metriche di Prometheus tramite l'IP privato.

Le regole di uscita del gruppo di sicurezza per l' CloudWatch agente devono consentire all'agente di connettersi alla CloudWatch porta dei carichi di lavoro Prometheus tramite IP privato.

Argomenti

Installa l' CloudWatch agente con la raccolta di metriche Prometheus sui cluster Amazon EKS e Kubernetes

Questa sezione spiega come configurare l' CloudWatch agente con il monitoraggio Prometheus in un cluster che esegue Amazon EKS o Kubernetes. Dopo aver eseguito questa operazione, l'agente esegue automaticamente lo scraping e l'importazione dei parametri per i seguenti carichi di lavoro in esecuzione in quel cluster.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

È inoltre possibile configurare l'agente per recuperare e importare carichi di lavoro e origini Prometheus.

Prima di seguire questi passaggi per installare l' CloudWatch agente per la raccolta di metriche Prometheus, devi avere un cluster in esecuzione su Amazon EKS o un cluster Kubernetes in esecuzione su un'istanza Amazon EC2.

Requisiti del gruppo di sicurezza VPC

Le regole di ingresso dei gruppi di sicurezza per i carichi di lavoro Prometheus devono aprire le porte CloudWatch Prometheus all'agente per lo scraping delle metriche di Prometheus tramite l'IP privato.

Le regole di uscita del gruppo di sicurezza per l' CloudWatch agente devono consentire all'agente di connettersi alla CloudWatch porta dei carichi di lavoro Prometheus tramite IP privato.

Argomenti

Impostazione dei ruoli IAM

Il primo passo consiste nell'impostare il ruolo IAM necessario nel cluster. Esistono due metodi:

- Impostare un ruolo IAM per un account di servizio, noto anche come ruolo di servizio. Questo metodo funziona sia per il tipo di avvio EC2 che per il tipo di avvio Fargate.
- Aggiungere una policy IAM al ruolo IAM utilizzato per il cluster. Funziona solo per il tipo di avvio EC2.

Impostazione di un ruolo di servizio (tipo di avvio EC2 e tipo di avvio Fargate)

Per impostare un ruolo di servizio, immetti il comando seguente. Sostituisci *MyCluster*con il nome del cluster.

```
eksctl create iamserviceaccount \
 --name cwagent-prometheus \
--namespace amazon-cloudwatch \
 --cluster MyCluster \
--attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
--approve \
--override-existing-serviceaccounts
```

Aggiunta di una policy al ruolo IAM del cluster (solo tipo di avvio EC2)

Per impostare la policy IAM in un cluster per il supporto di Prometheus

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nel riquadro di navigazione, seleziona Istanze.
3. È necessario trovare il prefisso del nome del ruolo IAM per il cluster. A tale scopo, seleziona la casella di controllo accanto al nome di un'istanza presente nel cluster e scegli Azioni, Impostazioni istanza, Allega/Sostituisci ruolo IAM. Copia quindi il prefisso del ruolo IAM, ad esempio `eksctl-dev303-workshop-nodegroup`.
4. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
5. Nel riquadro di navigazione, seleziona Ruoli.
6. Utilizza la casella di ricerca per trovare il prefisso copiato in precedenza in questa procedura e scegli il ruolo desiderato.
7. Scegli Collega policy.
8. Usa la casella di ricerca per trovare CloudWatchAgentServerPolicy. Seleziona la casella di controllo accanto a CloudWatchAgentServerPolicye scegli Allega politica.

Installazione dell' CloudWatchagente per raccogliere le metriche di Prometheus

È necessario installare l' CloudWatch agente nel cluster per raccogliere le metriche. L'installazione dell'agente è diversa per cluster Amazon EKS e cluster Kubernetes.

Eliminare le versioni precedenti dell' CloudWatch agente con il supporto Prometheus

Se hai già installato una versione dell' CloudWatch agente con supporto Prometheus nel tuo cluster, devi eliminare quella versione immettendo il seguente comando. Questo è necessario solo per le versioni precedenti dell'agente con supporto di Prometheus. Non è necessario eliminare l' CloudWatch agente che abilita Container Insights senza il supporto di Prometheus.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

Installazione dell' CloudWatch agente su cluster Amazon EKS con il tipo di avvio EC2

Per installare l' CloudWatch agente con supporto Prometheus su un cluster Amazon EKS, segui questi passaggi.

Per installare l' CloudWatch agente con supporto Prometheus su un cluster Amazon EKS

1. Immetti il comando seguente per verificare se lo spazio dei nomi `amazon-cloudwatch` è già stato creato:

```
kubectl get namespace
```

2. Se `amazon-cloudwatch` non viene visualizzato nei risultati, crearlo immettendo il seguente comando:

```
kubectl create namespace amazon-cloudwatch
```

3. Per distribuire l'agente con la configurazione predefinita e fare in modo che invii i dati alla AWS regione in cui è installato, inserisci il seguente comando:

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
service/cwagent-prometheus/prometheus-eks.yaml
```

   Per fare in modo che l'agente invii dati a un'area diversa, attenersi alla seguente procedura:

   a. Scaricare il file YAML per l'agente immettendo il seguente comando:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
service/cwagent-prometheus/prometheus-eks.yaml
```

   b. Apri il file con un editor di testo e cercare il blocco `cwagentconfig.json` del file.

   c. Aggiungere le linee evidenziate, specificando l'area desiderata:

```
cwagentconfig.json: |
    {
      "agent": {
        "region": "us-east-2"
      },
      "logs": { ...
```

   d. Salva il file e implementa l'agente utilizzando il file aggiornato.

```
kubectl apply -f prometheus-eks.yaml
```

Installazione dell' CloudWatch agente su cluster Amazon EKS con il tipo di lancio Fargate

Per installare l' CloudWatch agente con supporto Prometheus su un cluster Amazon EKS con il tipo di avvio Fargate, segui questi passaggi.

Per installare l' CloudWatch agente con supporto Prometheus su un cluster Amazon EKS con il tipo di avvio Fargate

1. Immettete il seguente comando per creare un profilo Fargate per l' CloudWatch agente in modo che possa essere eseguito all'interno del cluster. Sostituisci *MyCluster*con il nome del cluster.

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --name amazon-cloudwatch \
   --namespace amazon-cloudwatch
   ```

2. Per installare l' CloudWatch agente, immettere il seguente comando. Sostituisci *MyCluster*con il nome del cluster. Questo nome viene utilizzato nel nome del gruppo di log che memorizza gli eventi di log raccolti dall'agente e viene utilizzato anche come dimensione per le metriche raccolte dall'agente.

   Sostituisci *region* (regione) con il nome dell'area in cui desideri inviare i parametri. Ad esempio, `us-west-1`.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
   prometheus/prometheus-eks-fargate.yaml |
   sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
   kubectl apply -f -
   ```

Installazione dell' CloudWatch agente su un cluster Kubernetes

Per installare l' CloudWatch agente con supporto Prometheus su un cluster che esegue Kubernetes, inserisci il seguente comando:

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-k8s.yaml |
sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
kubectl apply -f -
```

Sostituisci *MyCluster*con il nome del cluster. Questo nome viene utilizzato nel nome del gruppo di log che memorizza gli eventi di log raccolti dall'agente e viene utilizzato anche come dimensione per le metriche raccolte dall'agente.

Sostituisci la *regione* con il nome della AWS regione in cui desideri inviare le metriche. Ad esempio, **us-west-1**.

Verifica dell'esecuzione dell'agente

In entrambi i cluster Amazon EKS e Kubernetes, puoi inserire il comando seguente per confermare che l'agente è in esecuzione.

```
kubectl get pod -l "app=cwagent-prometheus" -n amazon-cloudwatch
```

Se i risultati includono un singolo pod di agenti `Running` nello stato, l' CloudWatch agente sta eseguendo e raccogliendo le metriche di Prometheus. Per impostazione predefinita, l' CloudWatchagente raccoglie le metriche per App Mesh, NGINX, Memcached, Java/JMX e HAProxy ogni minuto. Per ulteriori informazioni su questi parametri, consulta Metriche di Prometheus raccolte dall'agente CloudWatch . Per istruzioni su come visualizzare le metriche di Prometheus in, consulta CloudWatch Visualizzazione dei parametri Prometheus

Puoi anche configurare l' CloudWatch agente per raccogliere metriche da altri esportatori di Prometheus. Per ulteriori informazioni, consulta la pagina Scraping di ulteriori origini Prometheus e importazione di tali parametri.

Scraping di ulteriori origini Prometheus e importazione di tali parametri

L' CloudWatch agente con monitoraggio Prometheus necessita di due configurazioni per analizzare le metriche di Prometheus. Una è per le configurazioni standard Prometheus come documentato in <scrape_config> nella documentazione di Prometheus. L'altra è per la configurazione dell'agente. CloudWatch

Per i cluster Amazon EKS, le configurazioni sono definite in `prometheus-eks.yaml` (per il tipo di avvio EC2) o `prometheus-eks-fargate.yaml` (per il tipo di avvio Fargate) come due mappe di configurazione:

- La sezione `name: prometheus-config` contiene le impostazioni per lo scraping di Prometheus.
- La `name: prometheus-cwagentconfig` sezione contiene la configurazione per l' CloudWatch agente. Puoi utilizzare questa sezione per configurare la modalità di raccolta delle metriche di

Prometheus da. CloudWatch Ad esempio, si specifica in quali metriche importare e si definiscono le relative dimensioni CloudWatch.

Per i cluster Kubernetes in esecuzione su istanze Amazon EC2, le configurazioni sono definite nel file YAML `prometheus-k8s.yaml` come due mappe di configurazione:

- La sezione `name: prometheus-config` contiene le impostazioni per lo scraping di Prometheus.
- La `name: prometheus-cwagentconfig` sezione contiene la configurazione per l' CloudWatch agente.

Per acquisire ulteriori fonti di metriche Prometheus e importare tali metriche in, è necessario modificare sia la configurazione dello scrape di Prometheus che la configurazione dell'agente, quindi ridistribuire l'agente con la configurazione aggiornata. CloudWatch CloudWatch

Requisiti del gruppo di sicurezza VPC

Le regole di ingresso dei gruppi di sicurezza per i carichi di lavoro Prometheus devono aprire le porte CloudWatch Prometheus all'agente per lo scraping delle metriche di Prometheus tramite l'IP privato.

Le regole di uscita del gruppo di sicurezza per l' CloudWatch agente devono consentire all'agente di connettersi alla CloudWatch porta dei carichi di lavoro Prometheus tramite IP privato.

Configurazione di Prometheus Scrape

<scrape_config>L' CloudWatch agente supporta le configurazioni scrape standard di Prometheus come documentato nella documentazione di Prometheus.https://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config È possibile modificare questa sezione per aggiornare le configurazioni già presenti in questo file e aggiungere ulteriori destinazioni di scraping Prometheus. Per impostazione predefinita, il file di configurazione campione contiene le seguenti righe di configurazione globali:

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval: definisce con quale frequenza recuperare le destinazioni.
- scrape_timeout: definisce quanto tempo attendere prima che una richiesta di scrape scada.

È inoltre possibile definire valori diversi per queste impostazioni a livello di processo, per ignorare le configurazioni globali.

Attività di scraping di Prometheus

Nei file YAML dell' CloudWatch agente sono già configurati alcuni processi di scraping predefiniti. Ad esempio, in `prometheus-eks.yaml`, i processi di scraping predefiniti sono configurati nelle righe `job_name` nella sezione `scrape_configs`. In questo file, la sezione `kubernetes-pod-jmx` predefinita seguente recupera i parametri di JMX Exporter.

```
- job_name: 'kubernetes-pod-jmx'
   sample_limit: 10000
   metrics_path: /metrics
   kubernetes_sd_configs:
   - role: pod
   relabel_configs:
   - source_labels: [__address__]
     action: keep
     regex: '.*:9404$'
   - action: labelmap
     regex: __meta_kubernetes_pod_label_(.+)
   - action: replace
     source_labels:
     - __meta_kubernetes_namespace
     target_label: Namespace
   - source_labels: [__meta_kubernetes_pod_name]
     action: replace
     target_label: pod_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_container_name
     target_label: container_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_controller_name
     target_label: pod_controller_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_controller_kind
     target_label: pod_controller_kind
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_phase
```

```
        target_label: pod_phase
```

Ciascuno di questi obiettivi predefiniti viene eliminato e le metriche vengono inviate agli eventi di registro utilizzando il formato CloudWatch metrico incorporato. Per ulteriori informazioni, consulta Incorporamento dei parametri nei log.

Gli eventi di log dei cluster Amazon EKS e Kubernetes sono archiviati nel gruppo di log ***/aws/containerinsights/ cluster_name /prometheus*** *in Logs.* CloudWatch Gli eventi di log dei cluster Amazon ECS sono archiviati nel gruppo di log /aws/ecs/containerinsights/***cluster_name***/prometheus.

Ogni processo di scraping è contenuto in un flusso di log diverso in questo gruppo di log. Ad esempio, il processo di scraping Prometheus `kubernetes-pod-appmesh-envoy` è definito per App Mesh. ***Tutte le metriche App Mesh Prometheus dei cluster Amazon EKS e Kubernetes vengono inviate al flusso di log denominato /aws/containerinsights/ cluster_name >prometheus/.*** kubernetes-pod-appmesh-envoy

Per aggiungere una nuova destinazione di scraping, aggiungere una nuova sezione `job_name` alla sezione `scrape_configs` del file YAML e riavviare l'agente. Per un esempio di questo processo, vedere Esercitazione per l'aggiunta di nuove destinazioni di scraping di Prometheus: parametri del server API Prometheus.

CloudWatch configurazione dell'agente per Prometheus

Il file di configurazione CloudWatch dell'agente ha una `prometheus` sezione sotto `metrics_collected` per la configurazione dello scraping di Prometheus. Include le opzioni di configurazione seguenti:

- cluster_name: specifica il nome del cluster da aggiungere come etichetta nell'evento log. Questo campo è facoltativo. Se li ometti, l'agente può rilevare il nome del cluster Amazon EKS o Kubernetes.
- log_group_name: specifica il nome del gruppo di log per i parametri Prometheus sottoposti a scraping. Questo campo è facoltativo. Se lo CloudWatch ometti, utilizza /aws/containerinsights/***cluster_name /prometheus per i log dei cluster*** Amazon EKS e Kubernetes.
- prometheus_config_path: specifica il percorso del file di configurazione di scraping di Prometheus. Se il valore di questo campo inizia con `env:`, il contenuto del file di configurazione di scraping di Prometheus verrà recuperato dalla variabile di ambiente del container. Non modificare questo campo.

- ecs_service_discovery: è la sezione per specificare la configurazione per l'individuazione del servizio Amazon ECS Prometheus. Per ulteriori informazioni, consulta la pagina Guida dettagliata per l'individuazione automatica dei cluster Amazon ECS.

  La sezione `ecs_service_discovery` può contenere i seguenti campi:

  - `sd_frequency` è la frequenza di individuazione degli elementi di esportazione di Prometheus. Specifica un numero e un suffisso di unità. Ad esempio, `1m` per una volta al minuto o `30s` per una volta ogni 30 secondi. I suffissi di unità validi sono `ns`, `us`, `ms`, `s`, `m` e `h`.

    Questo campo è facoltativo. Il valore predefinito è 60 secondi (1 minuto).

  - `sd_target_cluster` è il nome del cluster Amazon ECS di destinazione per l'individuazione automatica. Questo campo è facoltativo. L'impostazione predefinita è il nome del cluster Amazon ECS in cui è installato l' CloudWatch agente.

  - `sd_cluster_region` è la regione del cluster Amazon ECS di destinazione. Questo campo è facoltativo. L'impostazione predefinita è la regione del cluster Amazon ECS in cui è installato l' CloudWatch agente.

  - `sd_result_file` è il percorso del file YAML per i risultati di destinazione di Prometheus. La configurazione di scraping di Prometheus farà riferimento a questo file.

  - `docker_label` è una sezione facoltativa che è possibile utilizzare per specificare la configurazione per l'individuazione dei servizi basati su etichette Docker. Se ometti questa sezione, l'individuazione basata sull'etichetta Docker non viene utilizzata. Questa sezione può contenere i seguenti campi:

    - `sd_port_label` è il nome dell'etichetta Docker del container che specifica la porta del container per i parametri Prometheus. Il valore predefinito è `ECS_PROMETHEUS_EXPORTER_PORT`. Se il contenitore non ha questa etichetta docker, l' CloudWatch agente la salterà.

    - `sd_metrics_path_label` è il nome dell'etichetta Docker del container che specifica il percorso dei parametri di Prometheus. Il valore predefinito è `ECS_PROMETHEUS_METRICS_PATH`. Se il container non dispone di questa etichetta Docker, l'agente assume il percorso predefinito `/metrics`.

    - `sd_job_name_label` è il nome dell'etichetta Docker del container che specifica il nome del processo di scraping di Prometheus. Il valore predefinito è `job`. Se il contenitore non ha questa etichetta docker, l' CloudWatch agente utilizza il nome del lavoro nella configurazione dello scrape di Prometheus.

- `task_definition_list` è una sezione facoltativa che è possibile utilizzare per specificare la configurazione per l'individuazione dei servizi basati sulla definizione dell'attività. Se ometti questa sezione, l'individuazione basata sulla definizione dell'attività non viene utilizzata. Questa sezione può contenere i seguenti campi:
  - `sd_task_definition_arn_pattern` è il modello da utilizzare per specificare le definizioni delle attività Amazon ECS da individuare. Questa è un'espressione regolare.
  - `sd_metrics_ports` elenca gli elementi containerPort per i parametri di Prometheus. Separa gli elementi containerPort con il punto e virgola.
  - `sd_container_name_pattern` specifica i nomi dei container dell'attività di Amazon ECS. Questa è un'espressione regolare.
  - `sd_metrics_path` specifica il percorso del parametro Prometheus. Se ometti questa opzione, l'agente assume il percorso predefinito `/metrics`
  - `sd_job_name` specifica il nome del processo di scraping di Prometheus. Se si omette questo campo, l' CloudWatch agente utilizza il nome del lavoro nella configurazione dello scrape di Prometheus.
- metric_declaration: sono sezioni che specificano la matrice di log con formato metrico incorporato da generare. Esistono `metric_declaration` sezioni per ogni sorgente Prometheus da cui l'agente importa per impostazione predefinita CloudWatch . Ciascuna di queste sezioni include i seguenti campi:
  - `label_matcher` è un'espressione regolare che controlla il valore delle etichette elencate in `source_labels`. Le metriche corrispondenti sono abilitate per l'inclusione nel formato metrico incorporato inviato a. CloudWatch

    Se sono state specificate più etichette in `source_labels`, ti consigliamo di non utilizzare ^ o caratteri $ nell'espressione regolare per `label_matcher`.
  - `source_labels` specifica il valore delle etichette controllate dalla riga `label_matcher`.
  - `label_separator` specifica il separatore da utilizzare nella riga `label_matcher` se sono specificati `source_labels` multipli. Il valore predefinito è `;`. È possibile visualizzare questo valore predefinito utilizzato nella riga `label_matcher` nell'esempio seguente.
  - `metric_selectors`è un'espressione regolare che specifica le metriche da raccogliere e a cui inviare. CloudWatch
  - `dimensions`è l'elenco di etichette da utilizzare come CloudWatch dimensioni per ogni metrica selezionata.

Guarda l'esempio `metric_declaration` che segue.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

In questo esempio viene configurata una sezione di formato metrica incorporata da inviare come evento di log se sono soddisfatte le seguenti condizioni:

- Il valore di `Service` contiene `node-exporter` o `kube-dns`.
- Il valore di `Namespace` è `kube-system`.
- La metrica Prometheus `coredns_dns_request_type_count_total` contiene le etichette sia `Service` che `Namespace`.

L'evento di log inviato include la seguente sezione evidenziata:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
```

```
    ],
    "Namespace":"kube-system",
    "Service":"kube-dns",
    "coredns_dns_request_type_count_total":2562,
    "eks_amazonaws_com_component":"kube-dns",
    "instance":"192.168.61.254:9153",
    "job":"kubernetes-service-endpoints",
    ...
}
```

Esercitazione per l'aggiunta di nuove destinazioni di scraping di Prometheus: parametri del server API Prometheus

Kubernetes API Server espone le metriche Prometheus sugli endpoint per impostazione predefinita. L'esempio ufficiale per la configurazione di scraping Kubernetes API Server è disponibile su Github.

Il seguente tutorial mostra come eseguire i seguenti passaggi per iniziare a importare le metriche del Kubernetes API Server in: CloudWatch

- Aggiungere la configurazione di scraping Prometheus per Kubernetes API Server al file YAML dell'agente. CloudWatch
- Configurazione delle definizioni delle metriche in formato metrico incorporato nel file YAML dell'agente. CloudWatch
- (Facoltativo) Creazione di una CloudWatch dashboard per le metriche del Kubernetes API Server.

> **Note**
>
> Kubernetes API Server espone le metriche misuratore, contatore, istogramma e riepilogo. In questa versione del supporto alle metriche di Prometheus CloudWatch , importa solo le metriche con i tipi gauge, counter e summary.

Per iniziare a raccogliere le metriche di Prometheus del server API Kubernetes in CloudWatch

1. Scaricare la versione più recente del file `prometheus-eks.yaml`, `prometheus-eks-fargate.yaml` o `prometheus-k8s.yaml` immettendo uno dei seguenti comandi.

   Per un cluster Amazon EKS con tipo di avvio EC2, immetti il seguente comando:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-eks.yaml
```

Per un cluster Amazon EKS con tipo di avvio Fargate, immetti il seguente comando:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-eks-fargate.yaml
```

Per un cluster Kubernetes in esecuzione su un'istanza Amazon EC2, inserisci il comando seguente:

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-k8s.yaml
```

2. Apri il file con un editor di testo, trova la sezione `prometheus-config` e aggiungi la sezione seguente all'interno di tale sezione. Salvare quindi le modifiche:

```
    # Scrape config for API servers
    - job_name: 'kubernetes-apiservers'
      kubernetes_sd_configs:
        - role: endpoints
          namespaces:
            names:
              - default
      scheme: https
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        insecure_skip_verify: true
      bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      relabel_configs:
      - source_labels: [__meta_kubernetes_service_name,
 __meta_kubernetes_endpoint_port_name]
        action: keep
        regex: kubernetes;https
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
```

```
        target_label: Namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_service_name
        target_label: Service
```

3. Mentre hai ancora il file YAML aperto nell'editor di testo, trova la sezione `cwagentconfig.json`. Aggiungi la seguente sottosezione e salva le modifiche. Questa sezione inserisce le metriche del server API nell'elenco degli agenti consentiti. CloudWatch All'elenco degli agenti consentiti vengono aggiunti tre tipi di parametri del server API:

   - Conteggi di oggetti etcd
   - Metriche del controller di registrazione del server API
   - Metriche delle richieste del server API

```
{"source_labels": ["job", "resource"],
  "label_matcher": "^kubernetes-apiservers;(services|daemonsets.apps|
deployments.apps|configmaps|endpoints|secrets|serviceaccounts|replicasets.apps)",
  "dimensions": [["ClusterName","Service","resource"]],
  "metric_selectors": [
  "^etcd_object_counts$"
  ]
},
{"source_labels": ["job", "name"],
   "label_matcher": "^kubernetes-apiservers;APIServiceRegistrationController$",
   "dimensions": [["ClusterName","Service","name"]],
   "metric_selectors": [
   "^workqueue_depth$",
   "^workqueue_adds_total$",
   "^workqueue_retries_total$"
  ]
},
{"source_labels": ["job","code"],
  "label_matcher": "^kubernetes-apiservers;2[0-9]{2}$",
  "dimensions": [["ClusterName","Service","code"]],
  "metric_selectors": [
   "^apiserver_request_total$"
  ]
},
{"source_labels": ["job"],
  "label_matcher": "^kubernetes-apiservers",
```

```
  "dimensions": [["ClusterName","Service"]],
  "metric_selectors": [
  "^apiserver_request_total$"
  ]
},
```

4. Se l' CloudWatch agente con supporto Prometheus è già distribuito nel cluster, devi eliminarlo inserendo il seguente comando:

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

5. Implementa l' CloudWatch agente con la configurazione aggiornata immettendo uno dei seguenti comandi. Per un cluster Amazon EKS con tipo di avvio EC2, inserisci:

```
kubectl apply -f prometheus-eks.yaml
```

   Per un cluster Amazon EKS con tipo di avvio Fargate, immetti il seguente comando. *MyCluster*Sostituisci una *regione* con valori corrispondenti alla tua distribuzione.

```
cat prometheus-eks-fargate.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

   Per un cluster Kubernetes, immetti il seguente comando. *MyCluster*Sostituisci una *regione* con valori corrispondenti alla tua distribuzione.

```
cat prometheus-k8s.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Una volta fatto questo, dovresti vedere un nuovo flusso di log chiamato kubernetes-apiservers nel gruppo di log /aws/containerinsights/***cluster_name***/prometheus. Questo flusso di log deve includere eventi log con una definizione del formato del parametro incorporata come la seguente:

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
```

```
                "Name":"apiserver_request_total"
            }
         ],
         "Dimensions":[
            [
               "ClusterName",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "ClusterName":"my-cluster-name",
   "Namespace":"default",
   "Service":"kubernetes",
   "Timestamp":"1592267020339",
   "Version":"0",
   "apiserver_request_count":0,
   "apiserver_request_total":0,
   "code":"0",
   "component":"apiserver",
   "contentType":"application/json",
   "instance":"192.0.2.0:443",
   "job":"kubernetes-apiservers",
   "prom_metric_type":"counter",
   "resource":"pods",
   "scope":"namespace",
   "verb":"WATCH",
   "version":"v1"
}
```

Puoi visualizzare le tue metriche nella CloudWatch console nel namespace ContainerInsights/Prometheus. Facoltativamente, puoi anche creare una CloudWatch dashboard per le metriche del server API Prometheus Kubernetes.

(Facoltativo) Creazione di un pannello di controllo per i parametri del server API di Kubernetes

Per visualizzare le metriche del Kubernetes API Server nella dashboard, devi prima aver completato i passaggi nelle sezioni precedenti per iniziare a raccogliere queste metriche. CloudWatch

Per creare un pannello di controllo per i parametri del server API Kubernetes

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch

2. Assicurati di aver selezionato la AWS regione corretta.
3. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
4. Scegli Crea pannello di controllo). Immetti un nome per il nuovo dashboard e scegli Crea dashboard.
5. In Add to this dashboard (Aggiungi a questo pannello di controllo), scegli Cancel (Annulla).
6. Seleziona Actions (Operazioni), View/edit source (Visualizza/modifica origine).
7. Scaricare il seguente file JSON: origine del pannello di controllo API Kubernetes.
8. Apri il file JSON scaricato con un editor di testo e apporta le seguenti modifiche:
   - Sostituire tutte le stringhe `{{YOUR_CLUSTER_NAME}}` con il nome esatto del cluster. Assicurarti di non aggiungere spazi bianchi prima o dopo il testo.
   - Sostituisci tutte le stringhe `{{YOUR_AWS_REGION}}` con il nome della regione in cui vengono raccolti i parametri. Ad esempio, `us-west-2`. Assicurarti di non aggiungere spazi bianchi prima o dopo il testo.
9. Copia l'intero blob JSON e incollalo nella casella di testo della CloudWatch console, sostituendo ciò che è già contenuto nella casella.
10. Scegli Update (Aggiorna), Save dashboard (Salva pannello di controllo).

(Facoltativo) Impostazione dei carichi di lavoro Amazon EKS containerizzati di esempio per i test dei parametri di Prometheus

Per testare il supporto delle metriche Prometheus CloudWatch in Container Insights, puoi configurare uno o più dei seguenti carichi di lavoro containerizzati. L' CloudWatch agente con supporto Prometheus raccoglie automaticamente le metriche da ciascuno di questi carichi di lavoro. Per visualizzare le metriche raccolte per impostazione predefinita, vedere Metriche di Prometheus raccolte dall'agente CloudWatch .

Prima di installare uno qualsiasi di questi carichi di lavoro, è necessario installare Helm 3.x immettendo i seguenti comandi:

```
brew install helm
```

Per ulteriori informazioni, consulta la pagina Helm.

Argomenti

- Configurazione del carico di lavoro di esempio AWS App Mesh per Amazon EKS e Kubernetes

- Configurazione di NGINX con traffico di esempio su Amazon EKS e Kubernetes
- Impostazione di memcached con un esportatore di parametri su Amazon EKS e Kubernetes
- Configurazione del carico di lavoro di esempio Java/JMX su Amazon EKS e Kubernetes
- Impostazione di HAProxy con un esportatore di parametri su Amazon EKS e Kubernetes
- Esercitazione per l'aggiunta di una nuova destinazione di scraping di Prometheus: Redis su Amazon EKS e cluster Kubernetes

Configurazione del carico di lavoro di esempio AWS App Mesh per Amazon EKS e Kubernetes

Il supporto di Prometheus nei supporti di Container Insights. CloudWatch AWS App Mesh Nelle sezioni seguenti viene illustrato come configurare App Mesh.

CloudWatch Container Insights può anche raccogliere i log di accesso di App Mesh Envoy. Per ulteriori informazioni, consulta la pagina (Facoltativo) Abilitazione di log di accesso di App Mesh Envoy.

Argomenti

- Configurazione di un carico di lavoro di esempio AWS App Mesh su un cluster Amazon EKS con il tipo di avvio EC2 o un cluster Kubernetes
- Configura un carico di lavoro di AWS App Mesh esempio su un cluster Amazon EKS con il tipo di lancio Fargate

Configurazione di un carico di lavoro di esempio AWS App Mesh su un cluster Amazon EKS con il tipo di avvio EC2 o un cluster Kubernetes

Segui queste istruzioni se stai configurando App Mesh su un cluster che esegue Amazon EKS con il tipo di avvio EC2 o un cluster Kubernetes.

Configurazione delle autorizzazioni IAM

Devi aggiungere la AWSAppMeshFullAccesspolicy al ruolo IAM per il tuo gruppo di nodi Amazon EKS o Kubernetes. Su Amazon EKS, il nome del gruppo di nodi è simile a `eksctl-integ-test-eks-prometheus-NodeInstanceRole-ABCDEFHIJKL`. Su Kubernetes, potrebbe sembrare simile a `nodes.integ-test-kops-prometheus.k8s.local`.

Installazione di App Mesh

Per installare il controller App Mesh Kubernetes, segui le istruzioni in Controller di App Mesh.

Installazione di un'applicazione di esempio

aws-app-mesh-examplescontiene diverse procedure dettagliate per Kubernetes App Mesh. Per questa esercitazione, installerai un'applicazione a colori di esempio che mostra come i routing http possono utilizzare le intestazioni per la corrispondenza delle richieste in arrivo.

Per utilizzare un'applicazione App Mesh di esempio per testare Container Insights

1. Installa l'applicazione seguendo queste istruzioni: https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-http-headers.
2. Avvia un pod curler per generare traffico:

   ```
   kubectl -n default run -it curler --image=tutum/curl /bin/bash
   ```

3. Esegui il curl diversi endpoint modificando le intestazioni HTTP. Esegui il comando curl più volte, come illustrato:

   ```
   curl -H "color_header: blue" front.howto-k8s-http-headers.svc.cluster.local:8080/;
    echo;

   curl -H "color_header: red" front.howto-k8s-http-headers.svc.cluster.local:8080/;
    echo;

   curl -H "color_header: yellow" front.howto-k8s-http-
   headers.svc.cluster.local:8080/; echo;
   ```

4. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch
5. Nella AWS regione in cui è in esecuzione il cluster, scegli Metriche nel riquadro di navigazione. La metrica si trova nello spazio dei nomi ContainerInsights/Prometheus.
6. Per visualizzare gli eventi di CloudWatch Logs, scegli Log groups nel riquadro di navigazione. Gli eventi si trovano nel gruppo di log `/aws/containerinsights/`*`your_cluster_name`*`/prometheus` nel flusso di log `kubernetes-pod-appmesh-envoy`.

Eliminazione dell'ambiente di test di App Mesh

Al termine dell'utilizzo di App Mesh e dell'applicazione di esempio, utilizza i seguenti comandi per eliminare le risorse non necessarie. Elimina l'applicazione di esempio immettendo il seguente comando:

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-http-headers/
```

```
kubectl delete -f _output/manifest.yaml
```

Elimina il controller App Mesh immettendo il seguente comando:

```
helm delete appmesh-controller -n appmesh-system
```

Configura un carico di lavoro di AWS App Mesh esempio su un cluster Amazon EKS con il tipo di lancio Fargate

Segui queste istruzioni se stai configurando App Mesh su un cluster che esegue Amazon EKS con il tipo di avvio Fargate.

Configurazione delle autorizzazioni IAM

Per impostare le autorizzazioni IAM, immetti il comando seguente. *MyCluster*Sostituiscilo con il nome del cluster.

```
eksctl create iamserviceaccount --cluster MyCluster \
 --namespace howto-k8s-fargate \
 --name appmesh-pod \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
 --override-existing-serviceaccounts \
 --approve
```

Installazione di App Mesh

Per installare il controller App Mesh Kubernetes, segui le istruzioni in Controller di App Mesh. Assicurati di seguire le istruzioni per Amazon EKS con il tipo di avvio Fargate.

Installazione di un'applicazione di esempio

aws-app-mesh-examplescontiene diverse procedure dettagliate per Kubernetes App Mesh. Per questa esercitazione, installerai un'applicazione a colori di esempio che funziona per i cluster Amazon EKS con il tipo di avvio Fargate.

Per utilizzare un'applicazione App Mesh di esempio per testare Container Insights

1. Installa l'applicazione seguendo queste istruzioni: https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-fargate.

   Queste istruzioni presuppongono che tu stia creando un nuovo cluster con il profilo Fargate corretto. Se si desidera utilizzare un cluster Amazon EKS già configurato, puoi utilizzare i seguenti comandi per configurare tale cluster per questa dimostrazione. Sostituiscilo *MyCluster*con il nome del cluster.

   ```
   eksctl create iamserviceaccount --cluster MyCluster \
    --namespace howto-k8s-fargate \
    --name appmesh-pod \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
    --override-existing-serviceaccounts \
    --approve
   ```

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --namespace howto-k8s-fargate --name howto-k8s-fargate
   ```

2. Esegui il port forwarding dell'implementazione dell'applicazione front:

   ```
   kubectl -n howto-k8s-fargate port-forward deployment/front 8080:8080
   ```

3. Esegui il curl dell'App front:

   ```
   while true; do  curl -s http://localhost:8080/color; sleep 0.1; echo ; done
   ```

4. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

5. Nella AWS regione in cui è in esecuzione il cluster, scegli Metriche nel riquadro di navigazione. La metrica si trova nello spazio dei nomi ContainerInsights/Prometheus.

6. Per visualizzare gli eventi di CloudWatch Logs, scegli Log groups nel riquadro di navigazione. Gli eventi si trovano nel gruppo di log `/aws/containerinsights/`*`your_cluster_name`*`/prometheus` nel flusso di log `kubernetes-pod-appmesh-envoy`.

Eliminazione dell'ambiente di test di App Mesh

Al termine dell'utilizzo di App Mesh e dell'applicazione di esempio, utilizza i seguenti comandi per eliminare le risorse non necessarie. Elimina l'applicazione di esempio immettendo il seguente comando:

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-fargate/
kubectl delete -f _output/manifest.yaml
```

Elimina il controller App Mesh immettendo il seguente comando:

```
helm delete appmesh-controller -n appmesh-system
```

Configurazione di NGINX con traffico di esempio su Amazon EKS e Kubernetes

NGINX è un server web che può essere utilizzato anche come load balancer e proxy inverso. Per ulteriori informazioni sul modo in cui Kubernetes utilizza NGINX per ingresso, consulta kubernetes/ingress-nginx.

Per installare Ingress-NGINX con un servizio di traffico di esempio per testare il supporto di Container Insights Prometheus

1. Immetti il seguente comando per aggiungere il repository Helm ingress-nginx:

   ```
   helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
   ```

2. Eseguire i comandi seguenti:

   ```
   kubectl create namespace nginx-ingress-sample

   helm install my-nginx ingress-nginx/ingress-nginx \
   --namespace nginx-ingress-sample \
   --set controller.metrics.enabled=true \
   --set-string controller.metrics.service.annotations."prometheus\.io/port"="10254" \
   --set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
   ```

3. Verificare se i servizi sono stati avviati correttamente immettendo il seguente comando:

   ```
   kubectl get service -n nginx-ingress-sample
   ```

L'output di questo comando dovrebbe visualizzare diverse colonne, inclusa una colonna `EXTERNAL-IP`.

4. Impostare una variabile `EXTERNAL-IP` sul valore della colonna `EXTERNAL-IP` nella riga del controller di ingresso NGINX.

```
EXTERNAL_IP=your-nginx-controller-external-ip
```

5. Avviare un esempio di traffico NGINX inserendo il seguente comando.

```
SAMPLE_TRAFFIC_NAMESPACE=nginx-sample-traffic
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_traffic/nginx-traffic/nginx-traffic-sample.yaml |
sed "s/{{external_ip}}/$EXTERNAL_IP/g" |
sed "s/{{namespace}}/$SAMPLE_TRAFFIC_NAMESPACE/g" |
kubectl apply -f -
```

6. Immetti il comando seguente per verificare che tutti e tre i pod siano nello stato `Running`.

```
kubectl get pod -n $SAMPLE_TRAFFIC_NAMESPACE
```

Se sono in esecuzione, dovresti presto vedere le metriche nello spazio dei nomi ContainerInsights/Prometheus.

Per disinstallare NGINX e l'applicazione di traffico di esempio

1. Eliminare il servizio traffico di esempio immettendo il seguente comando:

```
kubectl delete namespace $SAMPLE_TRAFFIC_NAMESPACE
```

2. Elimina l'uscita NGINX in base al nome della versione di Helm.

```
helm uninstall my-nginx --namespace nginx-ingress-sample
kubectl delete namespace nginx-ingress-sample
```

Impostazione di memcached con un esportatore di parametri su Amazon EKS e Kubernetes

memcached è un sistema di memorizzazione nella cache di oggetti di memoria open source. Per ulteriori informazioni, consulta la pagina Cos'è Memcached?.

Se esegui memcached in un cluster con il tipo di avvio Fargate, devi impostare un profilo Fargate prima di eseguire la procedura descritta. Per impostare il profilo, inserisci il comando seguente. Sostituiscilo con il nome *MyCluster*del tuo cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace memcached-sample --name memcached-sample
```

Per installare memcached con un esportatore di metriche per testare il supporto di Container Insights Prometheus

1. Inserisci il seguente comando per aggiungere il repository:

   ```
   helm repo add bitnami https://charts.bitnami.com/bitnami
   ```

2. Immetti il seguente comando per creare un nuovo spazio di nomi:

   ```
   kubectl create namespace memcached-sample
   ```

3. Immetti il seguente comando per installare Memcached

   ```
   helm install my-memcached bitnami/memcached --namespace memcached-sample \
   --set metrics.enabled=true \
   --set-string serviceAnnotations.prometheus\\.io/port="9150" \
   --set-string serviceAnnotations.prometheus\\.io/scrape="true"
   ```

4. Immetti il comando seguente per confermare l'annotazione del servizio in esecuzione:

   ```
   kubectl describe service my-memcached-metrics -n memcached-sample
   ```

   Dovresti vedere le due annotazioni seguenti:

   ```
   Annotations:   prometheus.io/port: 9150
                  prometheus.io/scrape: true
   ```

Per disinstallare memcached

- Eseguire i comandi seguenti:

```
helm uninstall my-memcached --namespace memcached-sample
kubectl delete namespace memcached-sample
```

Configurazione del carico di lavoro di esempio Java/JMX su Amazon EKS e Kubernetes

JMX Exporter è un esportatore ufficiale di Prometheus che può recuperare ed esporre JMX mBeans JMX come metriche Prometheus. Per ulteriori informazioni, vedere prometheus/jmx_exporter.

Container Insights può raccogliere metriche Prometheus predefinite da Java Virtual Machine (JVM), Java e Tomcat (Catalina) utilizzando JMX Exporter.

Configurazione predefinita dello scraping di Prometheus

Per impostazione predefinita, l' CloudWatch agente con supporto Prometheus analizza i parametri Java/JMX Prometheus da ogni pod in un cluster Amazon EKS o Kubernetes. `http://CLUSTER_IP:9404/metrics` Questo avviene mediante l'individuazione di `role: pod` di `kubernetes_sd_config` Prometheus. 9404 è la porta predefinita allocata per JMX Exporter da Prometheus. Per ulteriori informazioni sull'individuazione di `role: pod`, consulta pod. È possibile configurare JMX Exporter per esporre le metriche su una porta o metrics_path diverso. Se modifichi la porta o il percorso, aggiorna il jmx scrape_config predefinito nella mappa di configurazione dell'agente. CloudWatch Esegui il seguente comando per ottenere la configurazione corrente dell' CloudWatch agente Prometheus:

```
kubectl describe cm prometheus-config -n amazon-cloudwatch
```

I campi da modificare sono i campi `/metrics` e `regex: '.*:9404$'`, come evidenziato nell'esempio seguente.

```
job_name: 'kubernetes-jmx-pod'
sample_limit: 10000
metrics_path: /metrics
kubernetes_sd_configs:
- role: pod
relabel_configs:
- source_labels: [__address__]
```

```
  action: keep
  regex: '.*:9404$'
- action: replace
  regex: (.+)
  source_labels:
```

Altra configurazione di scraping di Prometheus

Se esponi la tua applicazione in esecuzione su un set di pod con esportatori Java/JMX Prometheus da un servizio Kubernetes, puoi anche passare all'uso dell'individuazione di `role: service` o all'individuazione di `role: endpoint` di `kubernetes_sd_config` Prometheus. Per ulteriori informazioni su questi metodi di individuazione, consulta service (servizio), endpoints (endpoint) e <kubernetes_sd_config>..

Queste due modalità di rilevamento dei servizi forniscono altre meta etichette che potrebbero essere utili per creare le dimensioni delle CloudWatch metriche. Ad esempio, puoi rietichettare `__meta_kubernetes_service_name` con `Service` e includerlo nella dimensione dei tuoi parametri. Per ulteriori informazioni sulla personalizzazione delle CloudWatch metriche e delle relative dimensioni, consulta. CloudWatch configurazione dell'agente per Prometheus

Immagine Docker con JMX Exporter

Successivamente, crea un'immagine Docker. Le sezioni seguenti forniscono due file Docker di esempio.

Dopo aver creato l'immagine, caricarla in Amazon EKS o Kubernetes, quindi eseguire il comando seguente per verificare che i parametri Prometheus siano esposte da `JMX_EXPORTER` sulla porta 9404. Sostituire *`$JAR_SAMPLE_TRAFFIC_POD`* con il nome del pod in esecuzione e sostituire *`$JAR_SAMPLE_TRAFFIC_NAMESPACE`* con lo spazio dei nomi dell'applicazione.

Se esegui JMX Exporter in un cluster con il tipo di avvio Fargate, devi anche impostare un profilo Fargate prima di eseguire la procedura descritta. Per impostare il profilo, inserisci il comando seguente. *`MyCluster`*Sostituiscilo con il nome del cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace $JAR_SAMPLE_TRAFFIC_NAMESPACE\
 --name $JAR_SAMPLE_TRAFFIC_NAMESPACE
```

```
kubectl exec $JAR_SAMPLE_TRAFFIC_POD -n $JARCAT_SAMPLE_TRAFFIC_NAMESPACE -- curl
 http://localhost:9404
```

Esempio: immagine Docker Apache Tomcat con parametri Prometheus

Il server Apache Tomcat espone JMX mBeans per impostazione predefinita. È possibile integrare JMX Exporter con Tomcat per esporre JMX mBeans come metriche Prometheus. L'esempio seguente di file Docker mostra i passaggi per creare un'immagine di prova:

```
# From Tomcat 9.0 JDK8 OpenJDK
FROM tomcat:9.0-jdk8-openjdk

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter
COPY ./setenv.sh /usr/local/tomcat/bin
COPY your web application.war /usr/local/tomcat/webapps/

RUN chmod  o+x /usr/local/tomcat/bin/setenv.sh

ENTRYPOINT ["catalina.sh", "run"]
```

Nell'elenco seguente vengono illustrate le quattro righe `COPY` di questo file Docker.

- Scaricare l'ultimo file jar JMX Exporter da https://github.com/prometheus/jmx_exporter.
- `config.yaml` è il file di configurazione JMX Exporter. Per ulteriori informazioni, visitare il sito Web all'indirizzo https://github.com/prometheus/jmx_exporter#Configuration.

  Ecco un file di configurazione di esempio per Java e Tomcat:

  ```
  lowercaseOutputName: true
  lowercaseOutputLabelNames: true

  rules:
  - pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
  TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
  ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
    name: java_lang_OperatingSystem_$1
    type: GAUGE

  - pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
    name: java_lang_threading_$1
    type: GAUGE
  ```

```
- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `setenv.sh` è uno script di startup Tomcat per avviare JMX exporter insieme a Tomcat ed esporre le metriche di Prometheus sulla porta 9404 del localhost. Fornisce inoltre al JMX Exporter il percorso del file `config.yaml`.

```
$ cat setenv.sh
export JAVA_OPTS="-javaagent:/opt/jmx_exporter/
jmx_prometheus_javaagent-0.12.0.jar=9404:/opt/jmx_exporter/config.yaml $JAVA_OPTS"
```

- il tuo web application .war è il tuo file `war` dell'applicazione web che Tomcat deve caricare.

Crea un'immagine docker con questa configurazione e caricala in un repository di immagini.

Esempio: immagine Docker dell'applicazione Java Jar con parametri Prometheus

L'esempio seguente di file Docker mostra i passaggi per creare un'immagine di prova:

```
# Alpine Linux with OpenJDK JRE
FROM openjdk:8-jre-alpine

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./SampleJavaApplication-1.0-SNAPSHOT.jar /opt/jmx_exporter
COPY ./start_exporter_example.sh /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter

RUN chmod -R o+x /opt/jmx_exporter
RUN apk add curl

ENTRYPOINT exec /opt/jmx_exporter/start_exporter_example.sh
```

Nell'elenco seguente vengono illustrate le quattro righe `COPY` di questo file Docker.

- Scaricare l'ultimo file jar JMX Exporter da https://github.com/prometheus/jmx_exporter.
- `config.yaml` è il file di configurazione JMX Exporter. Per ulteriori informazioni, visitare il sito Web all'indirizzo https://github.com/prometheus/jmx_exporter#Configuration.

  Ecco un file di configurazione di esempio per Java e Tomcat:

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
```

```
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
```

```
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `start_exporter_example.sh` è lo script per avviare l'applicazione JAR con le metriche Prometheus esportate. Fornisce inoltre al JMX Exporter il percorso del file `config.yaml`.

```
$ cat start_exporter_example.sh
java -javaagent:/opt/jmx_exporter/jmx_prometheus_javaagent-0.12.0.jar=9404:/
opt/jmx_exporter/config.yaml -cp  /opt/jmx_exporter/SampleJavaApplication-1.0-
SNAPSHOT.jar com.gubupt.sample.app.App
```

- SampleJavaApplication-1.0-Snapshot.jar è il file jar dell'applicazione Java di esempio. Sostituirlo con l'applicazione Java che si desidera monitorare.

Crea un'immagine docker con questa configurazione e caricala in un repository di immagini.

Impostazione di HAProxy con un esportatore di parametri su Amazon EKS e Kubernetes

HAProxy è un'applicazione proxy open-source. Per ulteriori informazioni, consulta la pagina HAProxy.

Se esegui HAProxy in un cluster con il tipo di avvio Fargate, devi impostare un profilo Fargate prima di eseguire la procedura descritta. Per impostare il profilo, inserisci il comando seguente. Sostituiscilo con il nome *`MyCluster`*del cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace haproxy-ingress-sample --name haproxy-ingress-sample
```

Per installare HAProxy con un esportatore di metriche per testare il supporto di Container Insights Prometheus

1. Immetti il seguente comando per aggiungere il repository dell'incubatore di Helm:

   ```
   helm repo add haproxy-ingress https://haproxy-ingress.github.io/charts
   ```

2. Immetti il seguente comando per creare un nuovo spazio di nomi:

   ```
   kubectl create namespace haproxy-ingress-sample
   ```

3. Immetti i seguenti comandi per installare HAProxy:

```
helm install haproxy haproxy-ingress/haproxy-ingress \
--namespace haproxy-ingress-sample \
--set defaultBackend.enabled=true \
--set controller.stats.enabled=true \
--set controller.metrics.enabled=true \
--set-string controller.metrics.service.annotations."prometheus\.io/port"="9101" \
--set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
```

4. Immetti il seguente comando per confermare l'annotazione del servizio:

```
kubectl describe service haproxy-haproxy-ingress-metrics -n haproxy-ingress-sample
```

Dovresti vedere le annotazioni seguenti.

```
Annotations:   prometheus.io/port: 9101
               prometheus.io/scrape: true
```

Per disinstallare HAProxy

- Eseguire i comandi seguenti:

```
helm uninstall haproxy --namespace haproxy-ingress-sample
kubectl delete namespace haproxy-ingress-sample
```

Esercitazione per l'aggiunta di una nuova destinazione di scraping di Prometheus: Redis su Amazon EKS e cluster Kubernetes

Questa esercitazione offre un'introduzione pratica per recuperare i parametri Prometheus di un'applicazione Redis di esempio su Amazon EKS e Kubernetes. Redis (https://redis.io/) è uno store di strutture dati open source (con licenza BSD), in memoria, utilizzato come database, cache e broker di messaggi. Per ulteriori informazioni, consulta la pagina redis.

redis_exporter (con licenza MIT Licence) viene utilizzato per esporre i parametri Prometheus Redis sulla porta specificata (predefinita: 0.0.0.0:9121). Per ulteriori informazioni, consulta la pagina redis_exporter.

In questa esercitazione vengono utilizzate le immagini Docker nei due repository Docker Hub seguenti:

- redis
- redis_exporter

Per installare un carico di lavoro Redis di esempio che espone i parametri Prometheus

1. Imposta lo spazio dei nomi per il carico di lavoro Redis di esempio.

```
REDIS_NAMESPACE=redis-sample
```

2. Se esegui Redis in un cluster con il tipo di avvio Fargate, devi impostare un profilo Fargate. Per impostare il profilo, inserisci il comando seguente. Sostituiscilo *MyCluster*con il nome del tuo cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace $REDIS_NAMESPACE --name $REDIS_NAMESPACE
```

3. Inserisci il comando seguente per installare il carico di lavoro Redis di esempio.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_traffic/redis/redis-traffic-sample.yaml \
| sed "s/{{namespace}}/$REDIS_NAMESPACE/g" \
| kubectl apply -f -
```

4. L'installazione include un servizio denominato `my-redis-metrics` che espone il parametro Prometheus di Redis sulla porta 9121 Inserisci il seguente comando per ottenere informazioni sul servizio:

```
kubectl describe service/my-redis-metrics  -n $REDIS_NAMESPACE
```

   Nella `Annotations` sezione dei risultati, vedrai due annotazioni che corrispondono alla configurazione dello scrape Prometheus dell'agente, in modo che possa rilevare automaticamente CloudWatch i carichi di lavoro:

```
prometheus.io/port: 9121
prometheus.io/scrape: true
```

   La configurazione di scraping di Prometheus correlata è disponibile nella sezione `- job_name: kubernetes-service-endpoints` di `kubernetes-eks.yaml` o `kubernetes-k8s.yaml`.

Per iniziare a raccogliere le metriche di Redis Prometheus in CloudWatch

1. Scaricare la versione più recente del file `kubernetes-eks.yaml` o `kubernetes-k8s.yaml` immettendo uno dei seguenti comandi. Per un cluster Amazon EKS con tipo di avvio EC2, inserisci questo comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Per un cluster Amazon EKS con tipo di avvio Fargate, inserisci questo comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks-fargate.yaml
   ```

   Per un cluster Kubernetes in esecuzione su un'istanza Amazon EC2, inserisci questo comando.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-k8s.yaml
   ```

2. Apri il file con un editor di testo e trova la sezione `cwagentconfig.json`. Aggiungi la seguente sottosezione e salva le modifiche. Assicurati che il rientro segua il modello esistente.

   ```
   {
     "source_labels": ["pod_name"],
     "label_matcher": "^redis-instance$",
     "dimensions": [["Namespace","ClusterName"]],
     "metric_selectors": [
       "^redis_net_(in|out)put_bytes_total$",
       "^redis_(expired|evicted)_keys_total$",
       "^redis_keyspace_(hits|misses)_total$",
       "^redis_memory_used_bytes$",
       "^redis_connected_clients$"
     ]
   },
   {
     "source_labels": ["pod_name"],
     "label_matcher": "^redis-instance$",
     "dimensions": [["Namespace","ClusterName","cmd"]],
   ```

```
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName","db"]],
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

La sezione che hai aggiunto inserisce le metriche Redis nell'elenco degli agenti consentiti. CloudWatch Per l'elenco di questi parametri, consulta la sezione seguente.

3. Se l' CloudWatch agente con supporto Prometheus è già distribuito in questo cluster, è necessario eliminarlo immettendo il seguente comando.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

4. Distribuisci l' CloudWatch agente con la configurazione aggiornata immettendo uno dei seguenti comandi. Sostituisci *MyCluster*una *regione* in modo che corrisponda alle tue impostazioni.

   Per un cluster Amazon EKS con tipo di avvio EC2, inserisci questo comando.

```
kubectl apply -f prometheus-eks.yaml
```

   Per un cluster Amazon EKS con tipo di avvio Fargate, inserisci questo comando.

```
cat prometheus-eks-fargate.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

   Per un cluster Kubernetes, inserisci il seguente comando.

```
cat prometheus-k8s.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Visualizzazione dei parametri Prometheus di Redis

Questo tutorial invia le seguenti metriche allo spazio dei nomi ContainerInsights/Prometheus in. CloudWatch Puoi usare la CloudWatch console per visualizzare le metriche in quel namespace.

Nome parametro	Dimensioni	
`redis_net_input_bytes_total`	ClusterName, `Namespace`	
`redis_net_output_bytes_total`	ClusterName, `Namespace`	
`redis_expired_keys_total`	ClusterName, `Namespace`	
`redis_evicted_keys_total`	ClusterName, `Namespace`	
`redis_keyspace_hits_total`	ClusterName, `Namespace`	
`redis_keyspace_misses_total`	ClusterName, `Namespace`	
`redis_memory_used_bytes`	ClusterName, `Namespace`	
`redis_connected_clients`	ClusterName, `Namespace`	
`redis_commands_total`	ClusterName,, cmd `Namespace`	

Nome parametro	Dimensioni	
`redis_db_keys`	ClusterName, `Namespace db`	

> **Note**
>
> Il valore della dimensione cmd può essere `append`, `client`, `command`, `config`, `dbsize`, `flushall`, `get`, `incr`, `info`, `latency` o `slowlog`.
> Il valore della dimensione db può essere da `db0` a `db15`.

Puoi anche creare una CloudWatch dashboard per le metriche di Redis Prometheus.

Per creare un pannello di controllo per i parametri Prometheus di Redis

1. Crea variabili di ambiente, sostituendo i valori sotto in modo che corrispondano all'implementazione.

```
DASHBOARD_NAME=your_cw_dashboard_name
REGION_NAME=your_metric_region_such_as_us-east-1
CLUSTER_NAME=your_k8s_cluster_name_here
NAMESPACE=your_redis_service_namespace_here
```

2. Inserisci il seguente comando per creare il pannello di controllo.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Conversione del tipo di metrica Prometheus da parte dell'agente CloudWatch

Le librerie client Prometheus offrono quattro tipi di parametri principali:

- Contatore
- Misuratore

- Riepilogo
- Istogramma

L' CloudWatch agente supporta i tipi di metriche counter, gauge e summary. Il supporto per i parametri dell'istogramma e è pianificato per una release imminente.

Le metriche Prometheus con il tipo di metrica istogramma non supportato vengono eliminate dall'agente. CloudWatch Per ulteriori informazioni, consulta la pagina Registrazione dei parametri Prometheus eliminati.

Parametri misuratore

Un parametro misuratore Prometheus è un parametro che rappresenta un singolo valore numerico che può aumentare o diminuire arbitrariamente. L' CloudWatch agente analizza le metriche degli indicatori e invia direttamente questi valori.

Metriche dei contatori

Un parametro contatore Prometheus è un parametro cumulativo che rappresenta un singolo contatore che cresce monotonicamente, il cui valore può solo aumentare o essere reimpostato a zero. L' CloudWatchagente calcola un delta dallo scrape precedente e invia il valore delta come valore metrico nell'evento di registro. Quindi l' CloudWatch agente inizierà a produrre un evento di registro dal secondo scrape e continuerà con gli scrape successivi, se presenti.

Parametri di riepilogo

Un parametro di riepilogo Prometheus è un tipo di parametro complesso rappresentato da più punti dati. Offre un conteggio totale delle osservazioni e una somma di tutti i valori osservati. Calcola quantili configurabili su una finestra temporale scorrevole.

La somma e il conteggio di un parametro di riepilogo sono cumulativi, ma i quantili non lo sono. L'esempio seguente mostra la varianza dei quantili.

```
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 7.123e-06
go_gc_duration_seconds{quantile="0.25"} 9.204e-06
go_gc_duration_seconds{quantile="0.5"} 1.1065e-05
go_gc_duration_seconds{quantile="0.75"} 2.8731e-05
go_gc_duration_seconds{quantile="1"} 0.003841496
go_gc_duration_seconds_sum 0.37630427
go_gc_duration_seconds_count 9774
```

L' CloudWatch agente gestisce la somma e il conteggio di una metrica di riepilogo nello stesso modo in cui gestisce le metriche dei contatori, come descritto nella sezione precedente. L' CloudWatch agente conserva i valori quantili così come sono stati originariamente riportati.

Metriche di Prometheus raccolte dall'agente CloudWatch

L' CloudWatch agente con supporto Prometheus raccoglie automaticamente le metriche da diversi servizi e carichi di lavoro. Nelle sezioni seguenti sono elencati i parametri raccolti per impostazione predefinita. È inoltre possibile configurare l'agente per raccogliere altri parametri da questi servizi e per raccogliere parametri Prometheus da altri applicazioni e servizi. Per ulteriori informazioni sulla raccolta di parametri aggiuntivi, consulta CloudWatch configurazione dell'agente per Prometheus.

Le metriche di Prometheus raccolte dai cluster Amazon EKS e Kubernetes si trovano nello spazio dei nomi /Prometheus. ContainerInsights Le metriche di Prometheus raccolte dai cluster Amazon ECS si trovano nello spazio dei nomi ECS/ /Prometheus. ContainerInsights

Argomenti

- Parametri Prometheus per App Mesh
- Parametri Prometheus per NGINX
- Parametri Prometheus per Memcached
- Parametri Prometheus per Java/JMX
- Parametri Prometheus per HAProxy

Parametri Prometheus per App Mesh

I seguenti parametri vengono raccolti automaticamente da App Mesh.

CloudWatch Container Insights può anche raccogliere i log di accesso di App Mesh Envoy. Per ulteriori informazioni, consulta la pagina (Facoltativo) Abilitazione di log di accesso di App Mesh Envoy.

Parametri Prometheus per App Mesh su cluster Amazon EKS e Kubernetes

Nome parametro	Dimensioni	
`envoy_http_downstream_rq_total`	ClusterName, `Namespace`	

Nome parametro	Dimensioni	
envoy_http_downstream_rq_xx	ClusterName, Namespace ClusterName, envoy_http_conn_manager_prefix, Namespace envoy_response_code_class	
envoy_cluster_upstream_cx_rx_bytes_total	ClusterName, Namespace	
envoy_cluster_upstream_cx_tx_bytes_total	ClusterName, Namespace	
envoy_cluster_membership_healthy	ClusterName, Namespace	
envoy_cluster_membership_total	ClusterName, Namespace	
envoy_server_memory_heap_size	ClusterName, Namespace	
envoy_server_memory_allocated	ClusterName, Namespace	
envoy_cluster_upstream_cx_connect_timeout	ClusterName, Namespace	

Nome parametro	Dimensioni	
envoy_cluster_upstream_rq_pending_failure_eject	ClusterName, Namespace	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, Namespace	
envoy_cluster_upstream_rq_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_rx_reset	ClusterName, Namespace	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, Namespace	

Nome parametro	Dimensioni	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, Namespace	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, Namespace	

Nome parametro	Dimensioni	
envoy_cluster_upstream_flow_control_drained_total	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_success	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_overflow	ClusterName, Namespace	
envoy_server_live	ClusterName, Namespace	
envoy_server_uptime	ClusterName, Namespace	

Parametri Prometheus per App Mesh su cluster Amazon ECS

Nome parametro	Dimensioni	
envoy_http_downstream_rq_total	ClusterName, TaskDefinitionFamily	

Nome parametro	Dimensioni	
envoy_http_downstream_rq_xx	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_rx_bytes_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_tx_bytes_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_membership_healthy	ClusterName, TaskDefinitionFamily	
envoy_cluster_membership_total	ClusterName, TaskDefinitionFamily	
envoy_server_memory_heap_size	ClusterName, TaskDefinitionFamily	
envoy_server_memory_allocated	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_connect_timeout	ClusterName, TaskDefinitionFamily	

Nome parametro	Dimensioni	
envoy_cluster_upstream_rq_pending_failure_eject	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_rx_reset	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, TaskDefinitionFamily	

Nome parametro	Dimensioni	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, TaskDefinitionFamily	

Nome parametro	Dimensioni
envoy_cluster_upstream_flow_control_drained_total	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry_success	ClusterName, TaskDefinitionFamily
envoy_cluster_upstream_rq_retry_overflow	ClusterName, TaskDefinitionFamily
envoy_server_live	ClusterName, TaskDefinitionFamily
envoy_server_uptime	ClusterName, TaskDefinitionFamily
envoy_http_downstream_rq_xx	ClusterName TaskDefinitionFamily, envoy_http_conn_manager_prefix, envoy_response_code_class ClusterName TaskDefinitionFamily, envoy_response_code_class

Note

TaskDefinitionFamily è lo spazio dei nomi Kubernetes della mesh.

Il valore di `envoy_http_conn_manager_prefix` può essere `ingress`, `egress` o `admin`. Il valore di `envoy_response_code_class` può essere 1 (sta per `1xx`), 2 (sta per `2xx`), 3 (sta per `3xx`), 4 (sta per `4xx`) o 5 (sta per `5xx`).

Parametri Prometheus per NGINX

I seguenti parametri vengono raccolti automaticamente da NGINX su cluster Amazon EKS e Kubernetes.

Nome parametro	Dimensioni	
`nginx_ingress_controller_nginx_process_cpu_seconds_total`	ClusterNameNamespace ,, Servizio	
`nginx_ingress_controller_success`	ClusterName,`Namespace` , Servizio	
`nginx_ingress_controller_requests`	ClusterName,`Namespace` , Servizio	
`nginx_ingress_controller_nginx_process_connections`	ClusterName,`Namespace` , Servizio	
`nginx_ingress_controller_nginx_proce`	ClusterName,`Namespace` , Servizio	

Nome parametro	Dimensioni	
ss_connections_total		
nginx_ingress_controller_nginx_process_resident_memory_bytes	ClusterName,Namespace , Servizio	
nginx_ingress_controller_config_last_reload_successful	ClusterName,Namespace , Servizio	
nginx_ingress_controller_requests	ClusterName,Namespace , Servizio, stato	

Parametri Prometheus per Memcached

I seguenti parametri vengono raccolti automaticamente da Memcached su cluster Amazon EKS e Kubernetes.

Nome parametro	Dimensioni	
memcached_current_items	ClusterName,Namespace , Servizio	

Nome parametro	Dimensioni
memcached_current_connections	ClusterName,Namespace , Servizio
memcached_limit_bytes	ClusterName,Namespace , Servizio
memcached_current_bytes	ClusterName,Namespace , Servizio
memcached_written_bytes_total	ClusterName,Namespace , Servizio
memcached_read_bytes_total	ClusterName,Namespace , Servizio
memcached_items_evicted_total	ClusterName,Namespace , Servizio
memcached_items_reclaimed_total	ClusterName,Namespace , Servizio
memcached_commands_total	ClusterName,Namespace , Servizio ClusterName,Namespace , Servizio, comando ClusterName,Namespace , Servizio, stato, comando

Parametri Prometheus per Java/JMX

Parametri raccolti sui cluster Amazon EKS e Kubernetes

Sui cluster Amazon EKS e Kubernetes, Container Insights può raccogliere le seguenti parametri Prometheus predefinite da Java Virtual Machine (JVM), Java e Tomcat (Catalina) utilizzando JMX Exporter. Per ulteriori informazioni, vedere prometheus/jmx_exporter su Github.

Java/JMX su cluster Amazon EKS e Kubernetes

Nome parametro	Dimensioni	
jvm_classes_loaded	ClusterName , Namespace	
jvm_threads_current	ClusterName , Namespace	
jvm_threads_daemon	ClusterName , Namespace	
java_lang_operatingsystem_totalswapspacesize	ClusterName , Namespace	
java_lang_operatingsystem_systemcpuload	ClusterName , Namespace	
java_lang_operatingsystem_processcpuload	ClusterName , Namespace	
java_lang_operatingsystem_freeswapspacesize	ClusterName , Namespace	

Nome parametro	Dimensioni
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , Namespace
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , Namespace
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , Namespace
java_lang_operatingsystem_availableprocessors	ClusterName , Namespace
jvm_memory_bytes_used	ClusterName , Namespace , area
jvm_memory_pool_bytes_used	ClusterName , Namespace , pool

> **Note**
>
> I valori della dimensione `area` possono essere heap o nonheap.

I valori della dimensione `pool` possono essere `Tenured Gen`, `Compress Class Space`, `Survivor Space`, `Eden Space`, `Code Cache` o `Metaspace`.

Tomcat/JMX su cluster Amazon EKS e Kubernetes

Oltre alle metriche Java/JMX nella tabella precedente, vengono raccolte anche le seguenti metriche per il carico di lavoro Tomcat.

Nome parametro	Dimensioni	
`catalina_manager_activesessions`	`ClusterName` , `Namespace`	
`catalina_manager_rejectedsessions`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_bytesreceived`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_bytessent`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_requestcount`	`ClusterName` , `Namespace`	

Nome parametro	Dimensioni	
catalina_globalrequestprocessor_errorcount	ClusterName , Namespace	
catalina_globalrequestprocessor_processingtime	ClusterName , Namespace	

Java/JMX su cluster Amazon ECS

Nome parametro	Dimensioni	
jvm_classes_loaded	ClusterName , TaskDefinitionFamily	
jvm_threads_current	ClusterName , TaskDefinitionFamily	
jvm_threads_daemon	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_systemcpuload	ClusterName , TaskDefinitionFamily	

Nome parametro	Dimensioni	
java_lang_operatingsystem_processcpuload	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_freeswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_availableprocessors	ClusterName , TaskDefinitionFamily	

Nome parametro	Dimensioni	
jvm_memory_bytes_used	ClusterName , TaskDefinitionFamily, area	
jvm_memory_pool_bytes_used	ClusterName TaskDefinitionFamily, piscina	

> **Note**
>
> I valori della dimensione area possono essere heap o nonheap.
> I valori della dimensione pool possono essere Tenured Gen, Compress Class Space, Survivor Space, Eden Space, Code Cache o Metaspace.

Tomcat/JMX su cluster Amazon ECS

Oltre ai parametri Java/JMX nella tabella precedente, vengono raccolti anche i seguenti parametri per il carico di lavoro Tomcat su cluster Amazon ECS.

Nome parametro	Dimensioni	
catalina_manager_activesessions	ClusterName , TaskDefinitionFamily	
catalina_manager_rejectedsessions	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_bytesreceived	ClusterName , TaskDefinitionFamily	

Nome parametro	Dimensioni	
catalina_globalrequestprocessor_bytessent	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_requestcount	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_errorcount	ClusterName , TaskDefinitionFamily	
catalina_globalrequestprocessor_processingtime	ClusterName , TaskDefinitionFamily	

Parametri Prometheus per HAProxy

I seguenti parametri vengono raccolti automaticamente da HAProxy su cluster Amazon EKS e Kubernetes.

I parametri raccolti dipendono dalla versione di HAProxy Ingress in uso. Per ulteriori informazioni su HAProxy Ingress e le sue versioni, consulta haproxy-ingress.

Nome parametro	Dimensioni	Disponibilità
haproxy_backend_bytes_in_total	ClusterName , Namespace , Service	Tutte le versioni di HAProxy Ingress
haproxy_backend_bytes_out_total	ClusterName , Namespace , Service	Tutte le versioni di HAProxy Ingress
haproxy_backend_connection_errors_total	ClusterName , Namespace , Service	Tutte le versioni di HAProxy Ingress
haproxy_backend_connections_total	ClusterName , Namespace , Service	Tutte le versioni di HAProxy Ingress
haproxy_backend_current_sessions	ClusterName , Namespace , Service	Tutte le versioni di HAProxy Ingress
haproxy_backend_http_responses_total	ClusterName , Namespace , Service, code, backend	Tutte le versioni di HAProxy Ingress
haproxy_backend_status	ClusterName , Namespace , Service	Solo nelle versioni 0.10 o successive di HAProxy Ingress
haproxy_backend_up	ClusterName , Namespace , Service	Solo nelle versioni di HAProxy Ingress precedenti alla 0.10

Nome parametro	Dimensioni	Disponibilità
`haproxy_frontend_bytes_in_total`	`ClusterName` , `Namespace` , Service	Tutte le versioni di HAProxy Ingress
`haproxy_frontend_bytes_out_total`	`ClusterName` , `Namespace` , Service	Tutte le versioni di HAProxy Ingress
`haproxy_frontend_connections_total`	`ClusterName` , `Namespace` , Service	Tutte le versioni di HAProxy Ingress
`haproxy_frontend_current_sessions`	`ClusterName` , `Namespace` , Service	Tutte le versioni di HAProxy Ingress
`haproxy_frontend_http_requests_total`	`ClusterName` , `Namespace` , Service	Tutte le versioni di HAProxy Ingress
`haproxy_frontend_http_responses_total`	`ClusterName` , `Namespace` , Service, code, frontend	Tutte le versioni di HAProxy Ingress
`haproxy_frontend_request_errors_total`	`ClusterName` , `Namespace` , Service	Tutte le versioni di HAProxy Ingress

Nome parametro	Dimensioni	Disponibilità
`haproxy_frontend_requests_denied_total`	`ClusterName` , `Namespace` , Service	Tutte le versioni di HAProxy Ingress

> **Note**
>
> I valori della dimensione `code` possono essere `1xx`, `2xx`, `3xx`, `4xx`, `5xx` o `other`.
> I valori della dimensione `backend` possono essere:
>
> - `http-default-backend`, `http-shared-backend` oppure `httpsback-shared-backend` per HAProxy Ingress versione 0.0.27 o precedente.
> - `_default_backend` per HAProxy Ingress, versioni successive alla 0.0.27.
>
> I valori della dimensione `frontend` possono essere:
>
> - `httpfront-default-backend`, `httpfront-shared-frontend` oppure `httpfronts` per HAProxy Ingress versione 0.0.27 o precedente.
> - `_front_http` o `_front_https` per HAProxy Ingress, versioni successive alla 0.0.27.

Visualizzazione dei parametri Prometheus

È possibile monitorare e impostare allarmi su tutti i parametri Prometheus inclusi i parametri preaggregati curati da App Mesh, NGINX, Java/JMX, Memcached e HAProxy e qualsiasi altro esportatore Prometheus configurato manualmente che potresti aver aggiunto. Per ulteriori informazioni sulla raccolta di metriche da altri esportatori di Prometheus, vedere Esercitazione per l'aggiunta di nuove destinazioni di scraping di Prometheus: parametri del server API Prometheus.

Nella CloudWatch console, Container Insights fornisce i seguenti report predefiniti:

- Per i cluster Amazon EKS e Kubernetes, ci sono report predefiniti per App Mesh, NGINX, HAPROXY, Memcached e Java/JMX.
- Per i cluster Amazon ECS, ci sono report predefiniti per App Mesh e Java/JMX.

Container Insights offre inoltre pannelli di controllo personalizzati per ciascuno dei carichi di lavoro da cui Container Insights raccoglie parametri curati. Puoi scaricare questi dashboard da GitHub

Per visualizzare tutte le metriche Prometheus

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nell'elenco dei namespace, scegli ContainerInsights/Prometheus o ECS/ /Prometheus. ContainerInsights
4. Scegliete uno degli insiemi di quote nell'elenco seguente. Seleziona quindi la casella di controllo accanto alle metriche che desideri visualizzare.

Per visualizzare report predefiniti sulle metriche di Prometheus

1. CloudWatch Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione scegli Monitoraggio delle performance.
3. Nella casella a discesa nella parte superiore della pagina, scegli una delle opzioni Prometheus.

   Nell'altra casella a discesa scegliere un cluster da visualizzare

Abbiamo anche fornito pannelli di controllo personalizzati per NGINX, App Mesh, Memcached, HAProxy e Java/JMX.

Per utilizzare un pannello di controllo personalizzato fornito da Amazon

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo).
3. Scegli Crea pannello di controllo). Immetti un nome per il nuovo dashboard e scegli Crea dashboard.
4. In Add to this dashboard (Aggiungi a questo pannello di controllo), scegli Cancel (Annulla).
5. Seleziona Actions (Operazioni), View/edit source (Visualizza/modifica origine).
6. Scaricare uno dei seguenti file JSON:

   - Origine pannello di controllo personalizzato NGINX su Github.
   - Origine pannello di controllo personalizzato App Mesh su Github.
   - Origine pannello di controllo personalizzato Memcached su Github

- Origine pannello di controllo personalizzato HAProxy-Ingress su Github
- Origine pannello di controllo personalizzato Java/JMX su Github.

7. Apri il file JSON scaricato con un editor di testo e apporta le seguenti modifiche:

   - Sostituire tutte le stringhe `{{YOUR_CLUSTER_NAME}}` con il nome esatto del cluster. Assicurarti di non aggiungere spazi bianchi prima o dopo il testo.
   - Sostituisci tutte le `{{YOUR_REGION}}` stringhe con la AWS regione in cui è in esecuzione il cluster. Ad esempio, **`us-west-1`** Assicurati di non aggiungere spazi bianchi prima o dopo il testo.
   - Sostituisci tutte le stringhe `{{YOUR_NAMESPACE}}` con lo spazio dei nomi esatto del carico di lavoro.
   - Sostituisci tutte le stringhe `{{YOUR_SERVICE_NAME}}` con il nome del servizio esatto del carico di lavoro. Ad esempio, **`haproxy-haproxy-ingress-controller-metrics`**

8. Copia l'intero blob JSON e incollalo nella casella di testo della CloudWatch console, sostituendo ciò che è già presente nella casella.
9. Scegli Update (Aggiorna), Save dashboard (Salva pannello di controllo).

Risoluzione dei problemi relativi ai parametri Prometheus

Questa sezione fornisce informazioni sulla risoluzione dei problemi relativi all'impostazione delle metriche di Prometheus.

Argomenti

- Risoluzione dei problemi relativi ai parametri Prometheus su Amazon ECS
- Risoluzione dei problemi dei parametri Prometheus sui cluster Amazon EKS e Kubernetes

Risoluzione dei problemi relativi ai parametri Prometheus su Amazon ECS

Questa sezione offre informazioni sulla risoluzione dei problemi relativi all'impostazione dei parametri di Prometheus su cluster Amazon ECS.

Non vedo le metriche di Prometheus inviate a Logs CloudWatch

I parametri Prometheus devono essere importati come eventi di log nel gruppo di log /aws/ecs/containerinsights/cluster-name/Prometheus. Se il gruppo di log non viene creato o le metriche di Prometheus non vengono inviate al gruppo di log, è necessario innanzitutto verificare se gli obiettivi

Prometheus sono stati scoperti con successo dall'agente. CloudWatch Quindi controlla il gruppo di sicurezza e le impostazioni di autorizzazione dell'agente. CloudWatch I seguenti passaggi guidano l'utente per eseguire il debug.

Passaggio 1: abilitare la modalità di debug CloudWatch dell'agente

Innanzitutto, imposta l' CloudWatch agente in modalità di debug aggiungendo le seguenti righe in grassetto al file AWS CloudFormation modello, oppure. `cwagent-ecs-prometheus-metric-for-bridge-host.yaml` `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` Quindi salvare il file.

```
cwagentconfig.json: |
    {
      "agent": {
        "debug": true
      },
      "logs": {
        "metrics_collected": {
```

Crea un nuovo AWS CloudFormation changeset rispetto allo stack esistente. Imposta gli altri parametri nel changeset sugli stessi valori dello stack esistente. AWS CloudFormation L'esempio seguente riguarda un CloudWatch agente installato in un cluster Amazon ECS utilizzando il tipo di avvio EC2 e la modalità di rete bridge.

```
ECS_NETWORK_MODE=bridge
 CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
NEW_CHANGESET_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=$ECS_EXECUTION_ROLE_NAME
 \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
```

```
    --change-set-name $NEW_CHANGESET_NAME
```

Vai alla AWS CloudFormation console per esaminare il nuovo changeset,. `$NEW_CHANGESET_NAME` Dovrebbe essere applicata una modifica alla risorsa CW AgentConfig SSMParameter. Esegui il changeset e riavvia l'attività dell' CloudWatch agente immettendo i seguenti comandi.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service your_service_name_here \
--region $AWS_REGION
```

Attendi circa 10 secondi e inserisci il comando seguente.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service your_service_name_here \
--region $AWS_REGION
```

Passaggio 2: controllo dei log di individuazione del servizio ECS

La definizione del task ECS dell' CloudWatch agente abilita i log per impostazione predefinita nella sezione seguente. I log vengono inviati a Logs nel gruppo di CloudWatch log /ecs/. ecs-cwagent-prometheus

```
LogConfiguration:
  LogDriver: awslogs
    Options:
      awslogs-create-group: 'True'
      awslogs-group: "/ecs/ecs-cwagent-prometheus"
      awslogs-region: !Ref AWS::Region
      awslogs-stream-prefix: !Sub 'ecs-${ECSLaunchType}-awsvpc'
```

Filtra i log in base alla stringa `ECS_SD_Stats` per ottenere i parametri relativi all'individuazione dei servizi ECS, come mostrato nell'esempio seguente.

```
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeContainerInstances: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeInstancesRequest: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTasks: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_ListTasks: 1
```

```
2020-09-1T01:53:14Z D! ECS_SD_Stats: Exporter_DiscoveredTargetCount: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_EC2MetaData: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_ContainerInstance: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: Latency: 43.399783ms
```

Il significato di ogni parametro per un particolare ciclo di individuazione dei servizi ECS è il seguente:

- AWSCLI_DescribeContainerInstances— il numero di chiamate API effettuate. `ECS::DescribeContainerInstances`
- AWSCLI_DescribeInstancesRequest— il numero di chiamate `ECS::DescribeInstancesRequest` API effettuate.
- AWSCLI_DescribeTaskDefinition— il numero di chiamate `ECS::DescribeTaskDefinition` API effettuate.
- AWSCLI_DescribeTasks— il numero di chiamate `ECS::DescribeTasks` API effettuate.
- AWSCLI_ListTasks— il numero di chiamate `ECS::ListTasks` API effettuate.
- ExporterDiscoveredTargetCount— il numero di obiettivi Prometheus che sono stati scoperti ed esportati con successo nel file dei risultati di destinazione all'interno del contenitore.
- LRUCache_get_EC2: il numero di volte in cui i metadati delle istanze del contenitore sono MetaData stati recuperati dalla cache.
- LruCache_get_: il numero di volte in cui i metadati di definizione delle attività ECS sono stati recuperati dalla cache TaskDefinition.
- LruCache_size_: il numero di metadati di un'istanza di contenitore univoca memorizzati nella cache ContainerInstance.
- LruCache_size_ — il numero di definizioni di attività ECS univoche memorizzate nella cache TaskDefinition.
- Latency (Latenza): quanto tempo richiede il ciclo di individuazione dei servizi.

Verifica il valore di `ExporterDiscoveredTargetCount` per vedere se le destinazioni Prometheus individuate corrispondono alle tue aspettative. In caso contrario, i possibili motivi sono i seguenti:

- La configurazione dell'individuazione del servizio ECS potrebbe non corrispondere all'impostazione dell'applicazione. Per il rilevamento dei servizi basati su etichette docker, i contenitori di destinazione potrebbero non avere l'etichetta docker necessaria configurata nell'agente CloudWatch per individuarli automaticamente. Per la definizione delle attività ARN Regular

Expression-based service discovery di ECS, l'impostazione regex nell'agente potrebbe non corrispondere CloudWatch alla definizione dell'attività dell'applicazione.

- Il ruolo del task ECS dell' CloudWatch agente potrebbe non disporre dell'autorizzazione per recuperare i metadati delle attività ECS. Verifica che all' CloudWatch agente siano state concesse le seguenti autorizzazioni di sola lettura:
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeTasks`
  - `ecs:DescribeTaskDefinition`

Passaggio 3: controllo della connessione di rete e delle policy dei ruoli delle attività ECS

Se non ci sono ancora eventi di registro inviati al gruppo di log CloudWatch Logs di destinazione, anche se il valore di `Exporter_DiscoveredTargetCount` indica che sono stati scoperti obiettivi Prometheus, ciò potrebbe essere causato da una delle seguenti cause:

- L' CloudWatch agente potrebbe non essere in grado di connettersi alle porte di destinazione Prometheus. Controlla l'impostazione del gruppo di sicurezza dietro l'agente. CloudWatch L'IP privato dovrebbe consentire all' CloudWatch agente di connettersi alle porte dell'esportatore Prometheus.
- Il ruolo del task ECS dell' CloudWatch agente potrebbe non avere la policy gestita. CloudWatchAgentServerPolicy Il task role ECS dell' CloudWatch agente deve avere questa politica per poter inviare le metriche di Prometheus come eventi di registro. Se hai utilizzato il AWS CloudFormation modello di esempio per creare automaticamente i ruoli IAM, sia il ruolo di attività ECS che il ruolo di esecuzione ECS dispongono del privilegio minimo per eseguire il monitoraggio di Prometheus.

Risoluzione dei problemi dei parametri Prometheus sui cluster Amazon EKS e Kubernetes

Questa sezione offre informazioni sulla risoluzione dei problemi relativi all'impostazione dei parametri di Prometheus su cluster Amazon EKS e Kubernetes.

Procedura per la risoluzione dei problemi su Amazon EKS

Per confermare che l' CloudWatch agente è in esecuzione, inserisci il comando seguente.

```
kubectl get pod -n amazon-cloudwatch
```

L'output dovrebbe includere una riga con `cwagent-prometheus-id` nella colonna `NAME` e `Running` nel campo `STATUS column.`

Per visualizzare i dettagli sul pod in esecuzione, immetti il seguente comando. Sostituisci *pod-name* con il nome completo del tuo pod il cui nome inizia con `cw-agent-prometheus`.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

Se hai installato CloudWatch Container Insights, puoi utilizzare CloudWatch Logs Insights per interrogare i log dell' CloudWatch agente che raccoglie le metriche di Prometheus.

Per eseguire query sui log delle applicazioni

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/. CloudWatch
2. Nel riquadro di navigazione, scegli CloudWatch Logs Insights.
3. Seleziona il gruppo di log per i log delle applicazioni, /aws/containerinsights/***cluster-name***/application
4. Sostituisci l'espressione di query di ricerca con la query seguente e scegli Esegui query

```
fields ispresent(kubernetes.pod_name) as haskubernetes_pod_name, stream,
 kubernetes.pod_name, log |
filter haskubernetes_pod_name and kubernetes.pod_name like /cwagent-prometheus
```

Puoi anche confermare che le metriche e i metadati di Prometheus vengano inseriti come eventi di log. CloudWatch

Per confermare che i dati di Prometheus vengano ingeriti

1. CloudWatch Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli CloudWatch Logs Insights.
3. Selezionar/aws/containerinsights/***cluster-name***/prometheus
4. Sostituisci l'espressione di query di ricerca con la query seguente e scegli Esegui query

```
fields @timestamp, @message | sort @timestamp desc | limit 20
```

Registrazione dei parametri Prometheus eliminati

Questa versione non raccoglie i parametri Prometheus di tipo istogramma. Puoi utilizzare l' CloudWatch agente per verificare se alcune metriche di Prometheus vengono eliminate perché si tratta di metriche di istogrammi. Puoi anche registrare un elenco delle prime 500 metriche di Prometheus che vengono eliminate e non CloudWatch inviate a nessuno perché si tratta di metriche di istogrammi.

Per verificare se vengono eliminate le metriche, immetti il seguente comando:

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

Se vengono eliminate delle metriche, nel file `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` verranno visualizzate le seguenti righe.

```
I! Drop Prometheus metrics with unsupported types. Only Gauge, Counter and Summary are
 supported.
I! Please enable CWAgent debug mode to view the first 500 dropped metrics
```

Se vedi queste righe e vuoi sapere quali metriche vengono eliminate, segui la procedura seguente.

Per registrare un elenco delle parametri di Prometheus eliminati

1. Imposta l' CloudWatch agente in modalità debug aggiungendo le seguenti righe in grassetto al file `prometheus-eks.yaml` or `prometheus-k8s.yaml` e salva il file.

   ```
   {
         "agent": {
           "debug": true
         },
   ```

   Questa sezione del file dovrebbe quindi assomigliare a questa:

   ```
   cwagentconfig.json: |
       {
         "agent": {
           "debug": true
         },
         "logs": {
           "metrics_collected": {
   ```

2. Reinstalla l' CloudWatch agente per abilitare la modalità di debug inserendo i seguenti comandi:

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
kubectl apply -f prometheus.yaml
```

   Le metriche eliminate vengono registrate nel pod dell'agente. CloudWatch

3. Per recuperare i log dal contenitore dell' CloudWatch agente, immettete il seguente comando:

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

   Oppure, se hai installato la registrazione di Container Insights Fluentd, i log vengono salvati anche nel gruppo di log Logs /aws/containerinsights/ cluster_name CloudWatch /application.

   Per eseguire una query su questi registri, è possibile seguire la procedura per eseguire query sui log dell'applicazione in Procedura per la risoluzione dei problemi su Amazon EKS.

Dove vengono inserite le metriche di Prometheus come eventi di log log? CloudWatch

L' CloudWatch agente crea un flusso di log per ogni configurazione dello scrape job di Prometheus. Ad esempio, nei file `prometheus-eks.yaml` e `prometheus-k8s.yaml`, la riga `job_name: 'kubernetes-pod-appmesh-envoy'` recupera i parametri App Mesh. Il target di Prometheus è definito come `kubernetes-pod-appmesh-envoy`. Pertanto, tutte le metriche di App Mesh Prometheus vengono inserite come eventi di log nel flusso di log nel gruppo di log CloudWatch denominato /aws/ContainerInsights/Cluster-name/Prometheus. kubernetes-pod-appmesh-envoy

Non vedo le metriche di Amazon EKS o Kubernetes Prometheus nelle metriche CloudWatch

Innanzitutto, assicurati che le metriche Prometheus vengano ingerite come eventi di log nel gruppo di log /aws/containerinsights/cluster-name/Prometheus. Utilizzare le informazioni in Dove vengono inserite le metriche di Prometheus come eventi di log log? CloudWatch per controllare il flusso di log di destinazione. Se il flusso di log non viene creato o non ci sono nuovi eventi di log nel flusso di log, verificare quanto segue:

- Verificare che gli endpoint di esportazione delle metriche Prometheus siano impostati correttamente
- Verifica che le configurazioni di scraping di Prometheus nella sezione del file YAML `config map: cwagent-prometheus` dell'agente siano corrette. CloudWatch La configurazione dovrebbe

essere la stessa di un file di configurazione di Prometheus. Per ulteriori informazioni, vedere <scrape_config> nella documentazione di Prometheus.

Se le metriche di Prometheus vengono inserite correttamente come eventi di registro, verifica che le impostazioni del formato metrico incorporato vengano aggiunte agli eventi di registro per generare le metriche. CloudWatch

```
"CloudWatchMetrics":[
   {
      "Metrics":[
         {
            "Name":"envoy_http_downstream_cx_destroy_remote_active_rq"
         }
      ],
      "Dimensions":[
         [
            "ClusterName",
            "Namespace"
         ]
      ],
      "Namespace":"ContainerInsights/Prometheus"
   }
],
```

Per ulteriori informazioni sul formato della metrica incorporata, vedere Specifica: Embedded Metric Format .

Se non è presente un formato metrico incorporato negli eventi di registro, verifica che la `metric_declaration` sezione sia configurata correttamente nella sezione del file YAML di installazione dell'agente. `config map: prometheus-cwagentconfig` CloudWatch Per ulteriori informazioni, consulta la pagina Esercitazione per l'aggiunta di nuove destinazioni di scraping di Prometheus: parametri del server API Prometheus.

Integrazione con Application Insights

Amazon CloudWatch Application Insights ti aiuta a monitorare le tue applicazioni e a identificare e configurare parametri chiave, log e allarmi tra le risorse applicative e lo stack tecnologico. Per ulteriori informazioni, consulta la pagina Informazioni approfondite sulle CloudWatch applicazioni Amazon.

Puoi abilitare Application Insights per raccogliere dati aggiuntivi dalle applicazioni e dai microservizi containerizzati. Se non l'hai ancora fatto, puoi abilitarlo scegliendo Configurazione automatica di Application Insights sotto la visualizzazione delle prestazioni nel pannello di controllo di Container Insights.

Se hai già configurato CloudWatch Application Insights per monitorare le tue applicazioni containerizzate, la dashboard di Application Insights viene visualizzata sotto la dashboard di Container Insights.

Per ulteriori informazioni su Application Insights e applicazioni containerizzate, consulta Abilitazione di monitoraggio delle risorse di Application Insights per Amazon ECS e Amazon EKS.

Visualizzazione degli eventi del ciclo di vita di Amazon ECS in Approfondimenti sui container

Puoi visualizzare gli eventi del ciclo di vita di Amazon ECS nella console Approfondimenti sui container. Ciò ti consente di raggruppare i parametri, i registri e gli eventi dei container in un'unica vista, in modo da offrirti una visibilità operativa più completa.

Gli eventi includono eventi di modifica dello stato delle istanze di container, eventi di modifica dello stato delle attività ed eventi di operazioni di servizio. Vengono inviati automaticamente da Amazon ECS ad Amazon EventBridge e raccolti anche CloudWatch in formato registro eventi. Per ulteriori informazioni su questi eventi, consulta la pagina Eventi Amazon ECS.

I prezzi standard di Container Insights si applicano agli eventi del ciclo di vita di Amazon ECS. Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

Per configurare la tabella degli eventi del ciclo di vita e creare regole per un cluster, è necessario disporre delle autorizzazioni `events:PutRule`, `events:PutTargets` e `logs:CreateLogGroup`. È inoltre necessario assicurarsi che esista una politica delle risorse che EventBridge consenta di creare il flusso di log e inviare i log a Logs. CloudWatch Se questa policy delle risorse non esiste, puoi immettere il seguente comando per crearla:

```
aws --region region logs put-resource-policy --policy-name 'EventBridgeCloudWatchLogs'
 --policy-document '{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogStream",
```

```
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": ["events.amazonaws.com", "delivery.logs.amazonaws.com"]
            },
            "Resource": "arn:aws:logs:region:account-id:log-group:/aws/events/ecs/
containerinsights/*:*",
            "Sid": "TrustEventBridgeToStoreECSLifecycleLogEvents"
        }
    ],
    "Version": "2012-10-17"
}'
```

È possibile utilizzare il comando seguente per verificare se si dispone già di questa policy e per verificare il corretto funzionamento del collegamento.

```
aws logs describe-resource-policies --region region --output json
```

Per visualizzare la tabella degli eventi del ciclo di vita, è necessario disporre delle autorizzazioni `events:DescribeRule`, `events:ListTargetsByRule` e `logs:DescribeLogGroups`.

Per visualizzare gli eventi del ciclo di vita di Amazon ECS nella console Container Insights CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Scegli Insights (Informazioni dettagliate), Container Insights.
3. Scegli Visualizza dashboard delle prestazioni.
4. Nell'elenco a discesa successivo, scegli ECS Clusters (Cluster ECS), ECS Services (Servizi ECS) o ECS Tasks (Attività ECS).
5. Se nel passaggio precedente hai scelto ECS Services (Servizi ECS) o ECS Tasks (Attività ECS), seleziona la scheda Lifecycle events (Eventi del ciclo di vita).
6. Nella parte inferiore della pagina, se vedi Configura gli eventi del ciclo di vita, selezionala per creare EventBridge regole per il tuo cluster.

   Gli eventi vengono visualizzati sotto i riquadri di Approfondimenti sui container e sopra la sezione Approfondimenti sulle applicazioni. Per eseguire ulteriori analisi e creare visualizzazioni aggiuntive per questi eventi, scegli View in Logs Insights (Visualizza in Logs Insights) nella tabella Lifecycle Events (Eventi del ciclo di vita).

Risoluzione dei problemi relativi a Container Insights

Le seguenti sezioni possono supportare il processo di risoluzione dei problemi relativi a Container Insights.

Implementazione non riuscita su Amazon EKS o Kubernetes

Se l'agente non viene implementato correttamente su un cluster Kubernetes, prova quanto segue:

- Per ottenere l'elenco di pod esegui il seguente comando.

```
kubectl get pods -n amazon-cloudwatch
```

- Esegui il comando seguente e controlla gli eventi nella parte inferiore dell'output.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Esegui il comando seguente per controllare i log.

```
kubectl logs pod-name -n amazon-cloudwatch
```

Unauthorized panic: Cannot retrieve cadvisor data from kubelet (Operazione non autorizzata: impossibile recuperare i dati cadvisor dal kubelet)

Se l'implementazione non riesce e restituisce l'errore `Unauthorized panic: Cannot retrieve cadvisor data from kubelet`, è possibile che per il kubelet non sia stata abilitata la modalità di autorizzazione Webhook. Questa modalità è necessaria per Container Insights. Per ulteriori informazioni, consulta la pagina Verifica dei prerequisiti di .

Implementazione di Container Insights su un cluster eliminato e ricreato in Amazon ECS

Se elimini un cluster Amazon ECS esistente in cui Container Insights non è abilitato e lo crei nuovamente con lo stesso nome, non puoi abilitare Container Insights su questo nuovo cluster al momento della nuova creazione. Puoi abilitarlo ricreandolo e quindi immettendo il comando seguente:

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Errore di endpoint non valido

Se viene visualizzato un messaggio di errore analogo al seguente, verifica di aver sostituito tutti i segnaposto, ad esempio *cluster-name* e *region-name* nei comandi che usi con le informazioni corrette per l'implementazione.

```
"log": "2020-04-02T08:36:16Z E! cloudwatchlogs: code: InvalidEndpointURL, message:
 invalid endpoint uri, original error: &url.Error{Op:\"parse\", URL:\"https://
logs.{{region_name}}.amazonaws.com/\", Err:\"{\"}, &awserr.baseError{code:
\"InvalidEndpointURL\", message:\"invalid endpoint uri\", errs:[]error{(*url.Error)
(0xc0008723c0)}}\n",
```

I parametri non vengono visualizzati nella console

Se non vedi alcuna metrica di Container Insights in AWS Management Console, assicurati di aver completato la configurazione di Container Insights. I parametri vengono visualizzati solo dopo aver completato la configurazione di Container Insights. Per ulteriori informazioni, consulta la pagina Configurazione di Container Insights.

Parametri dei pod mancanti su Amazon EKS o Kubernetes dopo l'aggiornamento del cluster

Questa sezione può essere utile se mancano tutte o alcune metriche del pod dopo aver distribuito l' CloudWatch agente come demonset su un cluster nuovo o aggiornato oppure se viene visualizzato un registro degli errori con il messaggio. `W! No pod metric collected`

Questi errori possono essere causati da modifiche nel runtime del container, ad esempio containerd o il driver cgroup systemd docker. Di solito è possibile risolvere questo problema aggiornando il manifesto di implementazione in modo che il socket containerd dall'host sia montato nel container. Fai riferimento al file di esempio seguente:

```
# For full example see https://github.com/aws-samples/amazon-cloudwatch-container-
insights/blob/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: cloudwatch-agent
  namespace: amazon-cloudwatch
```

```
spec:
  template:
    spec:
      containers:
        - name: cloudwatch-agent
# ...
          # Don't change the mountPath
          volumeMounts:
# ...
            - name: dockersock
              mountPath: /var/run/docker.sock
              readOnly: true
            - name: varlibdocker
              mountPath: /var/lib/docker
              readOnly: true
            - name: containerdsock # NEW mount
              mountPath: /run/containerd/containerd.sock
              readOnly: true
# ...
      volumes:
# ...
        - name: dockersock
          hostPath:
            path: /var/run/docker.sock
        - name: varlibdocker
          hostPath:
            path: /var/lib/docker
        - name: containerdsock # NEW volume
          hostPath:
            path: /run/containerd/containerd.sock
```

Nessun parametro dei pod quando si utilizza Bottlerocket per Amazon EKS

Bottlerocket è un sistema operativo open source basato su Linux creato appositamente da AWS per eseguire container.

Bottlerocket utilizza un percorso `containerd` diverso sull'host, quindi è necessario modificare i volumi nella sua posizione. In caso contrario, verrà visualizzato un messaggio di errore nei log che include `W! No pod metric collected`. Guarda l'esempio seguente.

```
volumes:
  # ...
    - name: containerdsock
```

```
      hostPath:
        # path: /run/containerd/containerd.sock
        # bottlerocket does not mount containerd sock at normal place
        # https://github.com/bottlerocket-os/bottlerocket/
commit/91810c85b83ff4c3660b496e243ef8b55df0973b
        path: /run/dockershim.sock
```

Nessun parametro del filesystem del container quando si utilizza il runtime containerd per Amazon EKS o Kubernetes

Si tratta di un problema noto ed è in fase di elaborazione con il contributo della community. Per ulteriori informazioni, vedere La metrica di utilizzo del disco per containerd e Le metriche del file system del contenitore non sono supportate da cadvisor per containerd on. GitHub

Aumento imprevisto del volume di log da parte CloudWatch dell'agente durante la raccolta delle metriche di Prometheus

Si tratta di una regressione introdotta nella versione 1.247347.6b250880 dell'agente. CloudWatch Questa regressione è già stata risolta nelle versioni più recenti dell'agente. L'impatto è stato limitato agli scenari in cui i clienti raccoglievano i log dell' CloudWatch agente stesso e utilizzavano anche Prometheus. Per ulteriori informazioni, consulta [prometheus] L'agente stampa tutte le metriche eliminate in log on. GitHub

Ultima immagine Docker menzionata nelle note di rilascio non trovata da Dockerhub

Aggiorniamo la nota di rilascio e il tag su Github prima di avviare internamente la versione effettiva. Di solito ci vogliono 1-2 settimane per vedere l'ultima immagine Docker sui log dopo aver pubblicato il numero di versione su Github. Non è prevista una release notturna per l'immagine del contenitore dell'agente. CloudWatch È possibile creare l'immagine direttamente dal codice sorgente nella seguente posizione: https://github.com/aws/ amazon-cloudwatch-agent /tree/main//amazon-cloudwatch-container-insightscloudwatch-agent-dockerfile

CrashLoopBackoff errore sull'agente CloudWatch

Se vedi un `CrashLoopBackOff` errore relativo all' CloudWatch agente, assicurati che le autorizzazioni IAM siano impostate correttamente. Per ulteriori informazioni, consulta Verifica dei prerequisiti di .

CloudWatch agente o pod Fluentd bloccato in sospeso

Se hai un CloudWatch agente o un pod Fluentd bloccato `Pending` o con un `FailedScheduling` errore, determina se i tuoi nodi dispongono di risorse di elaborazione sufficienti in base al numero di core e alla quantità di RAM richiesta dagli agenti. Immetti il seguente comando per descrivere il pod:

```
kubectl describe pod cloudwatch-agent-85ppg -n amazon-cloudwatch
```

Creazione della propria immagine Docker per l'agente CloudWatch

Puoi creare la tua immagine Docker per CloudWatch l'agente facendo riferimento al Dockerfile che si trova all'indirizzo https://github.com/aws-samples/ /blob/latest/ /Dockerfile. amazon-cloudwatch-container-insights cloudwatch-agent-dockerfile

Il file Dockere supporta la creazione di immagini multi-architettura direttamente tramite `docker buildx`.

Implementazione di altre funzionalità degli agenti nei contenitori CloudWatch

È possibile implementare funzionalità di monitoraggio aggiuntive nei contenitori utilizzando l' CloudWatch agente. Le caratteristiche principali comprendono:

- Formato metrica incorporato: per ulteriori informazioni, consulta Incorporamento dei parametri nei log.
- StatsD: per ulteriori informazioni, consulta la pagina Recupero dei parametri personalizzati con StatsD .

Le istruzioni e i file necessari si GitHub trovano nelle seguenti posizioni:

- Per i container Amazon ECS, consulta Esempio di definizioni delle attività Amazon ECS basate sulle modalità di implementazione.
- Per i contenitori Amazon EKS e Kubernetes, consulta Esempio di file YAML Kubernetes basati sulle modalità di implementazione.

Lambda Insights

CloudWatch Lambda Insights è una soluzione di monitoraggio e risoluzione dei problemi per applicazioni serverless in esecuzione su. AWS Lambda La soluzione raccoglie, aggrega e riassume i parametri a livello di sistema, tra cui il tempo della CPU, la memoria, l'utilizzo del disco e della rete. Vengono inoltre raccolte, aggregate e riepilogate informazioni diagnostiche quali avvii a freddo e arresti del worker Lambda per aiutare a isolare i problemi con le funzioni Lambda e risolverli rapidamente.

Lambda Insights utilizza una nuova estensione CloudWatch Lambda, fornita come layer Lambda. Quando installi questa estensione su una funzione Lambda, raccoglie metriche a livello di sistema ed emette un singolo evento di registro delle prestazioni per ogni chiamata di quella funzione Lambda. CloudWatch utilizza la formattazione metrica incorporata per estrarre le metriche dagli eventi di registro.

Per ulteriori informazioni sulle estensioni Lambda, consulta Uso AWS Lambda delle estensioni. Per ulteriori informazioni sul formato della metrica incorporata, vedere Incorporamento dei parametri nei log.

È possibile utilizzare Lambda Insights con qualsiasi funzione Lambda che utilizza un runtime Lambda che supporta le estensioni Lambda. Per un elenco di questi runtime, consulta API delle estensioni Lambda.

Prezzi

Per ogni funzione registrare abilitata per Lambda Insights, si paga solo per ciò che si utilizza per parametri e registri. Per un esempio di prezzo, consulta la pagina CloudWatch Prezzi di Amazon.

Ti viene addebitato il tempo di esecuzione consumato dall'estensione in incrementi di 1 ms. Per maggiori informazioni sui prezzi di Lambda consulta Prezzi AWS Lambda.

Guida introduttiva a Lambda Insights

Per abilitare Lambda Insights su una funzione Lambda, puoi utilizzare un interruttore a pressione singola nella console Lambda. In alternativa, puoi usare AWS CLI, AWS CloudFormation, la AWS Serverless Application Model CLI o. AWS Cloud Development Kit (AWS CDK)

Le sezioni seguenti forniscono informazioni dettagliate su come completare questi passaggi.

Argomenti

Versioni disponibili dell'estensione Lambda Insights

Questa sezione elenca le versioni dell'estensione Lambda Insights e gli ARN da utilizzare per queste estensioni in ciascuna regione. AWS

Argomenti

piattaforme x86-64

Questa sezione elenca le versioni dell'estensione Lambda Insights per piattaforme x86-64 e gli ARN da utilizzare per queste estensioni in ciascuna regione. AWS

> **Important**
>
> Le estensioni Lambda Insights 1.0.317.0 e successive non supportano Amazon Linux 1.

1.0.317.0

La versione 1.0.317.0 include la rimozione del supporto per la piattaforma Amazon Linux 1 e correzioni di bug. Include anche il supporto per le regioni. AWS GovCloud (US)

ARNs per la versione 1.0.317.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:52`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:52`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:52`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:52`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:43`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:43`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:25`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:29`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:20`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:50`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:33`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:51`

Regione	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:52
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:52
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:79
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:51
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:12
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:42
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:42
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:52
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:52
Europa (Londra)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:52
Europa (Milano)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:43
Europa (Parigi)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:51
Europa (Spagna)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:27

Regione	ARN
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:49`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:26`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:20`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:43`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:26`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:51`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension:19`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension:19`

1,0295,0

La versione 1.0.295.0 include aggiornamenti delle dipendenze per tutti i runtime compatibili.

ARN per la versione 1.0.295.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:51`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:51`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:51`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:51`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:42`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:42`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:24`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:28`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:19`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:49`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:32`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:50`

Regione	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:51
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:51
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:78
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:50
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:11
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:41
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:41
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:51
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:51
Europa (Londra)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:51
Europa (Milano)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:42
Europa (Parigi)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:50
Europa (Spagna)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:26

Regione	ARN
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:48`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:25`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:19`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:42`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:25`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:50`

1.0.275.0

La versione 1.0.275.0 include importanti aggiornamenti delle dipendenze per tutti i runtime compatibili.

ARN per la versione 1.0.275.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:49`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:49`

Regione	ARN
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:49`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:49`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:40`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:40`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:22`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:26`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:17`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:47`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:30`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:48`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:49`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:49`

Regione	ARN
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:76
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:48
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:9
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:39
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:39
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:49
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:49
Europa (Londra)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:49
Europa (Milano)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:40
Europa (Parigi)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:48
Europa (Spagna)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:24
Europa (Stoccolma)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:46
Europa (Zurigo)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:23

Regione	ARN
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:17`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:40`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:23`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:48`

1.0.273.0

La versione 1.0.273.0 include importanti correzioni di bug per tutti i runtime compatibili e aggiunge il supporto per Canada West (Calgary).

ARNs per la versione 1.0.273.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:45`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:45`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:45`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:45`

Regione	ARN
Africa (Città del Capo)	arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:35
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:35
Asia Pacific (Hyderabad)	arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:17
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:21
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:12
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:43
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:26
Asia Pacific (Seul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:44
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:45
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:45
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:72
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:44
Canada occidentale (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:4

Regione	ARN
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:36`
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:36`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:45`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:45`
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:45`
Europa (Milano)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:35`
Europa (Parigi)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:44`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:19`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:42`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:17`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:12`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:35`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:18`

Regione	ARN
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:44`

1.0.229.0

La versione 1.0.229.0 include importanti correzioni di bug per tutti i runtime compatibili e aggiunge il supporto per la regione di Israele (Tel Aviv).

ARN per la versione 1.0.229.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:38`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:38`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:38`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:38`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:28`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:28`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:10`

Regione	ARN
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:14
Asia Pacifico (Melbourne)	arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:5
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:36
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:19
Asia Pacific (Seul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:37
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:38
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:38
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:60
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:37
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:29
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:29
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:38
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:38

Regione	ARN
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:38`
Europa (Milano)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:28`
Europa (Parigi)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:37`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:12`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:35`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:11`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:5`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:28`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:11`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:37`

1.0.178.0

La versione 1.0.178.0 aggiunge il supporto per le seguenti regioni. AWS

- Asia Pacific (Hyderabad)
- Asia Pacifico (Giacarta)
- Europa (Spagna)

- Europa (Zurigo)
- Medio Oriente (Emirati Arabi Uniti)

ARN per la versione 1.0.178.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:35`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:33`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:33`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:33`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:25`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:25`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:8`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:11`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:31`

Regione	ARN
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2
Asia Pacific (Seul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:32
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:33
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:33
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:50
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:32
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:26
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:26
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:35
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:33
Europa (Londra)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:33
Europa (Milano)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:25
Europa (Parigi)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:32

Regione	ARN
Europa (Spagna)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:10`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:30`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:7`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:25`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:9`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:32`

1.0.143.0

La versione 1.0.143.0 include correzioni di bug in compatibilità con Python 3.7 e Go 1.x. Il runtime Lambda di Python 3.6 sta per essere dichiarato obsoleto. Per ulteriori informazioni, consulta Runtime Lambda.

ARN per la versione 1.0.143.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:21`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:21`

Regione	ARN
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:20`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:21`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:13`
Asia Pacific (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:13`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:21`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:20`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:21`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:21`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:32`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:20`
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:14`

Regione	ARN
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:14`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:21`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:21`
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:21`
Europa (Milano)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:13`
Europa (Parigi)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:20`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:20`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:13`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:20`

1.0.135.0

La versione 1.0.135.0 include correzioni di bug per come Lambda Insights raccoglie e segnala l'utilizzo del disco e del descrittore di file. Nelle versioni precedenti dell'estensione, il parametro `tmp_free` ha riportato lo spazio libero massimo nella directory `/tmp` durante l'esecuzione di una funzione. Questa versione modifica la metrica per segnalare invece il valore minimo, rendendola più utile quando si valuta l'utilizzo del disco. Per ulteriori informazioni sulle quote di archiviazione delle directory `tmp`, consulta Quote di Lambda.

La versione 1.0.135.0 ora riporta anche l'utilizzo del descrittore di file (`fd_use` e `fd_max`) come valore massimo tra i processi anziché segnalare il livello del sistema operativo.

ARN per la versione 1.0.135.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:18`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:18`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:18`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:18`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:11`
Asia Pacific (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:11`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:18`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:1`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:18`

Regione	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:18
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:18
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:25
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:18
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:11
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:11
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:18
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:18
Europa (Londra)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:18
Europa (Milano)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:11
Europa (Parigi)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:18
Europa (Stoccolma)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:18
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:11

Regione	ARN
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:18`

1.0.119.0

ARN per la versione 1.0.119.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:16`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:16`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:16`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:16`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:9`
Asia Pacific (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:9`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:16`
Asia Pacifico (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:16`

Regione	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:16
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:16
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:23
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:16
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:9
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:9
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:16
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:16
Europa (Londra)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:16
Europa (Milano)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:9
Europa (Parigi)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:16
Europa (Stoccolma)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:16
Medio Oriente (Bahrein)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:9

Regione	ARN
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:16`

1.0.98.0

Questa versione rimuove la registrazione non necessaria e risolve anche un problema con le invocazioni locali della CLI AWS Serverless Application Model . Per ulteriori informazioni su questo problema, consulta Aggiungere LambdaInsightsExtension risultati in timeout con 'sam local invoke'.

ARN per la versione 1.0.98.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:14`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:14`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:14`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:8`
Asia Pacific (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:8`

Regione	ARN
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:14
Asia Pacifico (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:14
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:14
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:14
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:14
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:14
Cina (Pechino)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:8
Cina (Ningxia)	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:8
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:14
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:14
Europa (Londra)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:14
Europa (Milano)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:8
Europa (Parigi)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:14

Regione	ARN
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:14`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:8`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:14`

1.0.89.0

Questa versione corregge il timestamp dell'evento di prestazioni per rappresentare sempre l'inizio della chiamata della funzione.

ARN per la versione 1.0.89.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:12`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:12`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:12`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:12`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:12`

Regione	ARN
Asia Pacifico (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:12`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:12`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:12`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:12`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:12`
Europe (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:12`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:12`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:12`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:12`

1.0.86.0

Con la versione 1.0.54.0 dell'estensione, i parametri di memoria sono stati talvolta riportati in modo errato e talvolta erano superiori al 100%. La versione 1.0.86.0 corregge il problema di misurazione

della memoria utilizzando gli stessi dati dell'evento come parametri della piattaforma Lambda. Questo significa che potresti vedere un cambiamento radicale nei valori dei parametri della memoria registrata. Ciò si ottiene utilizzando la nuova API Lambda Logs. Questa fornisce una misurazione più accurata dell'utilizzo della memoria sandbox Lambda. Tuttavia, occorre sapere che l'API Lambda Logs non può fornire eventi di report della piattaforma se una sandbox di funzioni va in timeout e viene successivamente disattivata. In questo caso. Lambda Insights non è in grado di registrare i parametri di invocazione. Per ulteriori informazioni sull'API Lambda Logs, consulta API di AWS Lambda Logs.

Nuove funzionalità nella versione 1.0.86.0

- Utilizza l'API Lambda Logs per correggere il parametro della memoria. Questo risolve il problema precedente in cui le statistiche della memoria erano superiori al 100%.
- Viene introdotta `Init Duration` come nuova metrica. CloudWatch
- Utilizza la chiamata ARN per aggiungere un versione dimensione per gli alias e le versioni invocate. Se si utilizzano alias o versioni Lambda per ottenere distribuzioni incrementali (ad esempio distribuzioni blu/verde), è possibile visualizzare i parametri in base all'alias richiamato. La versione non viene applicata se la funzione non utilizza un alias o una versione. Per ulteriori informazioni, consulta Alias della funzione Lambda.
- Aggiunge un `billed_mb_ms field` agli eventi di prestazioni per visualizzare il costo per invocazione. Questo non considera alcun costo associato alla simultaneità sottoposta a provisioning.
- Aggiunge `billed_duration` e `duration` agli eventi di prestazioni.

ARN per la versione 1.0.86.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:11`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:11`

Regione	ARN
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:11`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:11`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:11`
Asia Pacifico (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:11`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:11`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:11`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:11`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:11`
Europe (London)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:11`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:11`

Regione	ARN
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:11`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:11`

1.0.54.0

La versione 1.0.54.0 era la versione iniziale dell'estensione Lambda Insights.

ARN per la versione 1.0.54.0

La tabella seguente elenca gli ARN da utilizzare per questa versione dell'estensione in ogni AWS regione in cui è disponibile.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:2`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:2`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:2`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:2`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:2`
Asia Pacifico (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:2`

Regione	ARN
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:2
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:2
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:2
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:2
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:2
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:2
Europe (London)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:2
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:2
Europa (Stoccolma)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:2
Sud America (San Paolo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:2

Piattaforme ARM64

Questa sezione elenca le versioni dell'estensione Lambda Insights per le piattaforme ARM64 e gli ARN da utilizzare per queste estensioni in ciascuna regione. AWS

> **Important**
>
> Le estensioni Lambda Insights 1.0.317.0 e successive non supportano Amazon Linux 1.

1.0.317.0

La versione 1.0.317.0 include la rimozione del supporto per la piattaforma Amazon Linux 1 e correzioni di bug. Include anche il supporto per le regioni. AWS GovCloud (US)

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:21`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:17`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:17`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:5`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:17`

Regione	ARN
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:21
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:16
Asia Pacific (Seul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:19
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:30
Canada (Centrale)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:17
Europa (Francoforte)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europa (Irlanda)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europa (Londra)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europa (Milano)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:17
Europa (Parigi)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:17
Europa (Spagna)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:5

Regione	ARN
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:17`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension-Arm64:1`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension-Arm64:1`

1,0295,0

La versione 1.0.295.0 include aggiornamenti delle dipendenze per tutti i runtime compatibili.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:20`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`

Regione	ARN
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:16`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:16`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:4`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:16`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:20`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:15`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:17`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:29`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`

Regione	ARN
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Europa (Milano)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:16`
Europa (Parigi)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:4`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:16`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`

1.0.275.0

La versione 1.0.275.0 include correzioni di bug per tutti i runtime compatibili e supporto per le regioni Europa (Spagna) e Asia Pacifico (Hyderabad).

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`

Regione	ARN
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Africa (Città del Capo)	arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:14
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:14
Asia Pacific (Hyderabad)	arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:2
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:14
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:13
Asia Pacific (Seul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:15
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Asia Pacifico (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:27

Regione	ARN
Canada (Centrale)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Europa (Milano)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:14`
Europa (Parigi)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:2`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:14`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`

1.0.273.0

La versione 1.0.273.0 include correzioni di bug per tutti i runtime compatibili.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:12
Stati Uniti orientali (Ohio)	arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:14
Stati Uniti occidentali (California settentrionale)	arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:9
Stati Uniti occidentali (Oregon)	arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12
Africa (Città del Capo)	arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:9
Asia Pacifico (Hong Kong)	arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:9
Asia Pacifico (Giacarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:9
Asia Pacifico (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Asia Pacific (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:9
Asia Pacific (Seul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:11
Asia Pacifico (Singapore)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:12
Asia Pacifico (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:12

Regione	ARN
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:23`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europa (Milano)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:9`
Europa (Parigi)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:10`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:9`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`

1.0.229.0

La versione 1.0.229.0 include correzioni di bug per tutti i runtime compatibili. Aggiunge inoltre il supporto per le seguenti Regioni:

- Stati Uniti occidentali (California settentrionale)
- Africa (Città del Capo)

- Asia Pacifico (Hong Kong)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Osaka-Locale)
- Asia Pacific (Seul)
- Canada (Centrale)
- Europa (Milano)
- Europa (Parigi)
- Europa (Stoccolma)
- Medio Oriente (Bahrein)
- Sud America (San Paolo)

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:7`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Stati Uniti occidentali (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:2`

Regione	ARN
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:7`
Asia Pacific (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:4`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:11`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europa (Spagna)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:2`
Europa (Parigi)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Europa (Stoccolma)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`

Regione	ARN
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:2`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`

1.0.135.0

La versione 1.0.135.0 include correzioni di bug per come Lambda Insights raccoglie e segnala l'utilizzo del disco e del descrittore di file. Nelle versioni precedenti dell'estensione, il parametro `tmp_free` ha riportato lo spazio libero massimo nella directory `/tmp` durante l'esecuzione di una funzione. Questa versione modifica la metrica per segnalare invece il valore minimo, rendendola più utile quando si valuta l'utilizzo del disco. Per ulteriori informazioni sulle quote di archiviazione delle directory `tmp`, consulta Quote di Lambda.

La versione 1.0.135.0 ora riporta anche l'utilizzo del descrittore di file (`fd_use` e `fd_max`) come valore massimo tra i processi anziché segnalare il livello del sistema operativo.

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
US West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`

Regione	ARN
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`

1.0.119.0

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`
US West (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`

Regione	ARN
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europa (Londra)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`

Utilizzo della console per abilitare Lambda Insights su una funzione Lambda esistente

Utilizza i seguenti passaggi nella console Lambda per abilitare Lambda Insights su una funzione Lambda esistente.

Per attivare Lambda Insights su una funzione Lambda

1. Apri la console all'indirizzo https://console.aws.amazon.com/lambda/. AWS Lambda
2. Scegli il nome di una funzione, quindi seleziona la scheda Configuration (Configurazione) nella schermata seguente.
3. Nella scheda Configurazione, scegli Strumenti di monitoraggio e operazioni nel menu di navigazione a sinistra, quindi scegli Modifica.

   Viene visualizzata una schermata in cui è possibile modificare gli strumenti di monitoraggio.
4. Per il monitoraggio avanzato di Lambda Insights, scegli Modifica.
5. In CloudWatch Lambda Insights, abilita il monitoraggio avanzato, quindi scegli Salva.

Utilizzo AWS CLI di per abilitare Lambda Insights su una funzione Lambda esistente

Segui questi passaggi per abilitare Lambda Insights su una funzione Lambda esistente. AWS CLI

Fase 1: aggiornamento delle autorizzazioni della funzione

Per aggiornare le autorizzazioni della funzione

- CloudWatchLambdaInsightsExecutionRolePolicyAssocia la policy IAM gestita al ruolo di esecuzione della funzione immettendo il seguente comando.

```
aws iam attach-role-policy \
--role-name function-execution-role \
--policy-arn "arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy"
```

Fase 2: installazione dell'estensione Lambda

Installa l'estensione Lambda immettendo il seguente comando. Sostituisci il valore ARN per il parametro `layers` con l'ARN corrispondente alla regione e alla versione di estensione da utilizzare. Per ulteriori informazioni, consulta Versioni disponibili dell'estensione Lambda Insights.

```
aws lambda update-function-configuration \
   --function-name function-name \
   --layers "arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14"
```

Passaggio 3: abilitare l'endpoint CloudWatch VPC Logs

Questo passaggio è necessario solo per le funzioni eseguite in una sottorete privata senza accesso a Internet e se non è già stato configurato un endpoint VPC (Virtual Private Cloud) di CloudWatch Logs.

Se è necessario eseguire questa fase, immetti il seguente comando, sostituendo i placeholder con le informazioni per il VPC.

Per ulteriori informazioni, consulta Utilizzo dei CloudWatch log con gli endpoint VPC dell'interfaccia.

```
aws ec2 create-vpc-endpoint \
--vpc-id vpcId \
--vpc-endpoint-type Interface \
--service-name com.amazonaws.region.logs \
--subnet-id subnetId
--security-group-id securitygroupId
```

Utilizzo della AWS SAM CLI per abilitare Lambda Insights su una funzione Lambda esistente

Segui questi passaggi per abilitare Lambda Insights su una funzione Lambda esistente. AWS SAM AWS CLI

Se non hai già installato l'ultima versione della AWS SAM CLI, devi prima installarla o aggiornarla. Per ulteriori informazioni, consulta Installazione della AWS SAM CLI.

Fase 1: installazione del livello

Per rendere disponibile l'estensione Lambda Insights per tutte le funzioni Lambda, aggiungi una proprietà `Layers` alla sezione `Globals` del modello SAM con l'ARN del livello Lambda Insights. L'esempio seguente utilizza il livello per la versione iniziale di Lambda Insights. Per la versione più recente del livello di estensione Lambda Insights, consulta Versioni disponibili dell'estensione Lambda Insights.

```
Globals:
  Function:
    Layers:
      - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Per abilitare questo livello solo per una singola funzione, aggiungi la proprietà `Layers` alla funzione come illustrato in questo esempio.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Fase 2: aggiunta della policy gestita

Per ogni funzione, aggiungi la policy CloudWatchLambdaInsightsExecutionRolePolicyIAM.

AWS SAM non supporta le politiche globali, quindi è necessario abilitarle singolarmente su ciascuna funzione, come mostrato in questo esempio. Per ulteriori informazioni sulle policy globali, consulta la sezione Policy globali.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Policies:
        - CloudWatchLambdaInsightsExecutionRolePolicy
```

Invocazione locale

La AWS SAM CLI supporta le estensioni Lambda. Tuttavia, ogni invocazione eseguita localmente reimposta l'ambiente di runtime. I dati di Lambda Insights non saranno disponibili dalle invocazioni locali perché il runtime viene riavviato senza un evento di arresto. Per ulteriori informazioni, consulta la Release 1.6.0 - Aggiungere il supporto per il test locale delle estensioni. AWS Lambda

Risoluzione dei problemi

Per risolvere i problemi di installazione di Lambda Insights, aggiungi la seguente variabile di ambiente alla tua funzione Lambda per abilitare la registrazione di debug.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Environment:
        Variables:
          LAMBDA_INSIGHTS_LOG_LEVEL: info
```

Utilizzo AWS CloudFormation per abilitare Lambda Insights su una funzione Lambda esistente

Segui questi passaggi per AWS CloudFormation abilitare Lambda Insights su una funzione Lambda esistente.

Fase 1: installazione del livello

Aggiungere il livello Lambda Insights alla proprietà `Layers` all'interno dell'ARN del livello Lambda Insight. L'esempio seguente utilizza il livello per la versione iniziale di Lambda Insights. Per la versione più recente del livello di estensione Lambda Insights, consulta Versioni disponibili dell'estensione Lambda Insights.

```
Resources:
  MyFunction:
    Type: AWS::Lambda::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Fase 2: aggiunta della policy gestita

Aggiungi la policy CloudWatchLambdaInsightsExecutionRolePolicyIAM al tuo ruolo di esecuzione della funzione.

```
Resources:
  MyFunctionExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy'
```

Fase 3: aggiunta dell'endpoint VPC (facoltativo)

Questo passaggio è necessario solo per le funzioni eseguite in una sottorete privata senza accesso a Internet e se non è già stato configurato un endpoint VPC (Virtual Private Cloud) di CloudWatch Logs. Per ulteriori informazioni, consulta Utilizzo dei CloudWatch log con gli endpoint VPC dell'interfaccia.

```
Resources:
  CloudWatchLogsVpcPrivateEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      PrivateDnsEnabled: 'true'
      VpcEndpointType: Interface
      VpcId: !Ref: VPC
      ServiceName: !Sub com.amazonaws.${AWS::Region}.logs
      SecurityGroupIds:
        - !Ref InterfaceVpcEndpointSecurityGroup
      SubnetIds:
        - !Ref PublicSubnet01
        - !Ref PublicSubnet02
        - !Ref PublicSubnet03
```

Utilizzo AWS CDK di per abilitare Lambda Insights su una funzione Lambda esistente

Segui questi passaggi per abilitare Lambda Insights su una funzione Lambda esistente. AWS CDK Per utilizzare questi passaggi, devi già utilizzare il AWS CDK per gestire le tue risorse.

I comandi di questa sezione sono disponibili TypeScript.

Innanzitutto, aggiorna le autorizzazioni della funzione.

```
executionRole.addManagedPolicy(
 ManagedPolicy.fromAwsManagedPolicyName('CloudWatchLambdaInsightsExecutionRolePolicy')
);
```

Quindi, installa l'estensione sulla funzione Lambda. Sostituisci il valore ARN per il parametro `layerArn` con l'ARN corrispondente alla regione e alla versione di estensione da utilizzare. Per ulteriori informazioni, consulta Versioni disponibili dell'estensione Lambda Insights.

```
import lambda = require('@aws-cdk/aws-lambda');
const layerArn = 'arn:aws:lambda:us-
west-1:580247275435:layer:LambdaInsightsExtension:14';
const layer = lambda.LayerVersion.fromLayerVersionArn(this, 'LayerFromArn', layerArn);
```

Se necessario, abilita l'endpoint del cloud privato virtuale (VPC) per Logs. CloudWatch Questo passaggio è necessario solo per le funzioni eseguite in una sottorete privata senza accesso a Internet e se non hai già configurato un endpoint CloudWatch VPC Logs.

```
const cloudWatchLogsEndpoint = vpc.addInterfaceEndpoint('cwl-gateway', {
  service: InterfaceVpcEndpointAwsService.CLOUDWATCH_LOGS,
});

cloudWatchLogsEndpoint.connections.allowDefaultPortFromAnyIpv4();
```

Utilizzo di Serverless Framework per abilitare Lambda Insights su una funzione Lambda esistente

Procedi come segue per utilizzare Serverless Framework per abilitare Lambda Insights su una funzione Lambda esistente. Per ulteriori informazioni su Serverless Framework, consulta serverless.com.

Questo viene fatto tramite un plugin Lambda Insights per Serverless. Per ulteriori informazioni, vedere. serverless-plugin-lambda-insights

Se non disponi già della versione più recente dell'interfaccia a riga di comando serverless installata, devi prima installarla o aggiornarla. Per ulteriori informazioni, consulta Guida introduttiva a Serverless Framework Open Source & AWS.

Per utilizzare Serverless Framework per abilitare Lambda Insights su una funzione Lambda

1. Installa il plugin Serverless per Lambda Insights eseguendo il seguente comando nella directory Serverless:

```
npm install --save-dev serverless-plugin-lambda-insights
```

2. Nel tuo file`serverless.yml`, aggiungi il plugin nella sezione `plugins` del file come mostrato:

```
provider:
  name: aws
plugins:
  - serverless-plugin-lambda-insights
```

3. Abilitazione di Lambda Insights.
   - Puoi abilitare Lambda Insights per ogni singola funzione aggiungendo la seguente proprietà al file serverless.yml

     ```
     functions:
       myLambdaFunction:
         handler: src/app/index.handler
         lambdaInsights: true #enables Lambda Insights for this function
     ```

   - Puoi abilitare Lambda Insights per tutte le funzioni all'interno del file `serverless.yml` aggiungendo la seguente sezione personalizzata:

     ```
     custom:
       lambdaInsights:
         defaultLambdaInsights: true #enables Lambda Insights for all functions
     ```

4. Reimplementa il servizio Serverless immettendo il seguente comando:

```
serverless deploy
```

Questo reimplementa tutte le funzioni e abilita Lambda Insights per le funzioni specificate. Consente di abilitare Lambda Insights aggiungendo il livello Lambda Insights e allegando le autorizzazioni necessarie utilizzando la policy IAM `arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy`.

Abilitazione di Lambda Insights su un'implementazione di immagini del container Lambda

Per abilitare Lambda Insights su una funzione Lambda implementata come immagine di container, aggiungi le righe nel Dockerfile. Queste righe installano l'agente Lambda Insights come estensione nell'immagine di container. Le righe da aggiungere sono diverse per i container x86-64 e i container ARM64.

> **Note**
>
> L'agente Lambda Insights è supportato solo sui runtime Lambda che utilizzano Amazon Linux 2.

Argomenti

- Implementazione di immagini di container x86-64
- Implementazione dell'immagine di container ARM64

Implementazione di immagini di container x86-64

Per abilitare Lambda Insights su una funzione Lambda implementata come immagine di container su un container x86-64, aggiungi le seguenti righe nel Dockerfile. Queste righe installano l'agente Lambda Insights come estensione nell'immagine di container.

```
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm
```

Dopo aver creato la funzione Lambda, assegna la policy CloudWatchLambdaInsightsExecutionRolePolicyIAM al ruolo di esecuzione della funzione e Lambda Insights viene abilitato sulla funzione Lambda basata sull'immagine del contenitore.

> **Note**
>
> Per utilizzare una versione precedente dell'estensione Lambda Insights, sostituisci l'URL nei comandi precedenti con questo URL: `https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension.1.0.111.0.rpm`. Attualmente sono disponibili solo versioni 1.0.111.0 e successive di Lambda Insights. Per ulteriori informazioni, consulta Versioni disponibili dell'estensione Lambda Insights.

Per verificare la firma del pacchetto dell'agente Lambda Insights su un server Linux

1. Inserisci il seguente comando per scaricare la chiave pubblica.

```
shell$ wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
lambda-insights-extension.gpg
```

2. Immetti il comando seguente per importare la chiave pubblica nel keyring.

```
shell$ gpg --import lambda-insights-extension.gpg
```

   L'output sarà simile al seguente: Prendi nota del valore `key` poiché sarà necessario nella fase successiva. In questo output di esempio, il valore della chiave è `848ABDC8`.

```
gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
gpg: Total number processed: 1
gpg: imported: 1  (RSA: 1)
```

3. Verifica l'impronta digitale eseguendo il seguente comando: Sostituisci `key-value` con il valore della chiave del passaggio precedente.

```
shell$  gpg --fingerprint key-value
```

   La stringa dell'impronta nell'output di questo comando dovrebbe essere `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8`. Se la stringa dell'impronta non corrisponde, non installare l'agente e contatta AWS.

4. Dopo aver verificato l'impronta, puoi utilizzarla per verificare il pacchetto dell'agente Lambda Insights. Scarica il file della firma del pacchetto immettendo il seguente comando:

```
shell$  wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm.sig
```

5. Verifica la firma eseguendo il seguente comando:

```
shell$ gpg --verify lambda-insights-extension.rpm.sig lambda-insights-extension.rpm
```

L'output deve essere simile al seguente:

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

Nell'output previsto, potrebbe esserci un avviso relativo a una firma attendibile. Una chiave è considerata attendibile solo se è stata firmata dall'utente o da un firmatario fidato. Questo non significa che la firma non sia valida, ma soltanto che la chiave pubblica non è stata verificata.

Se l'output contiene `BAD signature`, verifica di avere eseguito i passaggi correttamente. Se continui a ricevere una `BAD signature` risposta, contatta AWS ed evita di utilizzare il file scaricato.

Esempio x86-64

Questa sezione include un esempio di abilitazione di Lambda Insights su una funzione Lambda Python basata su immagini di container.

Un esempio di abilitazione di Lambda Insights su un'immagine del container Lambda

1. Crea un Dockerfile simile a quello riportato di seguito:

```
FROM public.ecr.aws/lambda/python:3.8

// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm
```

```
COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Crea un file Python denominato `index.py` simile a quello riportato di seguito.

```
def handler(event, context):
  return {
    'message': 'Hello World!'
  }
```

3. Inserisci il Dockerfile e `index.py` nella stessa directory. Quindi, in tale directory, esegui i passaggi qui descritti per creare l'immagine Docker e caricarla su Amazon ECR.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Usa l'immagine Amazon ECR appena creata per creare la funzione Lambda.
5. Assegna la policy CloudWatchLambdaInsightsExecutionRolePolicyIAM al ruolo di esecuzione della funzione.

Implementazione dell'immagine di container ARM64

Per abilitare Lambda Insights su una funzione Lambda implementata come immagine di container in esecuzione su un container AL2_aarch64 (che utilizza l'architettura ARM64), aggiungi le seguenti righe nel Dockerfile. Queste righe installano l'agente Lambda Insights come estensione nell'immagine di container.

```
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension-arm64.rpm && \
```

```
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm
```

Dopo aver creato la funzione Lambda, assegna la policy CloudWatchLambdaInsightsExecutionRolePolicyIAM al ruolo di esecuzione della funzione e Lambda Insights viene abilitato sulla funzione Lambda basata sull'immagine del contenitore.

> **Note**
>
> Per utilizzare una versione precedente dell'estensione Lambda Insights, sostituisci l'URL nei comandi precedenti con questo URL: `https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.1.0.229.0.rpm`. Attualmente sono disponibili solo versioni 1.0.229.0 e successive di Lambda Insights. Per ulteriori informazioni, consulta Versioni disponibili dell'estensione Lambda Insights.

Per verificare la firma del pacchetto dell'agente Lambda Insights su un server Linux

1. Inserisci il seguente comando per scaricare la chiave pubblica.

   ```
   shell$ wget https://lambda-insights-extension-arm64.s3-ap-
   northeast-1.amazonaws.com/lambda-insights-extension.gpg
   ```

2. Immetti il comando seguente per importare la chiave pubblica nel keyring.

   ```
   shell$ gpg --import lambda-insights-extension.gpg
   ```

   L'output sarà simile al seguente: Prendi nota del valore `key` poiché sarà necessario nella fase successiva. In questo output di esempio, il valore della chiave è `848ABDC8`.

   ```
   gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
   gpg: Total number processed: 1
   gpg: imported: 1  (RSA: 1)
   ```

3. Verifica l'impronta digitale eseguendo il seguente comando: Sostituisci `key-value` con il valore della chiave del passaggio precedente.

   ```
   shell$  gpg --fingerprint key-value
   ```

La stringa dell'impronta nell'output di questo comando dovrebbe essere `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8`. Se la stringa dell'impronta non corrisponde, non installare l'agente e contatta AWS.

4. Dopo aver verificato l'impronta, puoi utilizzarla per verificare il pacchetto dell'agente Lambda Insights. Scarica il file della firma del pacchetto immettendo il seguente comando:

```
shell$  wget https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm.sig
```

5. Verifica la firma eseguendo il seguente comando:

```
shell$ gpg --verify lambda-insights-extension-arm64.rpm.sig lambda-insights-
extension-arm64.rpm
```

L'output deve essere simile al seguente:

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

Nell'output previsto, potrebbe esserci un avviso relativo a una firma attendibile. Una chiave è considerata attendibile solo se è stata firmata dall'utente o da un firmatario fidato. Questo non significa che la firma non sia valida, ma soltanto che la chiave pubblica non è stata verificata.

Se l'output contiene `BAD signature`, verifica di avere eseguito i passaggi correttamente. Se continui a ricevere una `BAD signature` risposta, contatta AWS ed evita di utilizzare il file scaricato.

Esempio ARM64

Questa sezione include un esempio di abilitazione di Lambda Insights su una funzione Lambda Python basata su immagini di container.

Un esempio di abilitazione di Lambda Insights su un'immagine del container Lambda

1. Crea un Dockerfile simile a quello riportato di seguito:

```
FROM public.ecr.aws/lambda/python:3.8
// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Crea un file Python denominato `index.py` simile a quello riportato di seguito.

```
def handler(event, context):
  return {
    'message': 'Hello World!'
  }
```

3. Inserisci il Dockerfile e `index.py` nella stessa directory. Quindi, in tale directory, esegui i passaggi qui descritti per creare l'immagine Docker e caricarla su Amazon ECR.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Usa l'immagine Amazon ECR appena creata per creare la funzione Lambda.
5. Assegna la policy CloudWatchLambdaInsightsExecutionRolePolicyIAM al ruolo di esecuzione della funzione.

Visualizzazione dei parametri di Lambda Insights

Dopo aver installato l'estensione Lambda Insights su una funzione Lambda che è stata richiamata, puoi utilizzare la CloudWatch console per visualizzare le tue metriche. Puoi visualizzare una panoramica multifunzione o concentrarti su una singola funzione.

Per un elenco di parametri Lambda Insights, consulta Parametri raccolti da Lambda Insights.

Per visualizzare la panoramica multifunzione dei parametri Lambda Insights

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nel riquadro di navigazione sinistro, sotto Lambda Insights, scegli Multi-function (Multifunzione).

   Nella parte superiore della pagina vengono visualizzati grafici con parametri aggregati di tutte le funzioni Lambda nella regione in cui è abilitato Lambda Insights. In basso nella pagina è riportata una tabella che elenca le funzioni.
3. Per filtrare in base al nome della funzione per ridurre il numero di funzioni visualizzate, digita parte del nome della funzione nella casella accanto alla parte superiore della pagina.
4. Per aggiungere questa visualizzazione a un pannello di controllo come widget, scegli Add to dashboard (Aggiungi a pannello di controllo).

Per visualizzare i parametri per una singola funzione

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione sinistro, sotto Lambda Insights, scegli Single-function (Funzione singola).

   Nella parte superiore della pagina vengono visualizzati i grafici con i parametri per la funzione selezionata.
3. Se hai attivato X-Ray, puoi scegliere un singolo ID traccia. Questo apre la pagina della mappa del tracciamento X-Ray per tale invocazione e da lì è possibile ridurre per vedere la traccia distribuita e gli altri servizi coinvolti nella gestione di quella specifica transazione. Per ulteriori informazioni sulla mappa del tracciamento X-Ray, consulta Uso della mappa del tracciamento X-Ray.
4. Per aprire CloudWatch Logs Insights e ingrandire un errore specifico, scegli Visualizza i log in base alla tabella nella parte inferiore della pagina.

5. Per aggiungere questa visualizzazione a un pannello di controllo come widget, scegli Add to dashboard (Aggiungi a pannello di controllo).

Integrazione con Application Insights

Amazon CloudWatch Application Insights ti aiuta a monitorare le tue applicazioni e a identificare e configurare parametri chiave, log e allarmi tra le risorse applicative e lo stack tecnologico. Per ulteriori informazioni, consulta Informazioni approfondite sulle CloudWatch applicazioni Amazon.

È possibile abilitare Application Insights per raccogliere dati aggiuntivi dalle funzioni Lambda. Se non è stato ancora fatto, è possibile abilitarlo scegliendo Configurazione automatica di Application Insights sotto la visualizzazione delle prestazioni nel pannello di controllo di Lambda Insights.

Se hai già configurato CloudWatch Application Insights per monitorare le tue funzioni Lambda, la dashboard di Application Insights viene visualizzata sotto la dashboard Lambda Insights, nella scheda Application Insights.

Parametri raccolti da Lambda Insights

Lambda Insights raccoglie diversi parametri dalle funzioni Lambda in cui è installato. Alcune di queste metriche sono disponibili come dati aggregati di serie temporali in Metrics. CloudWatch Altre metriche non vengono aggregate in dati di serie temporali, ma possono essere trovate nelle voci di registro in formato metrico incorporato utilizzando Logs Insights. CloudWatch

Le seguenti metriche sono disponibili come dati aggregati di serie temporali in Metrics nel namespace. CloudWatch `LambdaInsights`

Nome parametro	Dimensioni	Descrizione
`cpu_total_time`	function_name function_name, versione	Somma di `cpu_system_time` e `cpu_user_time` . Unità: millisecondi
`init_duration`	function_name function_name, versione	La quantità di tempo dedicato alla fase `init` del ciclo di

Nome parametro	Dimensioni	Descrizione
		vita dell'ambiente di esecuzione Lambda. Unità: millisecondi
`memory_utilization`	function_name function_name, versione	La memoria massima misurata come percentuale della memoria allocata per la funzione. Unità: percentuale
`rx_bytes`	function_name function_name, versione	Il numero di byte ricevuti dalla funzione. Unità: byte
`tmp_used`		La quantità di spazio utilizzato nella directory `/tmp`. Unità: byte
`tx_bytes`	function_name function_name, versione	Il numero di byte inviati dalla funzione. Unità: byte

Nome parametro	Dimensioni	Descrizione
`total_memory`	function_name function_name, versione	La quantità di memoria in MB allocata per la funzione Lambda. Corrisponde alla dimensione della memoria della tua funzione. Unità: megabyte
`total_network`	function_name function_name, versione	Somma di `rx_bytes` e `tx_bytes`. Anche per le funzioni che non eseguono attività I/O, questo valore è solitamente maggiore di zero a causa delle chiamate di rete effettuate dal runtime Lambda. Unità: byte
`used_memory_max`	function_name function_name, versione	Memoria misurata della sandbox di funzioni. Unità: megabyte

Le seguenti metriche sono disponibili nelle voci di registro in formato metrico incorporato utilizzando Logs Insights. CloudWatch Per ulteriori informazioni su CloudWatch Logs Insights, consulta Analisi dei dati di registro con Logs Insights. CloudWatch

Per ulteriori informazioni sul formato della metrica incorporata, vedere Incorporamento dei parametri nei log.

Nome parametro	Descrizione
`cpu_system_time`	La quantità di tempo impiegato dalla CPU per eseguire il codice del kernel. Unità: millisecondi
`cpu_total_time`	Somma di `cpu_system_time` e `cpu_user_time` . Unità: millisecondi
`cpu_user_time`	Quantità di tempo impiegato dalla CPU per l'esecuzione del codice utente. Unità: millisecondi
`fd_max`	Numero massimo di descrittori di file disponibili. Unità: numero
`fd_use`	Numero massimo di descrittori di file in uso. Unità: numero
`memory_utilization`	La memoria massima misurata come percentuale della memoria allocata per la funzione. Unità: percentuale
`rx_bytes`	Il numero di byte ricevuti dalla funzione. Unità: byte
`tx_bytes`	Il numero di byte inviati dalla funzione. Unità: byte
`threads_max`	Il numero di thread in uso dal processo della funzione. Come autore di una funzione, non hai

Nome parametro	Descrizione
	il controllo del numero iniziale di thread creati dal runtime. Unità: numero
`tmp_max`	La quantità di spazio disponibile nella directory `/tmp`. Unità: byte
`total_memory`	La quantità di memoria in MB allocata per la funzione Lambda. Corrisponde alla dimensione della memoria della tua funzione. Unità: megabyte
`total_network`	Somma di `rx_bytes` e `tx_bytes`. Anche per le funzioni che non eseguono attività I/O, questo valore è solitamente maggiore di zero a causa delle chiamate di rete effettuate dal runtime Lambda. Unità: byte
`used_memory_max`	Memoria misurata della sandbox di funzioni. Unità: byte

Risoluzione dei problemi e problemi noti

Il primo passo per risolvere eventuali problemi consiste nell'abilitare la registrazione di debug sull'estensione Lambda Insights. Per farlo, imposta la seguente variabile di ambiente sulla funzione Lambda:`LAMBDA_INSIGHTS_LOG_LEVEL=info`. Per ulteriori informazioni, vedere Utilizzo delle variabili di ambiente AWS Lambda.

L'estensione emette i log nello stesso gruppo di log della tua funzione (`/aws/lambda/`*`function-name`*`)`. Esamina questi log per verificare se l'errore potrebbe essere correlato a un problema di installazione.

Non vedo alcun parametro da Lambda Insights

Se non vedi i parametri Lambda Insights che ti aspetti di debug loggingedere, controlla le possibilità seguenti:

- Le metriche potrebbero essere semplicemente ritardate: se la funzione non è stata ancora richiamata o i dati non sono ancora stati cancellati, le metriche non verranno visualizzate. CloudWatch Per ulteriori informazioni, consulta Problemi noti più avanti in questa sezione.
- Verifica che la funzione Lambda disponga delle autorizzazioni corrette: assicurati che la policy CloudWatchLambdaInsightsExecutionRolePolicyIAM sia assegnata al ruolo di esecuzione della funzione.
- Verifica il runtime Lambda: Lambda Insights supporta solo alcuni runtime Lambda. Per un elenco dei runtime supportati, consulta Lambda Insights.

  Ad esempio, per utilizzare Lambda Insights su Java 8, è necessario utilizzare il runtime `java8.al2`, non il runtime `java8`.
- Verifica l'accesso alla rete: la funzione Lambda potrebbe trovarsi su una sottorete privata VPC senza accesso a Internet e non disponi di un endpoint VPC configurato per i log. CloudWatch Per facilitare il debug di questo problema, puoi impostare la variabile di ambiente `LAMBDA_INSIGHTS_LOG_LEVEL=info`.

Problemi noti

Il ritardo dei dati può arrivare fino a 20 minuti. Al completamento di un gestore di funzioni, Lambda blocca la sandbox, che blocca anche l'estensione Lambda Insights. Mentre la funzione è in esecuzione, usiamo una strategia di batching adattivo basata sulla funzione TPS per l'output dei dati. Tuttavia, se la funzione smette di essere invocata per un periodo prolungato e ci sono ancora dati di eventi nel buffer, questi dati possono essere ritardati fino a quando Lambda arresta la sandbox inattiva. Quando Lambda disattiva la sandbox, svuotiamo i dati nel buffer.

Esempio di evento di telemetria

Ogni invocazione di una funzione Lambda che ha abilitato Lambda Insights scrive un singolo evento di log nel gruppo di log `/aws/lambda-insights`. Ogni evento di log contiene parametri in Embedded Metric Format. Per ulteriori informazioni sul formato della metrica incorporata, vedere Incorporamento dei parametri nei log.

Per analizzare questi eventi di log, puoi utilizzare i seguenti metodi:

- La sezione Lambda Insights della CloudWatch console, come spiegato in. Visualizzazione dei parametri di Lambda Insights
- Registra le interrogazioni sugli eventi utilizzando CloudWatch Logs Insights. Per ulteriori informazioni, consulta Analisi dei dati di registro con CloudWatch Logs Insights.
- Metriche raccolte nel `LambdaInsights` namespace, che puoi rappresentare graficamente utilizzando le metriche. CloudWatch

Di seguito è riportato un esempio di evento di log Lambda Insights con Embedded Metric Format.

```
{
    "_aws": {
        "Timestamp": 1605034324256,
        "CloudWatchMetrics": [
            {
                "Namespace": "LambdaInsights",
                "Dimensions": [
                    [ "function_name" ],
                    [ "function_name", "version" ]
                ],
                "Metrics": [
                    { "Name": "memory_utilization", "Unit": "Percent" },
                    { "Name": "total_memory", "Unit": "Megabytes" },
                    { "Name": "used_memory_max", "Unit": "Megabytes" },
                    { "Name": "cpu_total_time", "Unit": "Milliseconds" },
                    { "Name": "tx_bytes", "Unit": "Bytes" },
                    { "Name": "rx_bytes", "Unit": "Bytes" },
                    { "Name": "total_network", "Unit": "Bytes" },
                    { "Name": "init_duration", "Unit": "Milliseconds" }
                ]
            }
        ],
        "LambdaInsights": {
            "ShareTelemetry": true
        }
    },
    "event_type": "performance",
    "function_name": "cpu-intensive",
    "version": "Blue",
    "request_id": "12345678-8bcc-42f7-b1de-123456789012",
    "trace_id": "1-5faae118-12345678901234567890",
    "duration": 45191,
```

```
    "billed_duration": 45200,
    "billed_mb_ms": 11571200,
    "cold_start": true,
    "init_duration": 130,
    "tmp_free": 538329088,
    "tmp_max": 551346176,
    "threads_max": 11,
    "used_memory_max": 63,
    "total_memory": 256,
    "memory_utilization": 24,
    "cpu_user_time": 6640,
    "cpu_system_time": 50,
    "cpu_total_time": 6690,
    "fd_use": 416,
    "fd_max": 32642,
    "tx_bytes": 4434,
    "rx_bytes": 6911,
    "timeout": true,
    "shutdown_reason": "Timeout",
    "total_network": 11345,
    "agent_version": "1.0.72.0",
    "agent_memory_avg": 10,
    "agent_memory_max": 10
}
```


Usa Contributor Insights per analizzare dati ad alta cardinalità

Puoi usare Contributor Insights per analizzare i dati di log e creare serie temporali che visualizzino i dati dei collaboratori. Puoi visualizzare i parametri relative ai primi N collaboratori, al numero totale di collaboratori univoci e al loro utilizzo. Questo ti permette di individuare gli interlocutori principali e comprendere chi o cosa sta influenzando le prestazioni del sistema. Ad esempio, puoi trovare host danneggiati, identificare gli utenti di rete più intensivi o individuare gli URL che generano il maggior numero di errori.

Puoi creare le tue regole partendo da zero e, quando le utilizzi, puoi anche utilizzare le regole di AWS Management Console esempio che hai creato. AWS Le regole definiscono i campi di log che desideri utilizzare per definire i collaboratori, ad esempio `IpAddress`. Puoi inoltre filtrare i dati di log per individuare e analizzare il comportamento di singoli collaboratori.

CloudWatch fornisce inoltre regole integrate che è possibile utilizzare per analizzare le metriche di altri AWS servizi.

Tutte le regole analizzano i dati in entrata in tempo reale.

Se hai effettuato l'accesso a un account configurato come account di monitoraggio nell'osservabilità CloudWatch tra account, puoi creare regole di Contributor Insights in quell'account di monitoraggio che analizzano i gruppi di log negli account di origine e nell'account di monitoraggio. Puoi anche creare una singola regola che analizzi i gruppi di log in più account. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

> **Note**
>
> Quando usi Contributor Insights vengono addebitati i costi per ogni occorrenza di un evento di log che corrisponde a una regola. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

Argomenti

Creazione di una regola di Approfondimenti sulle contribuzioni

Puoi creare regole per analizzare i dati di log. Possono essere valutati eventuali log in formato JSON o CLF (Common Log Format). Ciò include i log personalizzati che seguono uno di questi formati e i log di AWS servizi come i log di flusso di Amazon VPC, i log delle query DNS di Amazon Route 53, i log dei container Amazon ECS e i log di Amazon, Amazon RDS e API Gateway AWS CloudTrail. SageMaker AWS AppSync

In una regola, quando specifichi i nomi dei campi o i valori, tutte le corrispondenze fanno distinzione tra maiuscole e minuscole.

Puoi utilizzare le regole di esempio incorporate quando crei una regola oppure puoi creare una regola personalizzata da zero. Contributor Insights include regole di esempio per i seguenti tipi di log:

- Log di Amazon API Gateway
- Log di query DNS pubblici di Amazon Route 53
- Log di query di Amazon Route 53 Resolver
- CloudWatch Registri di Container Insights
- Log di flusso VPC

Se hai effettuato l'accesso a un account configurato come account di monitoraggio in modalità osservabile CloudWatch tra più account, puoi creare regole di Contributor Insights per i gruppi di log negli account di origine collegati a questo account di monitoraggio, oltre a creare regole per i gruppi di log nell'account di monitoraggio. Puoi anche configurare una singola regola per monitorare i gruppi di log in diversi account. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

> **Important**
>
> Quando concedi l'`cloudwatch:PutInsightRule`autorizzazione a un utente, per impostazione predefinita quell'utente può creare una regola che valuta qualsiasi gruppo di log in Logs. CloudWatch È possibile aggiungere condizioni di policy IAM che limitano queste autorizzazioni affinché un utente includa ed escluda gruppi di log specifici. Per ulteriori informazioni, consulta la pagina Utilizzo delle chiavi di condizione per limitare l'accesso degli utenti di Contributor Insights ai gruppi di log.

Per creare una regola usando una regola di esempio incorporata

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Insights (Approfondimenti), quindi Contributor Insights (Approfondimenti sulle contribuzioni).
3. Scegli Crea regola.
4. Per Select log group(s) (Seleziona uno o più i gruppi di log), seleziona i gruppi di log che devono essere monitorati dalla regola. È possibile selezionare fino a 20 gruppi di log. Se hai effettuato l'accesso a un account di monitoraggio configurato per l'osservabilità CloudWatch tra account diversi, puoi selezionare i gruppi di log negli account di origine e puoi anche impostare un'unica regola per analizzare i gruppi di log in account diversi.

- (Facoltativo) Per selezionare tutti i gruppi di log che hanno nomi che iniziano con una stringa specifica, scegli dal menu a discesa Seleziona per corrispondenza prefisso, quindi inserire il prefisso. Se si tratta di un account di monitoraggio, puoi selezionare facoltativamente gli account in cui eseguire la ricerca. In caso contrario, vengono selezionati tutti gli account.

> **Note**
>
> Vengono addebitati i costi per ogni evento di log corrispondente a una regola. Se si sceglie dal menu a discesa Seleziona per corrispondenza prefisso, tenere presente a quanti gruppi di log può corrispondere il prefisso. Se si esegue una ricerca in più gruppi di log di quelli desiderati, è possibile che vengano addebitati costi imprevisti. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

5. In Rule type (Tipo di regola), scegli Sample rule (Regola di esempio). Quindi scegli Select sample rule (Seleziona regola di esempio) e seleziona la regola.
6. La regola di esempio ha compilato i campi Log format (Formato dei log), Contribution (Contributo), Filters (Filtri) e Aggregate on (Aggrega su). È possibile modificare questi valori, se necessario.
7. Seleziona Successivo.
8. In Rule name (Nome regola) immetti un nome. I caratteri validi sono A-Z, a-z, 0-9, trattino, trattino basso e punto.
9. Scegli se crei la regola in uno stato disabilitato o abilitato. Se si sceglie di abilitarla, l'analisi dei dati viene avviata immediatamente dalla regola. Vengono addebitati i costi quando si eseguono regole abilitate. Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

   Contributor Insights analizza nuovi eventi di log solo dopo che una regola è stata creata. Una regola non può elaborare gli eventi di log precedentemente elaborati da CloudWatch Logs.
10. (Facoltativo) In Tags (Tag), aggiungi una o più coppie chiave-valore come tag per questa regola. I tag possono aiutarti a identificare e organizzare le tue AWS risorse e a tenere traccia dei costi. AWS Per ulteriori informazioni, consulta Taggare le tue risorse Amazon CloudWatch .
11. Scegli Create (Crea) .

Per creare una regola da zero

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione, scegli Contributor Insights.
3. Scegli Crea regola.
4. Per Select log group(s) (Seleziona uno o più i gruppi di log), seleziona i gruppi di log che devono essere monitorati dalla regola. È possibile selezionare fino a 20 gruppi di log. Se hai effettuato l'accesso a un account di monitoraggio configurato per l'osservabilità CloudWatch tra account diversi, puoi selezionare i gruppi di log negli account di origine e puoi anche impostare un'unica regola per analizzare i gruppi di log in account diversi.

   - (Facoltativo) Per selezionare tutti i gruppi di log che hanno nomi che iniziano con una stringa specifica, scegli dal menu a discesa Seleziona per corrispondenza prefisso, quindi inserire il prefisso.

   > **Note**
   >
   > Vengono addebitati i costi per ogni evento di log corrispondente a una regola. Se si sceglie dal menu a discesa Seleziona per corrispondenza prefisso, tenere presente a quanti gruppi di log può corrispondere il prefisso. Se si esegue una ricerca in più gruppi di log di quelli desiderati, è possibile che vengano addebitati costi imprevisti. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

5. In Rule type (Tipo di regola), scegli Custom rule (Regola personalizzata).
6. Per Log format (Formato log), scegli JSON o CLF.
7. Puoi terminare la creazione della regola utilizzando la procedura guidata oppure scegliendo la scheda Syntax (Sintassi) e specificando manualmente la sintassi della regola.

   Per continuare a utilizzare la procedura guidata, procedere nel modo seguente:

   a. In Contribution (Collaborazione), Key (Chiave), immetti un tipo di collaboratore per il quale creare un report. Nel report verranno visualizzati i primi N valori per questo tipo di collaboratore.

      Le voci valide sono qualsiasi campo di log con valori. Esempi includono **`requestId`**, **`sourceIPaddress`** e **`containerID`**.

      Per informazioni sulla ricerca dei nomi dei campi di log per i log in un determinato gruppo di log, consulta Ricerca dei campi di log.

Le chiavi più grandi di 1 KB vengono troncate a 1 KB.

b. (Facoltativo) Scegli Add new key (Aggiungi nuova chiave) per aggiungere altre chiavi. È possibile includere fino a quattro chiavi in una regola. Se si immettono più chiavi, i collaboratori nel report vengono definiti da combinazioni di valori univoche delle chiavi. Ad esempio, se si specificano tre chiavi, ogni combinazione univoca di valori per le tre chiavi viene conteggiata come un collaboratore univoco.

c. (Facoltativo) Se si desidera aggiungere un filtro che restringa l'ambito dei risultati, scegli Add filter (Aggiungi filtro). Per Match (Corrispondenza), inserisci il campo di log in base al quale desideri filtrare. Per Condition (Condizione), scegli un operatore di confronto e immetti un valore per il quale desideri filtrare questo campo.

   È possibile aggiungere fino a quattro filtri in una regola. Più filtri vengono uniti mediante logica AND, quindi vengono valutati solo gli eventi di log che corrispondono a tutti i filtri.

   > **Note**
   >
   > Matrici che seguono gli operatori di confronto, ad·esempio `In`, `NotIn` o `StartsWith`, possono includere fino a dieci valori di stringa. Per ulteriori informazioni sulla sintassi delle regole di Contributor Insights, consulta Sintassi delle regole Contributor Insights.

d. Per Aggregate on (Aggrega su), scegli Count o Sum. Se si sceglie Count, la classificazione contributori sarà basata sul numero di occorrenze. Se si sceglie Sum, la classificazione sarà basata sulla somma aggregata dei valori del campo specificato per Contribution (Contributo), Value (Valore).

8. Per immettere la regola come un oggetto JSON anziché utilizzando la procedura guidata, procedere nel modo seguente:

   a. Scegli la scheda Syntax (Sintassi) .

   b. In Rule body (Corpo della regola), immetti l'oggetto JSON per la regola. Per informazioni sulla sintassi della regola, consulta Sintassi delle regole Contributor Insights.

9. Seleziona Successivo.

10. In Rule name (Nome regola) immetti un nome. I caratteri validi sono A-Z, a-z, 0-9, "-", "_' e ".".

11. Scegli se crei la regola in uno stato disabilitato o abilitato. Se si sceglie di abilitarla, l'analisi dei dati viene avviata immediatamente dalla regola. Vengono addebitati i costi quando si eseguono regole abilitate. Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

Contributor Insights analizza nuovi eventi di log solo dopo che una regola è stata creata. Una regola non può elaborare gli eventi di log precedentemente elaborati da CloudWatch Logs.

12. (Facoltativo) In Tags (Tag), aggiungi una o più coppie chiave-valore come tag per questa regola. I tag possono aiutarti a identificare e organizzare le tue AWS risorse e a tenere traccia dei costi. AWS Per ulteriori informazioni, consulta la pagina Taggare le tue risorse Amazon CloudWatch .
13. Seleziona Next (Successivo).
14. Conferma le impostazioni che hai inserito e scegli Create rule (Crea regola).

Le regole create possono essere disabilitate, abilitate o eliminate.

Per abilitare, disabilitare o eliminare una regola in Contributor Insights

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Contributor Insights.
3. Nell'elenco di regole, seleziona la casella di controllo accanto a una singola regola.

   Le regole integrate vengono create dai AWS servizi e non possono essere modificate, disabilitate o eliminate.
4. Scegli Actions (Operazioni) e seleziona l'opzione desiderata.

Ricerca di campi di log

Quando si crea una regola, è necessario conoscere i nomi dei campi nelle voci di log in un gruppo di log.

Per trovare i campi di log in un gruppo di log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, in Logs, (Log), scegli Insights.
3. Sopra l'editor della query, seleziona uno o più gruppi di log su cui eseguire query.

   Quando si seleziona un gruppo di log, CloudWatch Logs Insights rileva automaticamente i campi nei dati del gruppo di log e li visualizza nel riquadro destro in Campi rilevati.

Sintassi delle regole Contributor Insights

In questa sezione viene descritta la sintassi delle regole Contributor Insights. Utilizza questa sintassi solo durante la creazione di una regola immettendo un blocco JSON. Se utilizzi la procedura guidata per creare una regola, non è necessario conoscere la sintassi. Per ulteriori informazioni sulla creazione di regole utilizzando la procedura guidata, consulta Creazione di una regola di Approfondimenti sulle contribuzioni.

Tutta la corrispondenza di regole per registrare i nomi e i valori dei campi degli eventi di log rileva la distinzione tra maiuscole e minuscole.

Nell'esempio seguente viene illustrata la sintassi per log JSON.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*",
        "Log-group-name2"
    ],
    "LogFormat": "JSON",
    "Contribution": {
        "Keys": [
            "$.ip"
        ],
        "ValueOf": "$.requestBytes",
        "Filters": [
            {
                "Match": "$.httpMethod",
                "In": [
                    "PUT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Campi nelle regole Contributor Insights

Schema

Il valore di `Schema` per una regola che analizza i dati di CloudWatch Logs deve sempre essere `{"Name": "CloudWatchLogRule", "Version": 1}`

LogGroupNames

Una matrice di stringhe. Per ogni elemento nella matrice, puoi utilizzare facoltativamente `*` al termine di una stringa per includere tutti i gruppi di log con i nomi che iniziano con tale prefisso.

Fai attenzione all'utilizzo di caratteri jolly con nomi di gruppi di log. Vengono addebitati i costi per ogni evento di log corrispondente a una regola. Se involontariamente si esegue una ricerca in più gruppi di log di quelli previsti, è possibile che vengano addebitati costi imprevisti. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

LogGroupARN

Se si crea questa regola in un CloudWatch account di monitoraggio dell'osservabilità tra più account, è possibile utilizzarla `LogGroupARNs` per specificare i gruppi di log negli account di origine collegati all'account di monitoraggio e per specificare i gruppi di log nell'account di monitoraggio stesso. Nella regola puoi specificare `LogGroupNames` o `LogGroupARNs`, ma non entrambi.

`LogGroupARNs` è una matrice di stringhe. Per ogni elemento della matrice, puoi utilizzare facoltativamente `*` come un carattere jolly in determinate situazioni. Ad esempio, puoi indicare `arn:aws:logs:us-west-1:*:log-group/MyLogGroupName2` per specificare i gruppi di log denominati `MyLogGroupName2` in tutti gli account di origine e nell'account di monitoraggio, nella regione Stati Uniti occidentali (California settentrionale). Puoi inoltre indicare `arn:aws:logs:us-west-1:111122223333:log-group/GroupNamePrefix*` per specificare tutti i gruppi di log nella regione Stati Uniti occidentali (California settentrionale) al numero 111122223333 i cui nomi iniziano con `GroupNamePrefix`.

Non è possibile specificare un ID AWS account parziale come prefisso con una wild card.

Fai attenzione all'utilizzo di caratteri jolly con gli ARN dei gruppi di log. Vengono addebitati i costi per ogni evento di log corrispondente a una regola. Se involontariamente si esegue una ricerca in più gruppi di log di quelli previsti, è possibile che vengano addebitati costi imprevisti. Per ulteriori informazioni, consulta la pagina CloudWatch dei prezzi di Amazon.

LogFormat

I valori validi sono `JSON` e `CLF`.

Contributo

Questo oggetto include una matrice `Keys` con un massimo di quattro membri, facoltativamente un singolo `ValueOf` e facoltativamente una matrice di un massimo di quattro `Filters`.

Chiavi

Una matrice di un massimo di quattro campi di log che vengono utilizzati come dimensioni per classificare i collaboratori. Se si immettono più chiavi, ogni combinazione univoca di valori per le chiavi viene conteggiata come un collaboratore univoco. I campi devono essere specificati utilizzando la notazione formato proprietà JSON.

ValueOf

(Facoltativo) Specifica questa opzione solo durante la definizione di `Sum` come valore di `AggregateOn`. `ValueOf` specifica un campo di log con valori numerici. In questo tipo di regola, i collaboratori vengono classificati in base alla somma del valore di questo campo, anziché al numero di occorrenze nelle voci di log. Ad esempio, Se si desidera ordinare i collaboratori in base al `BytesSent` totale su un periodo, devi impostare `ValueOf` su `BytesSent` e specificare `Sum` per `AggregateOn`.

Filtri

(Facoltativo) Specifica una matrice di massimo quattro filtri per limitare gli eventi di log inclusi nel report. Se si specificano più filtri, questi vengono valutati da Contributor Insights con un operatore AND logico. Puoi utilizzare questa opzione per filtrare gli eventi di log irrilevanti nella ricerca oppure per selezionare un singolo collaboratore per analizzarne il comportamento.

Ogni membro nella matrice deve includere un campo `Match` e un campo che indica il tipo di operatore corrispondente da utilizzare.

Il campo `Match` specifica un campo di log da valutare nel filtro. Il campo di log viene specificato utilizzando la notazione del formato proprietà JSON.

Il campo operatore corrispondente deve essere uno dei seguenti: `In`, `NotIn`, `StartsWith`, `GreaterThan`, `LessThan`, `EqualTo`, `NotEqualTo` o `IsPresent`. Se il campo operatore è `In`, `NotIn`, o `StartsWith`, è seguito da una matrice di valori stringa da controllare. Contributor Insights valuta la matrice di valori stringa con un operatore OR. La matrice può includere fino a 10 valori stringa.

Se il campo operatore è `GreaterThan`, `LessThan`, `EqualTo`, o `NotEqualTo`, è seguito da un singolo valore numerico da confrontare.

Se il campo operatore è `IsPresent`, è seguito da `true` o `false`. Questo operatore corrisponde agli eventi di log a seconda che il campo di log specificato sia presente nell'evento di log. `isPresent` funziona solo con i valori nel nodo foglia delle proprietà JSON. Ad esempio, un filtro che ricerca corrispondenze per `c-count` non valuta un evento di log con un valore pari a `details.c-count.c1`.

Vedi i quattro esempi di filtro di seguito:

```
{"Match": "$.httpMethod", "In": [ "PUT", ] }
{"Match": "$.StatusCode", "EqualTo": 200 }
{"Match": "$.BytesReceived", "GreaterThan": 10000}
{"Match": "$.eventSource", "StartsWith": [ "ec2", "ecs" ] }
```

AggregateOn

I valori validi sono `Count` e `Sum`. Specifica se aggregare il report in base a un conteggio di occorrenze o a una somma dei valori del campo specificato nel campo `ValueOf`.

Notazione formato proprietà JSON

I campi `Keys`, `ValueOf` e `Match` seguono il formato della proprietà JSON con notazione punto, in cui $ rappresenta la radice dell'oggetto JSON. Questa è seguita da un punto e quindi da una stringa alfanumerica con il nome della proprietà secondaria. Sono supportati più livelli di proprietà.

Il primo carattere della stringa può essere solo A-Z o a-z. I caratteri seguenti della stringa possono essere A-Z, a-z o 0-9.

L'elenco seguente illustra esempi di formato validi della proprietà JSON

```
$.userAgent
$.endpoints[0]
$.users[1].name
$.requestParameters.instanceId
```

Campo aggiuntivo nelle regole per log CLF

Gli eventi di log CLF (Common Log Format) non hanno nomi per i campi come per JSON. Per fornire i campi da utilizzare per le regole Contributor Insights, un evento di log CLF può essere gestito come

matrice con un indice che inizia da 1. Puoi specificare il primo campo come **"1"** e il secondo campo come **"2"** e così via.

Per semplificare la lettura di una regola per un log CLF, puoi utilizzare `Fields`. Ciò consente di fornire un alias di denominazione per le posizioni dei campi CLF. Ad esempio, puoi specificare che la posizione "4" è un indirizzo IP. Una volta specificato, è possibile utilizzare `IpAddress` come proprietà in `Keys`, `ValueOf` e `Filters` nella regola.

Di seguito è riportato un esempio di regola per un log CLF che utilizza il campo `Fields`.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "IpAddress",
        "7": "StatusCode"
    },
    "Contribution": {
        "Keys": [
            "IpAddress"
        ],
        "Filters": [
            {
                "Match": "StatusCode",
                "EqualTo": 200
            }
        ]
    },
    "AggregateOn": "Count"
}
```

Esempi di regole Contributor Insights

Questa sezione contiene esempi che illustrano i casi d'uso per le regole Contributor Insights.

Registri di flusso VPC: trasferimenti di byte per indirizzo IP di origine e destinazione

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "srcaddr",
        "5": "dstaddr",
        "10": "bytes"
    },
    "Contribution": {
        "Keys": [
            "srcaddr",
            "dstaddr"
        ],
        "ValueOf": "bytes",
        "Filters": []
    },
    "AggregateOn": "Sum"
}
```

Registri di flusso VPC: massimo numero di richieste HTTPS

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "5": "destination address",
        "7": "destination port",
        "9": "packet count"
    },
    "Contribution": {
        "Keys": [
```

```
            "destination address"
        ],
        "ValueOf": "packet count",
        "Filters": [
            {
                "Match": "destination port",
                "EqualTo": 443
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Registri di flusso VPC: connessioni TCP rifiutate

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
```

```
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Risposte NXDomain di Route 53 per indirizzo di origine

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.rcode",
                "StartsWith": [
                    "NXDOMAIN"
                ]
            }
        ],
        "Keys": [
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Query Route 53 Resolver per nome di dominio

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
```

```
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_name"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Query Route 53 Resolver per tipo di query e indirizzo di origine

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_type",
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Visualizzazione di report Contributor Insights

Per visualizzare i grafici dei dati di report e una classifica dei collaboratori individuati dalle regole, procedi come indicato di seguito.

Per visualizzare report di regole

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, scegli Contributor Insights.

3. Nell'elenco di regole, scegli il nome di una regola.

   Il grafico visualizza i risultati della regola nelle ultime tre ore. La tabella sotto il grafico mostra i primi 10 collaboratori.

4. Per modificare il numero di collaboratori mostrati nella tabella, scegli Top 10 contributors (Primi 10 collaboratori) nella parte superiore del grafico.

5. Per filtrare il grafico in modo da visualizzare solo i risultati di un singolo collaboratore, scegli tale collaboratore nella legenda della tabella. Per visualizzare nuovamente tutti i collaboratori, scegli lo stesso collaboratore nella legenda.

6. Per modificare l'intervallo di tempo mostrato nel report, scegli 15m, 30m, 1h, 2h, 3h o custom (personalizzato) nella parte superiore del grafico.

   L'intervallo di tempo massimo per il report è di 24 ore, ma è possibile scegliere una finestra di 24 ore che si è verificata fino a 15 giorni fa. Per scegliere una finestra temporale nel passato, seleziona custom (personalizzato), absolute (assoluto), quindi specifichi la finestra temporale.

7. Per modificare la durata del periodo di tempo utilizzato per l'aggregazione e la classificazione dei collaboratori, scegli period (periodo) nella parte superiore del grafico. La visualizzazione di un periodo di tempo più lungo mostra generalmente un report più fluido con pochi picchi. Scegliendo un periodo di tempo più breve, è più probabile che vengano visualizzati picchi.

8. Per aggiungere questo grafico a un CloudWatch pannello di controllo, scegli Aggiungi al pannello di controllo.

9. Per aprire la finestra di interrogazione di CloudWatch Logs Insights, con i gruppi di log di questo rapporto già caricati nella casella di interrogazione, scegli Visualizza registri.

10. Per esportare i dati di report negli Appunti o in un file CSV, scegli Export (Esporta).

Rappresentazione grafica dei parametri generati dalle regole

Contributor Insights fornisce una funzione matematica dei parametri, `INSIGHT_RULE_METRIC`. È possibile utilizzare questa funzione per aggiungere dati da un report di Contributor Insights a un grafico nella scheda Metriche della console. CloudWatch Puoi anche impostare un allarme basato su questa funzione matematica. Per ulteriori informazioni sulle funzioni matematiche dei parametri, consulta Utilizzare la matematica dei parametri

Per utilizzare questa funzione matematica dei parametri, devi aver effettuato l'accesso a un account che dispone delle autorizzazioni `cloudwatch:GetMetricData` e `cloudwatch:GetInsightRuleReport`.

La sintassi è `INSIGHT_RULE_METRIC(`*`ruleName`*`, `*`metricName`*`)`. *`ruleName`* è il nome di una regola Contributor Insights. *`metricName`* è uno dei valori nell'elenco seguente. Il valore di *`metricName`* determina il tipo di dati restituito dalla funzione matematica.

- `UniqueContributors`: il numero di fattori univoci per ogni punto dati.
- `MaxContributorValue`: il valore del fattore principale per ogni punto dati. L'identità del collaboratore può cambiare per ogni punto dati nel grafico.

  Se questa regola viene aggregata per `Count`, il collaboratore principale per ogni punto dati è quello con il maggior numero di occorrenze in tale periodo. Se la regola viene aggregata per `Sum`, il collaboratore principale è quello con la somma maggiore nel campo di log specificato dal `Value` della regola, durante tale periodo.
- `SampleCount`: il numero di punti dati corrispondenti alla regola.
- `Sum`: la somma dei valori di tutti i fattori durante il periodo di tempo rappresentato da tale punto dati.
- `Minimum`: il valore minimo ottenuto da un'unica osservazione durante il periodo di tempo rappresentato da tale punto dati.
- `Maximum`: il valore massimo ottenuto da un'unica osservazione durante il periodo di tempo rappresentato da tale punto dati.
- `Average`: il valore medio di tutti i fattori durante il periodo di tempo rappresentato da tale punto dati.

Impostazione di un allarme sui dati dei parametri Contributor Insights

Usando la funzione `INSIGHT_RULE_METRIC`, puoi impostare allarmi sui parametri generati da Contributor Insights. Ad esempio, puoi creare un allarme in base alla percentuale di connessioni TCP (transmission control protocol) che sono state rifiutate. Per iniziare a utilizzare questo tipo di allarme, puoi creare regole come quelle mostrate nei due esempi seguenti:

Regola di esempio: "" RejectedConnectionsRule

```
{
    "Schema": {
```

```
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Regola di esempio: "TotalConnectionsRule"

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
```

```
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            }
    "AggregateOn": "Sum"
}
```

Dopo aver creato le regole, puoi selezionare la scheda Metriche nella CloudWatch Console, dove puoi utilizzare i seguenti esempi di espressioni matematiche delle metriche per rappresentare graficamente i dati riportati da Contributor Insights:

Esempio: espressioni matematiche dei parametri

```
e1 INSIGHT_RULE_METRIC("RejectedConnectionsRule", "Sum")
e2 INSIGHT_RULE_METRIC("TotalConnectionsRule", "Sum")
e3 (e1/e2)*100
```

Nell'esempio, l'espressione matematica dei parametri `e3` restituisce tutte le connessioni TCP rifiutate. Se si desidera ricevere una notifica quando il 20% delle connessioni TCP viene rifiutato, puoi modificare l'espressione cambiando la soglia da `100` a `20`.

> **Note**
>
> Puoi impostare un allarme su un parametro che stai monitorando dalla sezione Metrics (Parametri). Nella scheda Graphed metrics (Parametri nel grafico), puoi selezionare l'icona Create alarm (Crea allarme) nella colonna Actions (Operazioni). L'icona Create alarm (Crea allarme) ha l'aspetto di una campanella.

Per ulteriori informazioni sulla rappresentazione grafica di parametri e sull'utilizzo di funzioni matematiche dei parametri, consulta la seguente sezione: Aggiungere un'espressione matematica a un grafico CloudWatch .

Utilizzo di regole integrate di Contributor Insights

Puoi utilizzare le regole integrate di Contributor Insights per analizzare le metriche di altri servizi. AWS I seguenti servizi supportano le regole integrate:

- Contributor Insights per Amazon DynamoDBnella Guida per gli sviluppatori di Amazon DynamoDB.
- Utilizzo di regole integrate di Contributor Insights nella Guida di AWS PrivateLink .

Informazioni approfondite sulle CloudWatch applicazioni Amazon

Amazon CloudWatch Application Insights facilita l'osservabilità delle tue applicazioni e delle risorse sottostanti AWS . È utile per configurare i migliori monitor per le risorse dell'applicazione, per analizzare i dati in modo continuo per rilevare problemi con le applicazioni. Application Insights, che si basa su SageMakere altre AWS tecnologie, fornisce dashboard automatizzate che mostrano potenziali problemi con le applicazioni monitorate, che ti aiutano a isolare rapidamente i problemi in corso con le applicazioni e l'infrastruttura. La visibilità migliorata sull'integrità delle applicazioni fornita da Application Insights aiuta a ridurre il tempo medio di riparazione (MTTR) per risolvere i problemi dell'applicazione.

Quando aggiungi le tue applicazioni ad Amazon CloudWatch Application Insights, analizza le risorse nelle applicazioni e consiglia e configura i parametri e i log in per i componenti dell'applicazione. CloudWatch I componenti applicativi di esempio possono includere database back-end di SQL Server e livelli Microsoft IIS/Web. Application Insights analizza i modelli di parametri utilizzando i dati storici per rilevare anomalie e rilevare in modo continuo errori ed eccezioni dai log dell'applicazione, del sistema operativo e dell'infrastruttura. Queste osservazioni vengono correlate utilizzando una combinazione di algoritmi di classificazione e regole integrate. Quindi vengono creati automaticamente pannelli di controllo che mostrano osservazioni pertinenti e informazioni sulla gravità dei problemi per assegnare una priorità alle operazioni. Per problemi comuni in stack di applicazioni .NET e SQL, ad esempio latenza dell'applicazione, backup non riusciti di SQL Server, perdite di memoria, richieste HTTP di grandi dimensioni e operazioni di I/O interrotte, offre ulteriori approfondimenti che puntano a una possibile causa principale e alle fasi per la risoluzione. L'integrazione integrata con AWS SSM OpsCenter consente di risolvere i problemi eseguendo il documento Systems Manager Automation pertinente.

Sections

Che cos'è Amazon CloudWatch Application Insights?

CloudWatch Application Insights ti aiuta a monitorare le tue applicazioni che utilizzano istanze Amazon EC2 insieme ad altre risorse applicative. Identifica e configura i log dei parametri chiave tra risorse dell'applicazione e stack tecnologico (ad esempio, il database, Microsoft SQL Server, i server (IIS) web e di applicazione, il sistema operativo, i sistemi di bilanciamento del carico e le code). Controlla in modo continuo i parametri e i log per rilevare e correlare anomalie ed errori. Quando vengono rilevati errori e anomalie, Application Insights genera CloudWatch eventi che puoi utilizzare per impostare notifiche o intraprendere azioni. Per assistere nella risoluzione dei problemi, crea pannelli di controllo automatizzati per i problemi rilevati, che includono anomalie parametri ed errori di log correlati, insieme ad altri approfondimenti per indirizzarti verso la causa principale potenziale. I pannelli di controllo automatizzati consentono di eseguire operazioni di correzione per mantenere le applicazioni integre e prevenire l'impatto sugli utenti finali dell'applicazione. Inoltre, consente OpsItems di risolvere i problemi utilizzando AWS SSM. OpsCenter

È possibile configurare contatori importanti, come Mirrored Write Transaction/sec, Recovery Queue Length e Transaction Delay, nonché i registri degli eventi di Windows. CloudWatch Quando si verifica un evento o un problema di failover con il carico di lavoro SQL HA, ad esempio, un accesso limitato per interrogare un database di destinazione, Application Insights fornisce informazioni automatiche. CloudWatch

CloudWatch Application Insights si integra con AWS Launch Wizardper fornire un'esperienza di configurazione di monitoraggio con un solo clic per la distribuzione di carichi di lavoro SQL Server HA su. AWS Quando si seleziona l'opzione per configurare il monitoraggio e gli approfondimenti con Application Insights nella console Launch Wizard, CloudWatch Application Insights imposta automaticamente le metriche, i log e gli allarmi pertinenti e inizia a monitorare i carichi di lavoro CloudWatch appena distribuiti. È possibile visualizzare gli approfondimenti automatici e i problemi rilevati, oltre allo stato dei carichi di lavoro SQL Server HA, sulla console. CloudWatch

Indice

Funzionalità

Application Insights offre le caratteristiche seguenti.

Configurazione automatica di monitor per risorse dell'applicazione

CloudWatch Application Insights riduce il tempo necessario per configurare il monitoraggio delle applicazioni. A tale scopo, scansiona le risorse dell'applicazione, fornisce un elenco personalizzabile di parametri e log consigliati e li configura CloudWatch per fornire la visibilità necessaria sulle risorse dell'applicazione, come Amazon EC2 ed Elastic Load Balancers (ELB). Inoltre, consente di impostare allarmi dinamici con parametri monitorati. Gli allarmi vengono aggiornati automaticamente in base alle anomalie rilevate nelle ultime due settimane.

Rilevamento e notifica del problema

CloudWatch Application Insights rileva segnali di potenziali problemi con l'applicazione, come anomalie delle metriche ed errori di registro. Correla queste osservazioni per fare emergere i problemi potenziali con l'applicazione. Quindi genera CloudWatch eventi, che possono essere configurati per ricevere notifiche o intraprendere azioni. Questo elimina la necessità di creare singoli allarmi su parametri o errori di log.

Risoluzione dei problemi

CloudWatch Application Insights crea dashboard CloudWatch automatici per i problemi rilevati. I pannelli di controllo mostrano i dettagli del problema, incluse le anomalie parametri ed errori di log associati per semplificare la risoluzione dei problemi. Inoltre, forniscono approfondimenti aggiuntivi che puntano alle cause principali potenziali delle anomalie e degli errori.

Concetti

I seguenti concetti sono importanti per comprendere come Application Insights monitora l'applicazione.

Componente

Un raggruppamento automatico, autonomo o personalizzato di risorse simili che costituiscono un'applicazione. Si consiglia di raggruppare risorse simili in componenti personalizzati per migliorare il monitoraggio.

Osservazione

Un singolo evento (anomalia parametro, errore di log o eccezione) che viene rilevato con un'applicazione o risorsa dell'applicazione.

Problema

Problemi vengono rilevati correlando, classificando e raggruppando osservazioni correlate.

Per le definizioni di altri concetti chiave per CloudWatch Application Insights, consulta Amazon CloudWatch Concepts.

Prezzi

CloudWatch Application Insights imposta metriche e log consigliati per risorse applicative selezionate utilizzando CloudWatch metriche, registri ed eventi per le notifiche sui problemi rilevati. Queste funzionalità vengono addebitate sul tuo AWS account in base al prezzo. CloudWatch Per i problemi rilevati, gli SSM OpsItems vengono creati anche da Application Insights per avvisare l'utente dei problemi. Inoltre, Application Insights crea parametri SSM Parameter Store per configurare gli CloudWatch agenti sulle istanze. Le funzionalità di Amazon EC2 Systems Manager vengono addebitate in base ai prezzi di SSM. Non è previsto alcun addebito per la configurazione di assistenza, il monitoraggio di analisi dei dati o il rilevamento di problemi.

Costi di Application Insights CloudWatch

I costi per Amazon EC2 includono l'utilizzo delle seguenti funzionalità:

- CloudWatch Agente
  - CloudWatch Gruppi di log degli agenti
  - CloudWatch Metriche degli agenti
  - Gruppi di log Prometheus (per carichi di lavoro JMX)

I costi per tutte le risorse includono l'utilizzo delle seguenti funzionalità:

- CloudWatch allarmi (la maggior parte dei costi)
- SSM OpsItems (costo minimo)

Esempio di calcolo dei costi

I costi in questo esempio vengono considerati in base allo scenario seguente.

Hai creato un gruppo di risorse che include quanto segue:

- Un'istanza Amazon EC2 con SQL Server installato.
- Un volume Amazon EBS allegato.

Quando effettui l'onboarding di questo gruppo di risorse con CloudWatch Application Insights, viene rilevato il carico di lavoro di SQL Server installato sull'istanza Amazon EC2. CloudWatch Application Insights inizia a monitorare le seguenti metriche.

Per l'istanza di SQL Server verranno monitorati i seguenti parametri:

- CPUUtilization
- StatusCheckFailed
- Memory: byte impegnati della percentuale di memoria in uso
- Mbyte di memoria disponibili
- Byte totali al secondo dell'interfaccia di rete
- Uso in percentuale del file di paginazione
- Tempo disco in percentuale di PhysicalDisk
- Tempo processore in percentuale del processore
- SQLServer:Buffer Manager cache hit ratio
- SQLServer:Buffer Manager life expectancy

- SQLServer:General Statistics Processes blocked
- SQLServer:General Statistics User Connections
- SQLServer:Locks Number of Deadlocks/sec
- SQLServer:SQL Statistics Batch Requests/sec
- Lunghezza coda processore di sistema

Per i volumi collegati all'istanza di SQL Server verranno monitorati i seguenti parametri:

- VolumeReadBytes
- VolumeWriteBytes
- VolumeReadOps
- VolumeWriteOps
- VolumeTotalReadTime
- VolumeTotalWriteTime
- VolumeIdleTime
- VolumeQueueLength
- VolumeThroughputPercentage
- VolumeConsumedReadWriteOps
- BurstBalance

In questo scenario, i costi vengono calcolati in base alla pagina dei CloudWatch prezzi e alla pagina dei prezzi SSM:

- Parametri personalizzati

  In questo scenario, 13 delle metriche precedenti vengono assegnate all' CloudWatch utilizzo dell'agente. CloudWatch Questi parametri vengono trattati come metriche personalizzate. Il costo per ogni parametro personalizzato è di 0,30 USD al mese. Il costo totale di queste metriche personalizzati è di 13*0,30 USD = 3,90 USD al mese.
- Allarmi

  In questo scenario, CloudWatch Application Insights monitora 26 metriche in totale, il che crea 26 allarmi. Il costo per ogni allarme è di 0,10 USD al mese. Il costo totale degli allarmi è di 26 * 0,10 USD = 2,60 USD al mese.

- Log degli errori e importazione dei dati

  Il costo di un'importazione dei dati è di 0,05 USD per GB e lo storage per il log degli errori di SQL Server è di 0,03 USD per GB. Il costo totale per l'importazione dei dati e il log degli errori è di 0,05 USD/GB + 0,03 USD/GB= 0,08 USD/GB.
- Amazon EC2 Systems Manager OpsItems

  OpsItem Viene creato un SSM per ogni problema rilevato da CloudWatch Application Insights. Per un numero n di problemi nell'applicazione, il costo totale è di 0,00267 $* n al mese.

Servizi correlati

I seguenti servizi vengono utilizzati insieme ad CloudWatch Application Insights:

AWS Servizi correlati

- Amazon CloudWatch offre visibilità a livello di sistema sull'utilizzo delle risorse, sulle prestazioni delle applicazioni e sullo stato operativo. Raccoglie e tiene traccia delle metriche, invia notifiche di allarme, aggiorna automaticamente le risorse monitorate in base alle regole da te definite e ti consente di monitorare i tuoi parametri personalizzati. CloudWatch Application Insights viene avviato tramite, in CloudWatch particolare, all'interno dei dashboard operativi predefiniti. CloudWatch Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.
- CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. È possibile utilizzare Container Insights per monitorare le piattaforme Amazon ECS, Amazon Elastic Kubernetes Service e Kubernetes su Amazon EC2. Quando Application Insights è abilitato sulle console Container Insights o Application Insights, Application Insights visualizza i problemi rilevati nel pannello di controllo di Container Insights. Per ulteriori informazioni, consulta la pagina Container Insights .
- Amazon DynamoDB è un servizio di database NoSQL completamente gestito che consente di sollevarti dall'onere di gestire e ridimensionare un database distribuito e di non doverti più preoccupare di provisioning dell'hardware, installazione e configurazione, replica, applicazione di patch al software e dimensionamento del cluster. DynamoDB offre la crittografia dei dati inattivi, che permette di eliminare gli oneri operativi e la complessità previsti dalla protezione dei dati sensibili.
- Amazon EC2 offre capacità di elaborazione scalabile nel cloud. AWS Puoi utilizzare Amazon EC2 per avviare il numero di server virtuali necessari, configurare la sicurezza e i servizi di rete, nonché gestire lo storage. Puoi aumentare o ridurre le risorse per gestire le variazioni a livello di requisiti o

i picchi di popolarità, riducendo la necessità di elaborare previsioni relative al traffico. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon EC2 per istanze Linux o Guida per l'utente di Amazon EC2 per istanze Windows.

- Amazon Elastic Block Store (Amazon EBS) fornisce volumi di storage a livello di blocchi da utilizzare con le istanze Amazon EC2. Il comportamento dei volumi Amazon EBS è simile a quello dei dispositivi a blocchi non formattati e non elaborati. Puoi montare questi volumi come dispositivi sulle istanze. I volumi Amazon EBS collegati a un'istanza sono esposti come volumi di storage che persistono indipendentemente dalla durata dell'istanza stessa. Puoi creare un file system su questi volumi oppure impiegarli allo stesso modo di un dispositivo a blocchi (ad esempio, un disco rigido). Puoi modificare dinamicamente la configurazione di un volume collegato a un'istanza. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon EBS.
- Amazon EC2 Auto Scaling assicura di disporre del numero corretto di istanze EC2 disponibili per gestire il carico dell'applicazione. Per ulteriori informazioni, consulta Guida per l'utente di Dimensionamento automatico Amazon EC2.
- Elastic Load Balancing distribuisce le applicazioni o il traffico di rete in ingresso su più destinazioni, ad esempio istanze EC2, container e indirizzi IP, in più zone di disponibilità. Per ulteriori informazioni, consulta la Guida per l'utente di Elastic Load Balancing.
- IAM è un servizio web che ti aiuta a controllare in modo sicuro l'accesso alle AWS risorse per i tuoi utenti. Usa IAM per controllare chi può utilizzare AWS le tue risorse (autenticazione) e per controllare le risorse che possono utilizzare e come possono usarle (autorizzazione). Per ulteriori informazioni, consulta Authentication and Access Control for Amazon CloudWatch.
- AWS Lambdaconsente di creare applicazioni serverless composte da funzioni attivate da eventi e di distribuirle automaticamente utilizzando e. CodePipeline AWS CodeBuild Per ulteriori informazioni, consulta Applicazioni AWS Lambda.
- AWS Launch Wizard for SQL Server riduce il tempo necessario per implementare la soluzione SQL Server ad alta disponibilità nel cloud. Immetti i requisiti dell'applicazione, tra cui prestazioni, numero di nodi e connettività nella console di servizio, e AWS Launch Wizard identifica le AWS risorse giuste per distribuire ed eseguire l'applicazione SQL Server Always On.
- AWS I Resource Groups ti aiutano a organizzare le risorse che compongono la tua applicazione. Grazie a Resource Groups, puoi gestire e automatizzare le attività su un numero elevato di risorse simultaneamente. Solo un gruppo di risorse può essere registrato per una singola applicazione. Per ulteriori informazioni, consulta la Guida per l'utente di Resource Groups AWS .
- Amazon SQS offre una coda ospitata sicura, durevole e disponibile che consente di integrare e decuplicare i componenti e i sistemi software distribuiti. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon SQS.

- AWS Step Functionsè un compositore di funzioni senza server che consente di sequenziare una varietà di AWS servizi e risorse, incluse le AWS Lambda funzioni, in flussi di lavoro visivi strutturati. Per ulteriori informazioni, consulta la Guida per l'utente AWS Step Functions.
- AWS SSM OpsCenter aggrega e standardizza OpsItems tutti i servizi, fornendo al contempo dati di indagine contestuali su ciascuna OpsItem risorsa correlata e correlata. OpsItems OpsCenter fornisce inoltre documenti di Systems Manager Automation (runbook) che è possibile utilizzare per risolvere rapidamente i problemi. È possibile specificare dati ricercabili e personalizzati per ciascuno di essi. OpsItem È inoltre possibile visualizzare report di riepilogo generati automaticamente suddivisi per stato e origine. OpsItems Per ulteriori informazioni, consulta la Guida per l'utente AWS Systems Manager .
- Amazon API Gateway è un AWS servizio per la creazione, la pubblicazione, la manutenzione, il monitoraggio e la protezione di REST, HTTP e WebSocket API su qualsiasi scala. Gli sviluppatori di API possono creare API che accedono AWS o ad altri servizi Web, oltre ai dati archiviati nel cloud. AWS Per ulteriori informazioni, consulta la Guida per l'utente di Amazon API Gateway.

  > **Note**
  >
  > Application Insights supporta solo i protocolli REST API (v1 del servizio API Gateway).

- Amazon Elastic Container Service (Amazon ECS) è un servizio completamente gestito di orchestrazione dei container. Puoi utilizzare Amazon ECS per eseguire le applicazioni più sensibili e mission-critical. Per ulteriori informazioni, consulta la Guida per lo sviluppatore di Amazon Elastic Container.
- Amazon Elastic Kubernetes Service (Amazon EKS) è un servizio gestito che puoi usare per eseguire AWS Kubernetes senza dover installare, gestire e mantenere il tuo piano di controllo o i tuoi nodi Kubernetes. Kubernetes è un sistema open source per automatizzare l'implementazione, il dimensionamento e la gestione di applicazioni containerizzate. Per maggiori informazioni, consulta la Guida per l'utente di Amazon EKS.
- Kubernetes su Amazon EC2. Kubernetes è un software open source che consente di distribuire e gestire applicazioni containerizzate su larga scala. Kubernetes gestisce cluster di istanze di calcolo Amazon EC2 ed esegue i container su tali istanze con processi di distribuzione, manutenzione e dimensionamento. Con Kubernetes puoi eseguire qualsiasi tipo di applicazione containerizzata con lo stesso set di strumenti on-premise e nel cloud. Per ulteriori informazioni, consulta Documentazione di Kubernetes: Nozioni di base.
- Amazon FSx consente di avviare ed eseguire file system popolari completamente gestiti da AWS. Con Amazon FSx, puoi sfruttare i set di funzionalità e le prestazioni dei comuni file system open

source e con licenza commerciale per evitare attività amministrative dispendiose in termini di tempo. Per ulteriori informazioni, consulta la documentazione di Amazon FSx.

- Amazon Simple Notification Service (SNS) è un servizio di messaggistica completamente gestito per entrambe le comunicazioni. application-to-application application-to-person È possibile configurare Amazon SNS per il monitoraggio da parte di Application Insights. Quando Amazon SNS è configurato come risorsa per il monitoraggio, Application Insights tiene traccia delle metriche SNS per determinare il motivo per cui i messaggi SNS potrebbero riscontrare problemi o fallire.
- Amazon Elastic File System (Amazon EFS) è un file system NFS elastico completamente gestito da utilizzare con Cloud AWS servizi e risorse locali. È progettato per scalare fino a petabyte su richiesta senza interrompere le applicazioni. Si aumenta e si riduce automaticamente con l'aggiunta e la rimozione di file, il che elimina la necessità di fornire e gestire la capacità per soddisfare la crescita. Per ulteriori informazioni, consulta la documentazione di Amazon Elastic File System.

Servizi di terze parti correlati

- Per alcuni carichi di lavoro e applicazioni monitorati in Application Insights, l'esportatore Prometheus JMX viene installato AWS Systems Manager utilizzando Distributor in modo che Application Insights possa recuperare metriche specifiche di Java. CloudWatch Quando si sceglie di monitorare un'applicazione Java, Application Insights installa automaticamente Prometheus JMX Exporter.

Componenti dell'applicazione supportati

CloudWatch Application Insights analizza il gruppo di risorse per identificare i componenti dell'applicazione. I componenti possono essere autonomi, raggruppati automaticamente (ad esempio istanze in un gruppo con scalabilità automatica o dietro un sistema di bilanciamento del carico) o personalizzati (raggruppando insieme singole istanze Amazon EC2).

I seguenti componenti sono supportati da CloudWatch Application Insights:

AWS componenti

- Amazon EC2
- Amazon EBS
- Amazon RDS
- Elastic Load Balancing: Application Load Balancer e Classic Load Balancer (tutte le istanze di destinazione di questi sistemi di bilanciamento del carico vengono identificati e configurati).

- Gruppi di Auto Scaling di Amazon EC2: Auto AWS Scaling (i gruppi Auto Scaling sono configurati dinamicamente per tutte le istanze di destinazione; se l'applicazione è scalabile, Application Insights configura automaticamente le nuove istanze). CloudWatch I gruppi di Auto Scaling non sono supportati per i gruppi di risorse basati sullo CloudFormation stack.
- AWS Lambda
- Amazon Simple Queue Service (Amazon SQS)
- Tabella Amazon DynamoDB
- Parametri dei bucket Amazon S3
- AWS Step Functions
- Fasi REST API di Amazon API Gateway
- Amazon Elastic Container Service (Amazon ECS): cluster, servizio e attività
- Amazon Elastic Kubernetes Service (Amazon EKS): cluster
- Kubernetes su Amazon EC2: cluster Kubernetes in esecuzione su EC2
- Argomento Amazon SNS

Qualsiasi altra risorsa di tipo di componente non viene attualmente monitorata da Application Insights. CloudWatch Se un tipo di componente supportato non viene visualizzato nell'applicazione Application Insights, il componente potrebbe essere già registrato e gestito da un'altra applicazione di tua proprietà monitorata da Application Insights.

Stack tecnologici supportati

È possibile utilizzare CloudWatch Application Insights per monitorare le applicazioni in esecuzione su sistemi operativi Windows Server e Linux selezionando l'opzione del menu a discesa a livello di applicazione per una delle seguenti tecnologie:

- Front-end: server Web Microsoft Internet Information Services (IIS)
- Livello Worker:
  - .NET Framework
  - .NET Core
- Applicazioni:
  - Java
  - Implementazioni SAP NetWeaver standard, distribuite e ad alta disponibilità
- Active Directory

- SharePoint
- Database:
  - Microsoft SQL Server in esecuzione su Amazon RDS o Amazon EC2 (incluse le configurazioni di SQL Server High Availability, consulta Esempi di configurazione dei componenti).
  - MySQL in esecuzione su Amazon RDS, Amazon Aurora o Amazon EC2
  - PostgreSQL in esecuzione su Amazon RDS o Amazon EC2
  - Tabella Amazon DynamoDB
  - Oracle in esecuzione su Amazon RDS o Amazon EC2
  - Database SAP HANA su una singola istanza Amazon EC2 e più istanze EC2
  - Configurazione ad alta disponibilità del database SAP HANA Cross-AZ
  - Database SAP Sybase ASE su una singola istanza Amazon EC2
  - Configurazione ad alta disponibilità del database SAP Sybase ASE Cross-AZ

Se nessuno degli stack tecnologici elencati sopra si applica alle risorse dell'applicazione, è possibile monitorare lo stack delle applicazioni scegliendo Personalizzato dal menu a discesa Livello applicazione nella pagina Gestisci monitoraggio.

Come funziona Amazon CloudWatch Application Insights

Questa sezione contiene informazioni su come funziona CloudWatch Application Insights, tra cui:

- Monitoraggio delle applicazioni di Application Insights
- Conservazione dei dati
- Quote
- AWS Pacchetti Systems Manager (SSM) utilizzati da CloudWatch Application Insights
- AWS Documenti Systems Manager (SSM) utilizzati da CloudWatch Application Insights

Monitoraggio delle applicazioni di Application Insights

Application Insights monitora le applicazioni come segue.

Individuazione e configurazione dell'applicazione

La prima volta che un'applicazione viene aggiunta ad CloudWatch Application Insights, analizza i componenti dell'applicazione per consigliare metriche chiave, log e altre fonti di dati da monitorare per l'applicazione. Puoi quindi configurare l'applicazione in base a questi suggerimenti.

Pre-elaborazione dei dati

CloudWatch Application Insights analizza continuamente le fonti di dati monitorate tra le risorse dell'applicazione per scoprire anomalie metriche ed errori di registro (osservazioni).

Rilevamento intelligente del problema

Il motore di CloudWatch Application Insights rileva i problemi nell'applicazione correlando le osservazioni utilizzando algoritmi di classificazione e regole integrate. Per facilitare la risoluzione dei problemi, crea CloudWatch dashboard automatici, che includono informazioni contestuali sui problemi.

Avviso e operazione

Quando CloudWatch Application Insights rileva un problema con l'applicazione, genera CloudWatch eventi per avvisare l'utente del problema. Per ulteriori informazioni su come configurare tali eventi, consulta CloudWatch Eventi e notifiche di Application Insights per i problemi rilevati.

Scenario di esempio

Hai a disposizione un'applicazione ASP .NET che è supportata da un database SQL Server. Improvvisamente, il database inizia a presentare anomalie di funzionamento a causa di elevata pressione di memoria. Questo porta a un degrado delle prestazioni dell'applicazione e possibili errori HTTP 500 nei server Web e nel load balancer.

Con CloudWatch Application Insights e la sua analisi intelligente, puoi identificare il livello applicativo che causa il problema controllando la dashboard creata dinamicamente che mostra le metriche correlate e i frammenti di file di registro. In questo caso, il problema potrebbe essere a livello di database SQL.

Conservazione dei dati

CloudWatch Application Insights conserva i problemi per 55 giorni e le osservazioni per 60 giorni.

Quote

Per le quote predefinite per CloudWatch Application Insights, consulta gli endpoint e le quote di Amazon CloudWatch Application Insights. Salvo diversa indicazione, ogni quota è per regione. AWS Contatta AWS Support per richiedere un incremento della quota di servizio. Molti servizi contengono quote che non possono essere modificate. Per ulteriori informazioni sulle quote per un servizio specifico, consulta la documentazione relativa a tale servizio.

AWS Pacchetti Systems Manager (SSM) utilizzati da CloudWatch Application Insights

I pacchetti elencati in questa sezione vengono utilizzati da Application Insights e possono essere gestiti e distribuiti in modo indipendente con AWS Systems Manager Distributor. Per ulteriori informazioni su SSM Distributor, consulta AWS Systems Manager Distributor nella Guida per l'utente di AWS Systems Manager.

Pacchetti:

- AWSObservabilityExporter-JMXExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure
- AWSObservabilityExporter-HAClusterExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure
- AWSObservabilityExporter-SQLExporterInstallAndConfigure

`AWSObservabilityExporter-JMXExporterInstallAndConfigure`

È possibile recuperare i parametri Java specifici del carico di lavoro da Prometheus JMX exporter per Application Insights per configurare e monitorare gli allarmi. Nella console di Application Insights, nella scheda Gestisci monitoraggio, seleziona Applicazione JAVA dal menu a discesa Livello applicazione. Quindi in Configurazione di JAVA Prometheus Exporter, seleziona il Metodo di raccolta e il Numero porta JMX.

Per utilizzare AWS Systems Manager Distributor per impacchettare, installare e configurare il pacchetto di esportazione AWS Prometheus JMX fornito indipendentemente da Application Insights, completare i seguenti passaggi.

Prerequisiti per l'utilizzo del pacchetto SSM Prometheus JMX Exporter

- Versione SSM Agent 2.3.1550.0 o successiva installata
- La variabile di ambiente JAVA_HOME è impostata

Installazione e configurazione del paccetto **`AWSObservabilityExporter-JMXExporterInstallAndConfigure`**

I pacchetto `AWSObservabilityExporter-JMXExporterInstallAndConfigure` è un pacchetto Distributor SSM che è possibile utilizzare per installare e configurare Prometheus JMX Exporter.

Quando le metriche Java vengono inviate dall'esportatore Prometheus JMX, l'agente può essere configurato per recuperare le metriche per CloudWatch il servizio. CloudWatch

1. In base alle preferenze, prepara il file di configurazione YAML dell'esportatore Prometheus JMX che si trova nell'archivio Prometheus. GitHub Utilizzare le descrizioni di configurazione e opzioni di esempio come guida.
2. Copia il file di configurazione YAML di Prometheus JMX Exporter codificato come Base64 in un nuovo parametro SSM nell'Archivio parametri SSM.
3. Passare alla console SSM Distributor e aprire la scheda Proprietà di Amazon. AWSObservabilityExporterSeleziona ExporterInstallAndConfigure -JMX e scegli Installa una sola volta.
4. Aggiorna il parametro SSM creato nel primo passaggio sostituendo "Argomenti aggiuntivi" con quanto segue:

```
{
  "SSM_EXPORTER_CONFIGURATION": "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
  "SSM_EXPOSITION_PORT": "9404"
}
```

> **Note**
>
> La porta 9404 è la porta predefinita utilizzata per inviare i parametri Prometheus JMX. È possibile aggiornare questa porta.

Esempio: configura CloudWatch l'agente per recuperare le metriche Java

1. Installare il JMX exporter di Prometheus, come descritto nella procedura precedente. Verificare quindi che sia installato correttamente sull'istanza controllando lo stato della porta.

   Esempio di installazione riuscita sull'istanza di Windows

   ```
   PS C:\> curl http://localhost:9404 (http://localhost:9404/)
   StatusCode : 200
   StatusDescription : OK
   Content : # HELP jvm_info JVM version info
   ```

   Esempio di installazione riuscita sull'istanza di Linux

```
$ curl localhost:9404
# HELP jmx_config_reload_failure_total Number of times configuration have failed to
 be reloaded.
# TYPE jmx_config_reload_failure_total counter
jmx_config_reload_failure_total 0.0
```

2. Crea il file YAML di individuazione del servizio Prometheus. Il file di individuazione del servizio di esempio esegue queste operazioni:

   - Specifica la porta host di Prometheus JMX Exporter come `localhost: 9404`.
   - Allega etichette (`ApplicationComponentName`, `eInstanceId`) alle metriche, che possono essere impostate come dimensioni metriche. CloudWatch

```
$ cat prometheus_sd_jmx.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    Application: myApp
    ComponentName: arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/app/sampl-Appli-MMZW8E3GH4H2/aac36d7fea2a6e5b
    InstanceId: i-12345678901234567
```

3. Crea il file YAML di configurazione di Prometheus JMX Exporter. Il file di configurazione di esempio specifica quanto segue:

   - L'intervallo del processo di recupero dei parametri e periodo di timeout.
   - I processi di recupero dei parametri (noto anche come “scraping”) (`jmx` e `sap`), che includono il nome del processo, la serie temporale massima restituita alla volta e il percorso del file di individuazione del servizio.

```
$ cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
```

```
      - files: ["/tmp/prometheus_sd_jmx.yaml"]
  - job_name: sap
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_sap.yaml"]
```

4. Verifica che l' CloudWatch agente sia installato sulla tua istanza Amazon EC2 e che la versione sia 1.247346.1b249759 o successiva. Per installare l'agente sulla tua istanza EC2, consulta Installazione CloudWatch dell'agente. CloudWatch Per verificare la versione, consulta Ricerca di informazioni sulle versioni degli CloudWatch agenti.

5. Configura l' CloudWatch agente. Per ulteriori informazioni su come configurare il file di configurazione dell' CloudWatch agente, vedere Creare o modificare manualmente il file di configurazione dell' CloudWatch agente. Il seguente file di configurazione CloudWatch dell'agente di esempio esegue le seguenti operazioni:

   - Specifica il percorso del file di configurazione di Prometheus JMX Exporter.
   - Specifica il gruppo di log di destinazione in cui pubblicare i log dei parametri EMF.
   - Specifica due set di dimensioni per ogni nome del parametro.
   - Invia 8 metriche (4 nomi di metriche * 2 set di dimensioni per nome metrico) CloudWatch .

```
{
   "logs":{
      "logs_collected":{
         ....
      },
      "metrics_collected":{
         "prometheus":{
            "cluster_name":"prometheus-test-cluster",
            "log_group_name":"prometheus-test",
            "prometheus_config_path":"/tmp/prometheus.yaml",
            "emf_processor":{
               "metric_declaration_dedup":true,
               "metric_namespace":"CWAgent",
               "metric_unit":{
                  "jvm_threads_current":"Count",
                  "jvm_gc_collection_seconds_sum":"Second",
                  "jvm_memory_bytes_used":"Bytes"
               },
               "metric_declaration":[
                  {
```

```
                    "source_labels":[
                       "job"
                    ],
                    "label_matcher":"^jmx$",
                    "dimensions":[
                       [
                          "InstanceId",
                          "ComponentName"
                       ],
                       [
                          "ComponentName"
                       ]
                    ],
                    "metric_selectors":[
                       "^java_lang_threading_threadcount$",
                       "^java_lang_memory_heapmemoryusage_used$",
                       "^java_lang_memory_heapmemoryusage_committed$"
                    ]
                 }
              ]
           }
        }
     },
     "metrics":{
        ....
     }
}
```

`AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure`

È possibile recuperare i parametri Java specifici del carico di lavoro da Prometheus HANA database exporter per Application Insights per configurare e monitorare gli allarmi. Per ulteriori informazioni sul tagging, consulta Configurare il database SAP HANA per il monitoraggioin questa guida.

Per utilizzare AWS Systems Manager Distributor per impacchettare, installare e configurare il pacchetto di esportazione del database AWS Prometheus HANA fornito indipendentemente da Application Insights, completare i seguenti passaggi.

Prerequisiti per l'utilizzo del pacchetto database exporter di Prometheus HANA

- Versione SSM Agent 2.3.1550.0 o successiva installata

- Database SAP HANA
- Sistema operativo Linux (SUSE Linux, Linux) RedHat
- Un segreto con le credenziali di monitoraggio del database SAP HANA, utilizzando AWS Secrets Manager. Creare un segreto utilizzando il formato delle coppie chiave/valore, specificare il nome utente della chiave e immettere l'utente del database per il valore. Aggiungere unA seconda password chiave , e quindi per Valore inserire la password. Per ulteriori informazioni sulla creazione dei segreti, consulta Creazione di un segreto nella Guida per l'utente di AWS Secrets Manager . Il segreto deve essere formattato come segue:

```
{
  "username": "<database_user>",
  "password": "<database_password>"
}
```

Installazione e configurazione del paccetto **AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure**

I pacchetto `AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure` è un pacchetto Distributor SSM che è possibile utilizzare per installare e configurare database Exporter di Prometheus HANA. Quando le metriche del database HANA vengono inviate dall'esportatore del database Prometheus HANA, l' CloudWatch agente può essere configurato per recuperare le metriche per il servizio. CloudWatch

1. Crea un parametro SSM in Archivio parametri SSM per memorizzare le configurazioni Exporter. Di seguito è riportato un esempio del valore del parametro.

```
{\"exposition_port\":9668,\"multi_tenant\":true,\"timeout\":600,\"hana\":{\"host\":
\"localhost\",\"port\":30013,\"aws_secret_name\":\"HANA_DB_CREDS\",\"scale_out_mode
\":true}}
```

> **Note**
>
> In questo esempio, l'esportazione viene eseguita solo sull'istanza Amazon EC2 con il database `SYSTEM` attivo e rimarrà inattivo sulle altre istanze EC2 al fine di evitare parametri duplicati. L'esportatore può recuperare tutte le informazioni relative al tenant del database dal database `SYSTEM`.

2. Crea un parametro SSM in Archivio parametri SSM per memorizzare le query sui parametri di Exporter. Il pacchetto può accettare più di un parametro di metrica. Ciascun parametro deve avere un formato oggetto JSON valido. Il seguente è un esempio di valore del parametro:

```
{\"SELECT MAX(TIMESTAMP) TIMESTAMP, HOST, MEASURED_ELEMENT_NAME CORE,
 SUM(MAP(CAPTION, 'User Time', TO_NUMBER(VALUE), 0)) USER_PCT, SUM(MAP(CAPTION,
 'System Time', TO_NUMBER(VALUE), 0)) SYSTEM_PCT, SUM(MAP(CAPTION, 'Wait
 Time', TO_NUMBER(VALUE), 0)) WAITIO_PCT, SUM(MAP(CAPTION, 'Idle Time', 0,
 TO_NUMBER(VALUE))) BUSY_PCT, SUM(MAP(CAPTION, 'Idle Time', TO_NUMBER(VALUE), 0))
 IDLE_PCT FROM sys.M_HOST_AGENT_METRICS WHERE MEASURED_ELEMENT_TYPE = 'Processor'
 GROUP BY HOST, MEASURED_ELEMENT_NAME;\":{\"enabled\":true,\"metrics\":[{\"name\":
\"hanadb_cpu_user\",\"description\":\"Percentage of CPU time spent by HANA DB in user
 space, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"USER_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_system
\",\"description\":\"Percentage of CPU time spent by HANA DB in Kernel space,
 over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"SYSTEM_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_waitio
\",\"description\":\"Percentage of CPU time spent by HANA DB in IO mode, over the
 last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":\"WAITIO_PCT\",
\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_busy\",\"description
\":\"Percentage of CPU time spent by HANA DB, over the last minute (in seconds)\",
\"labels\":[\"HOST\",\"CORE\"],\"value\":\"BUSY_PCT\",\"unit\":\"percent\",\"type\":
\"gauge\"},{\"name\":\"hanadb_cpu_idle\",\"description\":\"Percentage of CPU time not
 spent by HANA DB, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE
\"],\"value\":\"IDLE_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"}]}}
```

   Per ulteriori informazioni sulle interrogazioni relative alle metriche, consulta il repository su. SUSE / hanadb_exporter GitHub

3. Passare alla console SSM Distributor e aprire la scheda Proprietà di Amazon. Seleziona AWSObservabilityExporter-SAP-HANADB * e scegli Installa una sola volta ExporterInstallAndConfigure.

4. Aggiorna il parametro SSM creato nel primo passaggio sostituendo "Argomenti aggiuntivi" con quanto segue:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_EXPORTER_METRICS_1": "{{ssm:<SSM_FIRST_METRICS_PARAMETER_STORE_NAME>}}",
  "SSM_EXPORTER_METRICS_2": "{{ssm:<SSM_SECOND_METRICS_PARAMETER_STORE_NAME>}}"
}
```

5. Seleziona le istanze Amazon EC2 con il database SAP HANA e scegli Esegui.

`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`

È possibile recuperare i parametri del cluster High Availability (HA) specifiche del carico di lavoro da Esportatore di cluster Prometheus HANA per Application Insights per configurare e monitorare gli allarmi per una configurazione High Availability del database SAP HANA. Per ulteriori informazioni sul tagging, consulta Configurare il database SAP HANA per il monitoraggioin questa guida.

Per utilizzare AWS Systems Manager Distributor per impacchettare, installare e configurare il pacchetto AWS di esportazione cluster Prometheus HA fornito indipendentemente da Application Insights, completare i seguenti passaggi.

Prerequisiti per l'utilizzo del cluster SSM Prometheus HA exporter

- Versione SSM Agent 2.3.1550.0 o successiva installata
- Cluster HA per Pacemaker, Corosync, SBD e DRBD
- Sistema operativo Linux (SUSE Linux, Linux) RedHat

Installazione e configurazione del paccetto **`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`**

l pacchetto `AWSObservabilityExporter-HAClusterExporterInstallAndConfigure` è un pacchetto di distribuzione SSM che è possibile utilizzare per installare e configurare Prometheus HA cluster Exporter. Quando le metriche del cluster vengono inviate dall'esportatore del database Prometheus HANA, l'agente può essere configurato per recuperare le metriche per CloudWatch il servizio. CloudWatch

1. Crea un parametro SSM in Archivio parametri SSM per archiviare le configurazioni Exporter in formato JSON. Di seguito è riportato un esempio del valore del parametro.

```
{\"port\":\"9664\",\"address\":\"0.0.0.0\",\"log-level\":\"info\",\"crm-mon-path
\":\"/usr/sbin/crm_mon\",\"cibadmin-path\":\"/usr/sbin/cibadmin\",\"corosync-
cfgtoolpath-path\":\"/usr/sbin/corosync-cfgtool\",\"corosync-quorumtool-path\":\"/
usr/sbin/corosync-quorumtool\",\"sbd-path\":\"/usr/sbin/sbd\",\"sbd-config-path\":
\"/etc/sysconfig/sbd\",\"drbdsetup-path\":\"/sbin/drbdsetup\",\"enable-timestamps
\":false}
```

Per ulteriori informazioni sulle configurazioni degli esportatori, consulta il repository su. ClusterLabs / ha_cluster_exporter GitHub

2. Passare alla console SSM Distributor e aprire la scheda Proprietà di Amazon. Seleziona AWSObservabilityExporter-HA ClusterExporterInstallAndConfigure * e scegli Installa una sola volta.
3. Aggiorna il parametro SSM creato nel primo passaggio sostituendo "Argomenti aggiuntivi" con quanto segue:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}"
}
```

4. Seleziona le istanze Amazon EC2 con il database SAP HANA e scegli Esegui.

`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`

Puoi recuperare metriche SAP specifiche del carico di lavoro NetWeaver da Prometheus SAP host exporter for Application Insights per configurare e monitorare gli allarmi per le implementazioni SAP Distributed e High Availability. NetWeaver Per ulteriori informazioni, consulta Inizia a usare Amazon CloudWatch Application Insights.

Per utilizzare AWS Systems Manager Distributor per comprimere, installare e configurare il pacchetto dell'esportatore host di SAP indipendentemente da Approfondimenti sulle applicazioni, procedi come segue.

Prerequisiti per l'utilizzo del pacchetto SSM per l'esportatore host di Prometheus SAP

- Versione SSM Agent 2.3.1550.0 o successiva installata
- Server di applicazioni SAP NetWeaver
- Sistema operativo Linux (SUSE Linux, RedHat Linux)

Installazione e configurazione del paccetto `AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`

Il `AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure` pacchetto è un pacchetto SSM Distributor che puoi utilizzare per installare e configurare l'esportatore di metriche SAP NetWeaver Prometheus. Quando le NetWeaver metriche SAP vengono inviate dall'esportatore

Prometheus, l' CloudWatch agente può essere configurato per recuperare le metriche per il servizio. CloudWatch

1. Crea un parametro SSM in Archivio parametri SSM per archiviare le configurazioni Exporter in formato JSON. Di seguito è riportato un esempio del valore del parametro.

```
{\"address\":\"0.0.0.0\",\"port\":\"9680\",\"log-level\":\"info\",\"is-HA\":false}
```

   - address

     L'indirizzo di destinazione a cui inviare i parametri di Prometheus. Il valore predefinito è `localhost`.

   - port

     La porta di destinazione a cui inviare i parametri di Prometheus. Il valore predefinito è `9680`.

   - is-HA

     `true` NetWeaver per implementazioni SAP High Availability. Per tutte le altre implementazioni il valore è `false`.

2. Passare alla console SSM Distributor e aprire la scheda Proprietà di Amazon. Seleziona AWSObservabilityExporter-SAP-SAP e HostExporterInstallAndConfigure scegli Installa una sola volta.

3. Aggiorna il parametro SSM creato nel primo passaggio sostituendo "Argomenti aggiuntivi" con quanto segue:

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_INSTANCES_NUM": "<instances_number seperated by comma>"
}
```

   Esempio

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:exporter_config_paramter}}",
  "SSM_INSTANCES_NUM": "11,12,10",
  "SSM_SID": "PR1"
}
```

4. Seleziona le istanze Amazon EC2 con NetWeaver applicazioni SAP e scegli Esegui.

> **Note**
>
> L'esportatore Prometheus elabora le metriche SAP su un endpoint locale. NetWeaver L'endpoint locale è accessibile solo agli utenti del sistema operativo sull'istanza Amazon EC2. Pertanto, le metriche sono disponibili per tutti gli utenti del sistema operativo dopo l'installazione del pacchetto dell'esportatore. L'endpoint locale predefinito è `localhost:9680/metrics`.

`AWSObservabilityExporter-SQLExporterInstallAndConfigure`

È possibile recuperare i parametri SQL Server specifici del carico di lavoro da Prometheus JMX exporter per Application Insights per monitorare gli allarmi principali.

Per utilizzare AWS Systems Manager Distributor per comprimere, installare e configurare il pacchetto dell'esportatore SAP indipendentemente da Approfondimenti sulle applicazioni, procedi come segue.

Prerequisiti per l'utilizzo del pacchetto SSM dell'esportatore SQL Prometheus

- Versione SSM Agent 2.3.1550.0 o successiva installata
- Istanza Amazon EC2 che esegue SQL Server su Windows con autenticazione utente SQL Server abilitata.
- Un utente SQL Server con le seguenti autorizzazioni:

```
GRANT VIEW ANY DEFINITION TO
```

```
GRANT VIEW SERVER STATE TO
```

- Un segreto contenente la stringa di connessione al database che utilizza AWS Secrets Manager. Per ulteriori informazioni sulla creazione dei segreti, consulta Creazione di un segreto nella Guida per l'utente di AWS Secrets Manager . Il segreto deve essere formattato come segue:

```
{
  "data_source_name":"sqlserver://<username>:<password>@localhost:1433"
}
```

> **Note**
>
> Se la password o il nome utente contengono caratteri speciali, è necessario codificare in percentuale i caratteri speciali per garantire una connessione corretta al database.

Installazione e configurazione del paccetto **AWSObservabilityExporter-SQLExporterInstallAndConfigure**

l pacchetto `AWSObservabilityExporter-SQLExporterInstallAndConfigure` è un pacchetto SSM Distributor che puoi utilizzare per installare e configurare l'esportatore di parametri SQL Prometheus. Quando le metriche vengono inviate dall'esportatore Prometheus, l'agente può essere configurato per recuperare CloudWatch le metriche per il servizio. CloudWatch

1. In base alle preferenze, prepara la configurazione YAML di SQL Exporter. La seguente configurazione di esempio ha un solo parametro configurato. Utilizza la configurazione di esempio per aggiornare la configurazione con parametri aggiuntivi o creare una configurazione personalizzata.

```
---
global:
  scrape_timeout_offset: 500ms
  min_interval: 0s
  max_connections: 3
  max_idle_connections: 3
target:
  aws_secret_name: <SECRET_NAME>
  collectors:
    - mssql_standard
collectors:
  - collector_name: mssql_standard
    metrics:
    - metric_name: mssql_batch_requests
      type: counter
      help: 'Number of command batches received.'
      values: [cntr_value]
      query: |
        SELECT cntr_value
        FROM sys.dm_os_performance_counters WITH (NOLOCK)
        WHERE counter_name = 'Batch Requests/sec'
```

2. Copia il file di configurazione YAML dell'esportatore SQL Prometheus codificato come Base64 in un nuovo parametro SSM nell'Archivio parametri SSM.
3. Passare alla console SSM Distributor e aprire la scheda Proprietà di Amazon. Seleziona -SQL e scegli AWSObservabilityExporter Installa una sola volta. ExporterInstallAndConfigure
4. Sostituisci gli "Argomenti aggiuntivi" con le seguenti informazioni. `SSM_PARAMETER_NAME` è il nome del parametro creato nel passaggio 2.

```
{
  "SSM_EXPORTER_CONFIGURATION":
    "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
    "SSM_PROMETHEUS_PORT": "9399",
    "SSM_WORKLOAD_NAME": "SQL"
}
```

5. Seleziona l'istanza Amazon EC2 con il database SQL Server, quindi scegli esegui.

AWS Documenti Systems Manager (SSM) utilizzati da CloudWatch Application Insights

Application Insights utilizza i documenti SSM elencati in questa sezione per definire le azioni che AWS Systems Manager esegue sulle istanze gestite. Questi documenti utilizzano la capacità `Run Command` di Systems Manager per automatizzare le attività necessarie per eseguire le funzionalità di monitoraggio di Application Insights. Le pianificazioni di esecuzione di questi documenti sono gestite da Application Insights e non possono essere modificate.

Per ulteriori informazioni sui documenti SSM, consulta Documenti AWS Systems Manager nella Guida per l'utente di AWS Systems Manager .

Documenti gestiti da CloudWatch Application Insights

La tabella seguente elenca i documenti SSM gestiti da Application Insights.

Nome del documento	Descrizione	Pianificazione delle esecuzioni
`AWSEC2-DetectWorkload`	Rileva automaticamente le applicazioni in esecuzione nell'ambiente applicativo che	Questo documento viene eseguito ogni ora nell'ambiente dell'applicazione per

Nome del documento	Descrizione	Pianificazione delle esecuzioni
	possono essere configurate per il monitoraggio da parte di Application Insights.	ottenere i dettagli up-to-date dell'applicazione.
`AWSEC2-CheckPerformanceCounterSets`	Verifica se gli spazi dei nomi dei contatori di prestazioni sono abilitati sulle istanze Windows di Amazon EC2.	Questo documento viene eseguito ogni ora nell'ambiente applicativo e monitora i parametri di dei contatori di prestazioni solo se gli spazi dei nomi corrispondenti sono abilitati.
`AWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure`	Installa e configura CloudWatch l'agente in base alla configurazione di monitoraggio dei componenti dell'applicazione.	Questo documento viene eseguito ogni 30 minuti per garantire che la configurazione dell' CloudWatch agente sia sempre accurata e. up-to-date Il documento viene inoltre eseguito immediatamente dopo aver apportato una modifica alla configurazione di monitoraggio dell'applicazione, ad esempio l'aggiunta o la rimozione di parametri o l'aggiornamento delle configurazioni dei log.

Documenti gestiti da AWS Systems Manager

I seguenti documenti vengono utilizzati da CloudWatch Application Insights e gestiti da Systems Manager.

`AWS-ConfigureAWSPackage`

Application Insights utilizza questo documento per installare e disinstallare i pacchetti Prometheus Exporter Distributor, per raccogliere parametri specifici del carico di lavoro e per consentire il

monitoraggio completo dei carichi di lavoro sulle istanze Amazon EC2 dei clienti. CloudWatch Application Insights installa i pacchetti Prometheus exporter distributor solo se il carico di lavoro di destinazione correlato è in esecuzione sull'istanza.

La tabella seguente elenca i pacchetti Prometheus exporter distributor e i carichi di lavoro di destinazione correlati.

Nome del pacchetto Prometheus exporter distributor	Carico di lavoro di destinazione
`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`	SAP HANA HA
`AWSObservabilityExporter-JMXExporterInstallAndConfigure`	Java/JMX
`AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure`	SAP HANA
`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`	NetWeaver
`AWSObservabilityExporter-SQLExporterInstallAndConfigure`	SQL Server (Windows) e SAP ASE (Linux)

`AmazonCloudWatch-ManageAgent`

Application Insights utilizza questo documento per gestire lo stato e la configurazione di CloudWatch Agent sulle istanze e per raccogliere parametri e log interni a livello di sistema dalle istanze Amazon EC2 su tutti i sistemi operativi.

Inizia a usare Amazon CloudWatch Application Insights

Per iniziare a usare CloudWatch Application Insights, verifica di aver soddisfatto i seguenti prerequisiti e di aver creato una policy IAM. Quindi, puoi iniziare a utilizzare il collegamento alla

console per abilitare CloudWatch Application Insights. Per configurare le risorse dell'applicazione, segui le fasi descritte in Impostare, configurare e gestire l'applicazione per il monitoraggio.

Indice

Accedi a CloudWatch Application Insights

È possibile accedere e gestire CloudWatch Application Insights tramite una delle seguenti interfacce:

- CloudWatch console. Per aggiungere monitor per la tua applicazione, scegli Application Insights in Insights nel riquadro di navigazione a sinistra della CloudWatch console. Dopo aver configurato l'applicazione, puoi utilizzare la CloudWatch console per visualizzare e analizzare i problemi rilevati.
- AWS Interfaccia a riga di comando (AWS CLI). È possibile utilizzare il AWS CLI per accedere alle operazioni AWS dell'API. Per ulteriori informazioni, vedere Installazione dell'interfaccia a riga di AWS comando nella Guida per l'utente dell'interfaccia a AWS riga di comando. Per informazioni sull'API di Application Insights, consulta l'Amazon CloudWatch Application Insights API Reference.

Prerequisiti

È necessario completare i seguenti prerequisiti per configurare un' CloudWatch applicazione con Application Insights:

- AWS Systems Manager abilitazione: installa Systems Manager Agent (SSM Agent) sulle tue istanze Amazon EC2 e abilita le istanze per SSM. Per la procedura di installazione dell'Agente SSM, consulta Configurazione di AWS Systems Manager nella Guida per l'utente di AWS Systems Manager .
- Ruolo dell'istanza EC2: è necessario collegare i seguenti ruoli di istanza Amazon EC2 per abilitare Systems Manager

  - È necessario allegare il ruolo `AmazonSSMManagedInstanceCore` per abilitare Systems Manager. Per ulteriori informazioni, consulta Esempi di policy basate sull'identità nella AWS Systems Manager.
  - È necessario allegare la `CloudWatchAgentServerPolicy` policy per consentire la trasmissione dei parametri e dei log delle istanze. CloudWatch Per ulteriori informazioni, consulta Creare ruoli e utenti IAM da utilizzare con l'agente. CloudWatch
- AWS gruppi di risorse: per integrare le tue applicazioni in CloudWatch Application Insights, crea un gruppo di risorse che includa tutte le AWS risorse associate utilizzate dallo stack di applicazioni. Questo comprende Application Load Balancer, istanze Amazon EC2 che eseguono IIS e front-end Web, livelli worker .NET e database SQL Server. Per ulteriori informazioni sui componenti delle applicazioni e sugli stack tecnologici supportati da Application Insights, consulta. Componenti dell'applicazione supportati CloudWatch Application Insights include automaticamente i gruppi di Auto Scaling che utilizzano gli stessi tag o CloudFormation stack del gruppo di risorse, poiché i gruppi di Auto Scaling non sono supportati dai gruppi di risorse. CloudFormation Per ulteriori informazioni, consulta Nozioni di base su AWS Resource Groups.
- Autorizzazioni IAM: per gli utenti che non dispongono di accesso amministrativo, è necessario creare una policy AWS Identity and Access Management (IAM) che consenta ad Application Insights di creare un ruolo collegato al servizio e collegarlo all'identità dell'utente. Per ulteriori informazioni su come creare la policy IAM, consulta Policy IAM.
- Ruolo collegato ai servizi: Application Insights utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato al servizio viene creato per te quando crei la tua prima applicazione Application Insights nella console Application Insights. Per ulteriori informazioni, consulta la pagina Utilizzo di ruoli collegati ai servizi per CloudWatch Application Insights.
- Supporto dei parametri dei contatori di prestazioni per le istanze EC2 Windows: per monitorare i parametri dei contatori di prestazioni sulle istanze Amazon EC2 Windows, è necessario installare i contatori di prestazioni sulle istanze. Per i parametri dei contatori di prestazioni e i nomi dei set di contatori di prestazioni corrispondenti, consulta la sezione relativa ai parametri dei contatori di prestazioni. Per ulteriori informazioni sui contatori di prestazioni, consulta la sezione relativa ai contatori di prestazioni.
- CloudWatch Agente Amazon: Application Insights installa e configura l' CloudWatch agente. Se hai installato un CloudWatch agente, Application Insights mantiene la tua configurazione. Per evitare un conflitto di unione, rimuovete la configurazione delle risorse che desiderate utilizzare in Application Insights dal file di configurazione dell' CloudWatch agente esistente. Per ulteriori informazioni, consulta Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.

Policy IAM

Per utilizzare CloudWatch Application Insights, devi creare una policy AWS Identity and Access Management (IAM) e collegarla al tuo utente, gruppo o ruolo. Per ulteriori informazioni su utenti, gruppi e ruoli, consulta Identità IAM (utenti, gruppi di utenti e ruoli). La policy IAM definisce le autorizzazioni utente.

Per creare una policy IAM tramite la console

Per creare una policy IAM tramite la console IAM, seguire i passaggi seguenti.

1. Vai alla console IAM. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
2. Nella parte superiore della pagina, seleziona Create policy (Crea policy).
3. Seleziona la scheda JSON.
4. Copia e incolla il seguente documento JSON nella scheda JSON.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "applicationinsights:*",
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "resource-groups:ListGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

5. Seleziona Review policy (Esamina policy).
6. Inserisci un nome per la policy, ad esempio «AppInsightsPolicy.» Opzionalmente, immetti una Description (Descrizione).
7. Seleziona Create Policy (Crea policy).
8. Nel pannello di navigazione sulla sinistra, scegli Gruppi di utenti, Utenti o Ruoli.
9. Seleziona il nome del gruppo di utenti, dell'utente o del ruolo a cui desideri collegare la policy.
10. Seleziona Add permissions (Aggiungi autorizzazioni).

11. Seleziona Attach existing policies directly (Collega direttamente le policy esistenti).
12. Cerca la policy appena creata e seleziona la casella di controllo a sinistra del nome di policy.
13. Seleziona Next: Review (Successivo: Rivedi).
14. Verifica che la policy corretta sia elencata e seleziona Add permissions (Aggiungi autorizzazioni).
15. Assicurati di accedere con l'utente associato alla policy che hai appena creato quando usi CloudWatch Application Insights.

Per creare una policy IAM utilizzando il AWS CLI

Per creare una policy IAM utilizzando il AWS CLI, esegui l'operazione create-policy dalla riga di comando utilizzando il documento JSON riportato sopra come file nella cartella corrente.

Per creare una policy IAM utilizzando AWS Tools for Windows PowerShell

Per creare una policy IAM utilizzando il AWS Tools for Windows PowerShell, esegui il cmdlt New-IAMPolicy utilizzando il documento JSON riportato sopra come file nella cartella corrente.

Autorizzazioni ruolo IAM per l'integrazione delle applicazioni basate sull'account

Se si desidera integrare tutte le risorse del proprio account e scegliere di non utilizzare la Policy gestita di Application Insights per avere accesso completo alla funzionalità di Application Insights, è necessario allegare le seguenti autorizzazioni al ruolo IAM in modo che Application Insights possa scoprire tutte le risorse dell'account:

```
"ec2:DescribeInstances"
"ec2:DescribeNatGateways"
"ec2:DescribeVolumes"
"ec2:DescribeVPCs"
"rds:DescribeDBInstances"
"rds:DescribeDBClusters"
"sqs:ListQueues"
"elasticloadbalancing:DescribeLoadBalancers"
"autoscaling:DescribeAutoScalingGroups"
"lambda:ListFunctions"
"dynamodb:ListTables"
"s3:ListAllMyBuckets"
"sns:ListTopics"
"states:ListStateMachines"
"apigateway:GET"
```

```
"ecs:ListClusters"
"ecs:DescribeTaskDefinition"
"ecs:ListServices"
"ecs:ListTasks"
"eks:ListClusters"
"eks:ListNodegroups"
"fsx:DescribeFileSystems"
"route53:ListHealthChecks"
"route53:ListHostedZones"
"route53:ListQueryLoggingConfigs"
"route53resolver:ListFirewallRuleGroups"
"route53resolver:ListFirewallRuleGroupAssociations"
"route53resolver:ListResolverEndpoints"
"route53resolver:ListResolverQueryLogConfigs"
"route53resolver:ListResolverQueryLogConfigAssociations"
"logs:DescribeLogGroups"
"resource-explorer:ListResources"
```

Impostare, configurare e gestire l'applicazione per il monitoraggio

Questa sezione fornisce i passaggi per impostare, configurare e gestire CloudWatch l'applicazione Application Insights utilizzando la console, la e. AWS CLI AWS Tools for Windows PowerShell

Argomenti

- Configura, configura e gestisci l'applicazione per il monitoraggio dalla CloudWatch console
- Impostare, configurare e gestire l'applicazione per il monitoraggio tramite la riga di comando
- CloudWatch Eventi e notifiche di Application Insights per i problemi rilevati

Configura, configura e gestisci l'applicazione per il monitoraggio dalla CloudWatch console

Questa sezione fornisce i passaggi per impostare, configurare e gestire l'applicazione per il monitoraggio dalla CloudWatch console.

Procedure della console

- Aggiunta e configurazione di un'applicazione
- Abilitazione di monitoraggio delle risorse di Application Insights per Amazon ECS e Amazon EKS
- Disabilitazione del monitoraggio per un componente dell'applicazione
- Eliminazione di un'applicazione

Aggiunta e configurazione di un'applicazione

Aggiungere e configurare un'applicazione dalla CloudWatch console

Per iniziare a utilizzare CloudWatch Application Insights dalla CloudWatch console, procedi nel seguente modo.

1. Avvia. Apri la pagina di destinazione della CloudWatch console. Nel pannello di navigazione a sinistra, scegli Application Insights sotto Application Insights (Informazioni dettagliate applicazione). La pagina che si apre mostra l'elenco delle applicazioni monitorate con CloudWatch Application Insights, insieme al relativo stato di monitoraggio.
2. Aggiunta di un'applicazione. Per configurare il monitoraggio per la propria applicazione, scegli Aggiungere un'applicazione. Quando si sceglie Aggiungere un'applicazione, viene richiesto di Scegliere il tipo di applicazione.

   - Applicazione basata su resource groups. Quando si seleziona questa opzione, è possibile scegliere quali resource groups monitorare in questo account. Per utilizzare più applicazioni su un componente, è necessario utilizzare il monitoraggio basato su gruppi di risorse.
   - Applicazione basata su account. Quando si seleziona questa opzione, è possibile monitorare tutte le risorse in questo account. Se si desidera monitorare tutte le risorse di un account, si consiglia questa opzione rispetto all'opzione basata sul gruppo di risorse perché il processo di integrazione delle applicazioni è più veloce.

   > **Note**
   >
   > Non è possibile combinare il monitoraggio basato su gruppi di risorse con il monitoraggio basato su account utilizzando Application Insights. Per modificare il tipo di applicazione, è necessario eliminare tutte le applicazioni monitorate e Scegliere tipo di applicazione.

   Quando aggiungi la tua prima applicazione per il monitoraggio, CloudWatch Application Insights crea un ruolo collegato al servizio nell'account, che concede ad Application Insights le autorizzazioni per chiamare altri AWS servizi per tuo conto. Per ulteriori informazioni sul ruolo collegato al servizio creato dall'account da Application Insights, consulta Utilizzo di ruoli collegati ai servizi per CloudWatch Application Insights.

3. Resource-based application monitoring

    1. Seleziona gruppo di risorse. Nella pagina Specificare i dettagli dell'applicazione, seleziona il gruppo di AWS risorse che contiene le risorse dell'applicazione dall'elenco a discesa. Queste risorse includono server front-end, sistemi di bilanciamento del carico, gruppi Auto Scaling e server di database.

        Se non hai creato un resource group per l'applicazione è possibile crearne uno scegliendo Crea nuovo resource group. Per ulteriori informazioni sui Resource Groups, consulta la Guida per l'utente di AWS Resource Groups .
    2. Monitora CloudWatch gli eventi. Seleziona la casella di controllo per integrare il monitoraggio di Application Insights con CloudWatch Events per ottenere informazioni da Amazon EBS, Amazon EC2 AWS CodeDeploy, Amazon ECS AWS Health , API e notifiche, Amazon RDS, Amazon S3 e. AWS Step Functions
    3. Integrazione con AWS Systems Manager OpsCenter. Per visualizzare e ricevere notifiche quando vengono rilevati problemi per applicazioni selezionate, selezionare la casella di controllo Genera Systems Manager OpsCenter OpsItems per azioni correttive. Per tenere traccia delle operazioni eseguite per risolvere gli elementi di lavoro operativi (OpsItems) correlati alle AWS risorse, fornisci l'argomento SNS ARN.
    4. Tag: facoltativi. CloudWatch Application Insights supporta sia gruppi di risorse basati su tag che gruppi di risorse CloudFormation basati su tag (ad eccezione dei gruppi Auto Scaling). Per ulteriori informazioni, consulta Utilizzo dell'editor di tag.
    5. Seleziona Successivo.

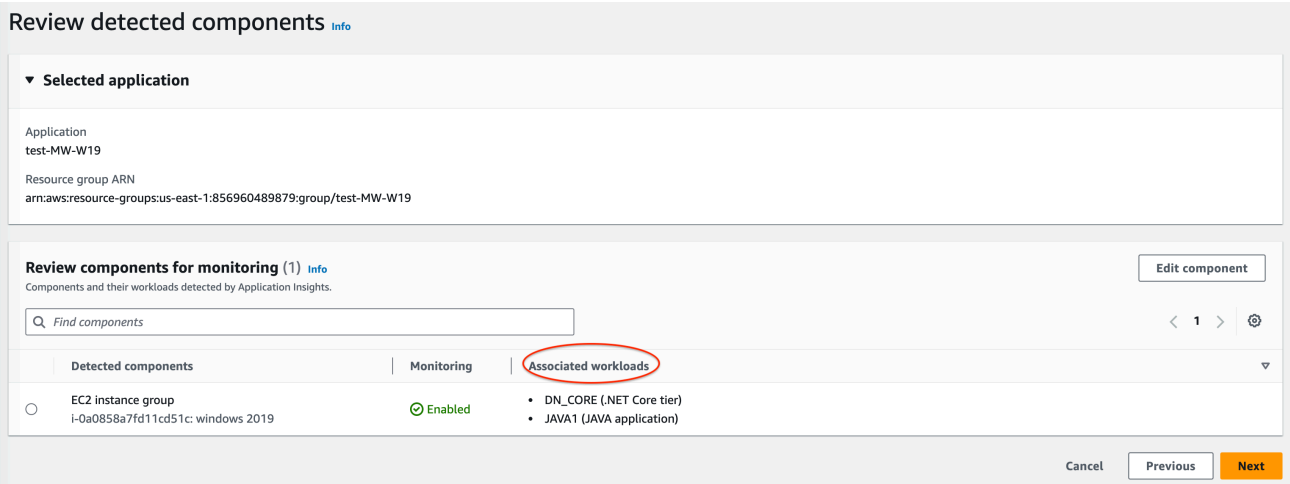
        Un ARN per l'applicazione verrà generato nel seguente formato.

        ```
        arn:partition:applicationinsights:region:account-id:application/resource-group/resource-group-name
        ```

        Esempio

        ```
        arn:aws:applicationinsights:us-east-1:123456789012:application/resource-group/my-resource-group
        ```
    6. Nella pagina Revisione componenti rilevati, in Rivedi i componenti per il monitoraggio, la tabella riporta i componenti rilevati e i carichi di lavoro rilevati associati.

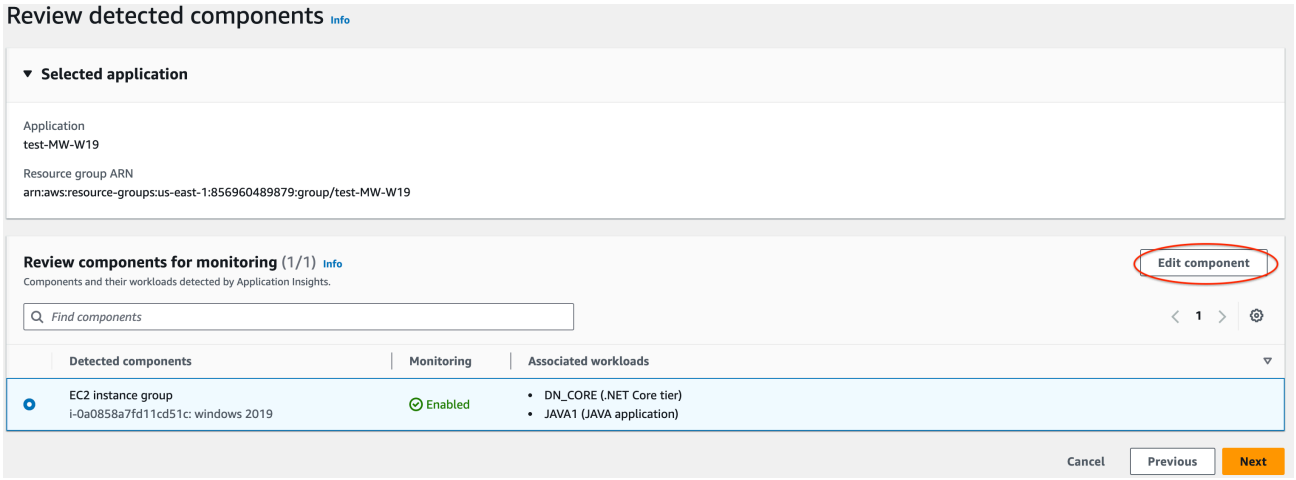
> **Note**
>
> Per i componenti che supportano più carichi di lavoro personalizzati, puoi monitorare fino a cinque carichi di lavoro per ogni componente. Questi carichi di lavoro verranno monitorati separatamente dal componente.



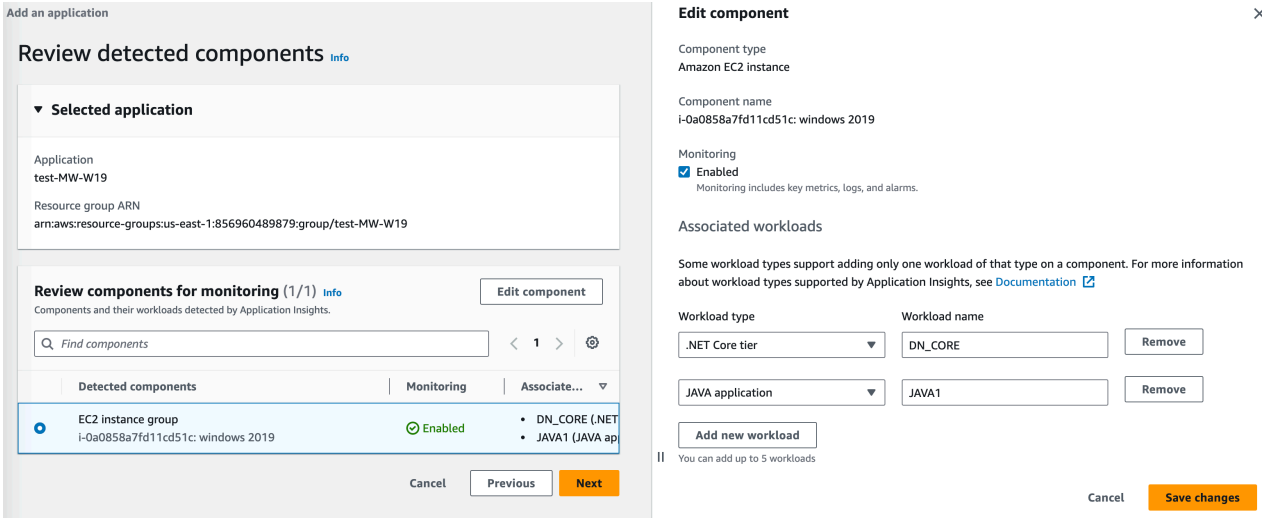
In Carichi di lavoro associati, ci sono diversi possibili messaggi che vengono visualizzati se un carico di lavoro non è elencato.

- Impossibile rilevare i carichi di lavoro: si è verificato un problema durante il tentativo di rilevare i carichi di lavoro. Assicurati di aver completato le fasi descritte in Prerequisiti. Se devi aggiungere i carichi di lavoro, scegli Modifica componente.
- Nessun carico di lavoro rilevato: non abbiamo rilevato alcun carico di lavoro. Potrebbe essere necessario aggiungere i carichi di lavoro. Per farlo, scegli Modifica componente.
- Non applicabile: il componente non supporta carichi di lavoro personalizzati e verrà monitorato con parametri, allarmi e log predefiniti. Non è possibile aggiungere carichi di lavoro a questi componenti.

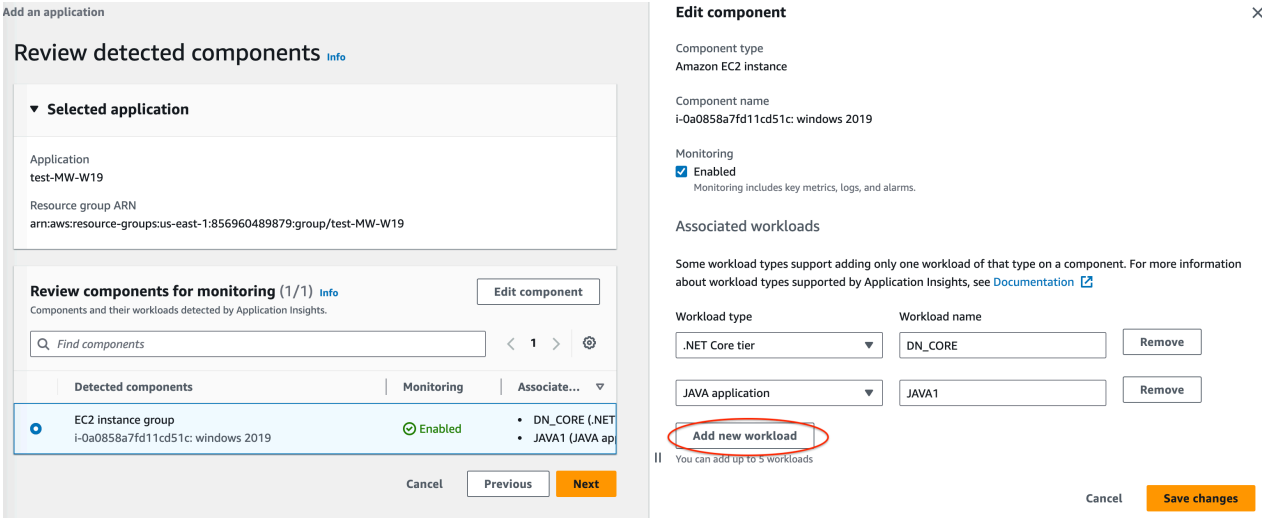
7. Per modificare un componente, selezionalo, quindi scegli Modifica componente. Si apre un pannello laterale con i carichi di lavoro rilevati sul componente. In questo pannello, potrai modificare i dettagli del componente e aggiungere nuovi carichi di lavoro.



- Per modificare il tipo o il nome del carico di lavoro, utilizza l'elenco a discesa.

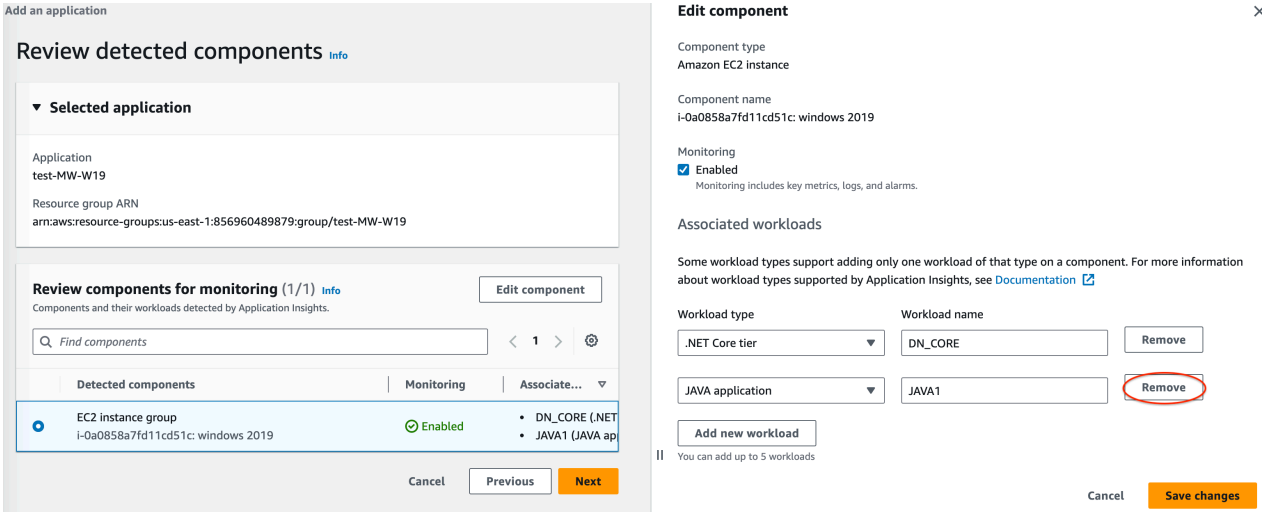


- Per aggiungere un carico di lavoro al componente, scegli Aggiungi nuovo carico di lavoro.

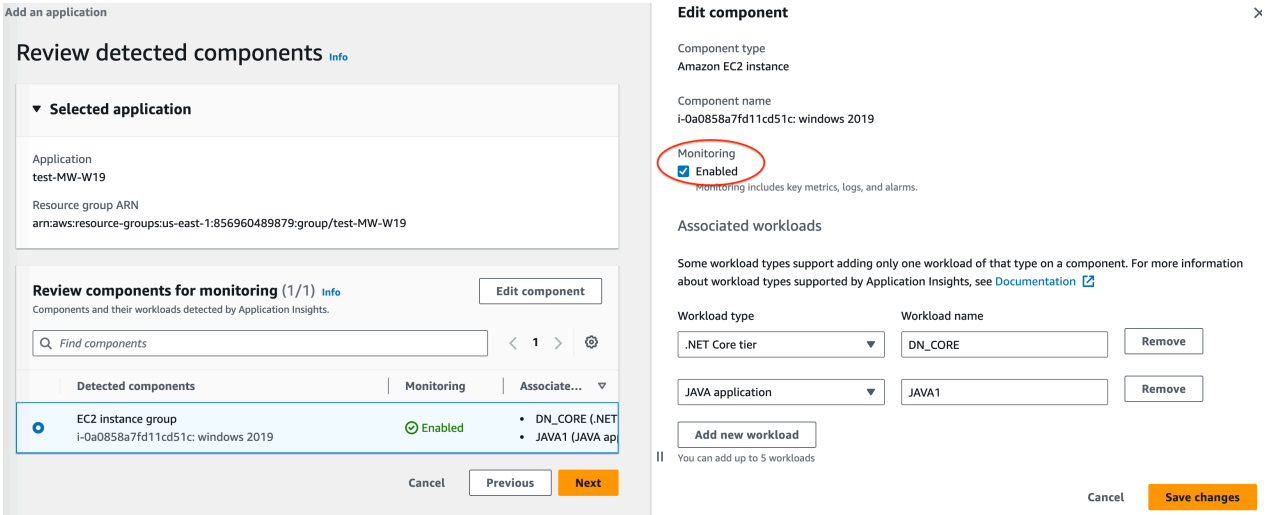


- Se l'opzione Aggiungi nuovo carico di lavoro non viene visualizzata, questo componente non supporta carichi di lavoro multipli.
- Se l'intestazione Carichi di lavoro associati non viene visualizzata, questo componente non supporta carichi di lavoro personalizzati.

- Per rimuovere un carico di lavoro, scegli Rimuovi accanto al carico di lavoro che desideri rimuovere dal monitoraggio.



- Per disabilitare il monitoraggio dell'intero componente, deseleziona la casella di controllo Monitoraggio.



- Una volta finito di modificare il componente, scegli Salva modifiche nell'angolo in basso a destra. Qualsiasi modifica ai carichi di lavoro per un componente è visibile nella tabella Rivedi i componenti per il monitoraggio in Carichi di lavoro associati.

8. Nella pagina Revisione componenti rilevati, seleziona Avanti.

9. La pagina Specifica i dettagli dei componenti include tutti i componenti con carichi di lavoro associati personalizzabili del passaggio precedente.

   > **Note**
   >
   > Se l'intestazione di un componente ha un tag facoltativo, i dettagli aggiuntivi per i carichi di lavoro in quel componente saranno facoltativi.

   Se un componente non viene visualizzato in questa pagina, significa che non dispone di dettagli aggiuntivi che possono essere specificati in questo passaggio.

10. Seleziona Successivo.
11. Nella pagina Rivedi e invia, esamina tutti i dettagli dei componenti e del carico di lavoro monitorati.
12. Scegli Invia.

Account-based application monitoring

1. Nome applicazione. Immetti un nome per la tua applicazione basata su account.
2. Monitoraggio automatico di nuove risorse. Per impostazione predefinita, Application Insights utilizza le impostazioni consigliate per configurare il monitoraggio dei componenti delle risorse aggiunti all'account dopo l'integrazione all'applicazione. È possibile escludere il monitoraggio delle risorse aggiunte dopo l'integrazione dell'applicazione deselezionando la casella di controllo.
3. Monitora CloudWatch gli eventi. Seleziona la casella di controllo per integrare il monitoraggio di Application Insights con CloudWatch Events per ottenere informazioni da Amazon EBS, Amazon EC2 AWS CodeDeploy, Amazon ECS AWS Health , API e notifiche, Amazon RDS, Amazon S3 e. AWS Step Functions
4. Integrazione con AWS Systems Manager OpsCenter. Per visualizzare e ricevere notifiche quando vengono rilevati problemi per applicazioni selezionate, selezionare la casella di controllo Genera Systems Manager OpsCenter OpsItems per azioni correttive. Per tenere traccia delle operazioni eseguite per risolvere gli elementi di lavoro operativi (OpsItems) correlati alle AWS risorse, fornisci l'argomento SNS ARN.

5. Tag: facoltativi. CloudWatch Application Insights supporta sia gruppi di risorse basati su tag che gruppi di risorse CloudFormation basati su tag (ad eccezione dei gruppi Auto Scaling). Per ulteriori informazioni, consulta Utilizzo dell'editor di tag.
6. Risorse rilevate. Tutte le risorse rilevate nel tuo account vengono aggiunte a questo elenco. Se Application Insights non è in grado di scoprire tutte le risorse del tuo account, viene visualizzato un messaggio di errore nella parte superiore della pagina. Questo messaggio include un collegamento alla documentazione su come aggiungere le autorizzazioni richieste.
7. Seleziona Successivo.

   Un ARN per l'applicazione verrà generato nel seguente formato.

   ```
   arn:partition:applicationinsights:region:account-id:application/
   TBD/application-name
   ```

   Esempio

   ```
   arn:aws:applicationinsights:us-east-1:123456789012:application/TBD/my-
   application
   ```

4. Dopo aver inviato la configurazione di monitoraggio dell'applicazione, verrà visualizzata la pagina dei dettagli dell'applicazione, dove è possibile visualizzare il Riepilogo domanda, l'elenco di Componenti monitorati e Componenti non monitorati e, selezionando le schede accanto a Componenti, la Cronologia della configurazione, Modelli di log e qualsiasi Tag applicato.

   Per visualizzare le informazioni dettagliate per l'applicazione, scegli Visualizzazione delle informazioni dettagliate.

   È possibile aggiornare le selezioni per il monitoraggio e l'integrazione CloudWatch degli eventi con AWS Systems Manager OpsCenter scegliendo Modifica.

   In Componenti è possibile selezionare il menu Operazioni per creare, modificare o separare un gruppo di istanze.

   È possibile gestire il monitoraggio dei componenti, inclusi livello applicazione, gruppi di log, log eventi, parametri e allarmi personalizzati, selezionando il punto elenco accanto a un componente e scegliendo Gestisci il monitoraggio.

Abilitazione di monitoraggio delle risorse di Application Insights per Amazon ECS e Amazon EKS

È possibile abilitare Application Insights per monitorare applicazioni e microservizi containerizzati dalla console Container Insights. Application Insights supporta il monitoraggio delle seguenti risorse:

- Cluster Amazon ECS
- Servizi Amazon ECS
- Attività di Amazon ECS
- Cluster Amazon EKS

Quando Application Insights è abilitato, fornisce metriche e log consigliati, rileva potenziali problemi, genera CloudWatch eventi e crea dashboard automatici per le applicazioni e i microservizi containerizzati.

È possibile abilitare Application Insights per le risorse containerizzate dalle console Container Insights o Application Insights.

Abilitazione di Application Insights dalla console di Container Insights

Dalla console Container Insights, sul panneo di controllo di Container Insights Controllo delle prestazioni, scegli Configurazione automatica di Application Insights. Quando Application Insights è abilitato, visualizza i dettagli sui problemi rilevati.

Abilitazione di Application Insights dalla console di Application Insights

Quando i cluster ECS vengono visualizzati nell'elenco dei componenti, Application Insights abilita automaticamente il monitoraggio aggiuntivo dei container con Container Insights.

Per i cluster EKS, è possibile abilitare un monitoraggio aggiuntivo con Container Insights per fornire informazioni diagnostiche, ad esempio errori di riavvio del container, che consentono di isolare e risolvere i problemi. Sono necessari ulteriori passaggi per configurare Container Insights per EKS. Per informazioni, consulta Configurazione di Container Insights su Amazon EKS e Kubernetes per le fasi della configurazione di Container Insights su EKS.

Il monitoraggio aggiuntivo per EKS con Container Insights è supportato sulle istanze Linux con EKS.

Per ulteriori informazioni sul supporto di Container Insights per i cluster ECS ed EKS, vedere Container Insights.

Disabilitazione del monitoraggio per un componente dell'applicazione

Per disabilitare il monitoraggio per un componente dell'applicazione, dalla pagina dei dettagli dell'applicazione seleziona il componente per cui disabilitare il monitoraggio. Scegli Operazioni, e poi Rimuovi dal monitoraggio.

Eliminazione di un'applicazione

Per eliminare un'applicazione, dalla CloudWatch dashboard, nel riquadro di navigazione a sinistra, scegli Application Insights in Insights. Seleziona l'applicazione da eliminare. In Operazioni, scegli Elimina applicazione. Questo elimina il monitoraggio ed elimina tutti i monitor salvati per i componenti dell'applicazione. Le risorse dell'applicazione non vengono eliminate.

Impostare, configurare e gestire l'applicazione per il monitoraggio tramite la riga di comando

Questa sezione fornisce i passaggi per l'impostazione, la configurazione e la gestione dell'applicazione per il monitoraggio tramite AWS CLI e AWS Tools for Windows PowerShell.

Procedure della riga di comando

- Aggiunta e gestione di un'applicazione
- Gestione e aggiornamento del monitoraggio
- Configurazione del monitoraggio per i gruppi di disponibilità SQL Always On
- Configurazione del monitoraggio per MySQL RDS
- Configurazione del monitoraggio per MySQL EC2
- Configurazione del monitoraggio per PostgreSQL RDS
- Configurazione del monitoraggio per PostgreSQL EC2
- Configurare il monitoraggio per Oracle RDS
- Configurare il monitoraggio per Oracle EC2

Aggiunta e gestione di un'applicazione

È possibile aggiungere, ottenere informazioni su, gestire e configurare l'applicazione Application Insights tramite la riga di comando.

Argomenti

- Aggiunta di un'applicazione

- Descrizione di un'applicazione
- Elenco dei componenti in un'applicazione
- Descrizione di un componente
- Raggruppamento di risorse simili in un componente personalizzato
- Annullamento del raggruppamento di un componente personalizzato
- Aggiornamento di un'applicazione
- Aggiornamento di un componente personalizzato

Aggiunta di un'applicazione

Aggiungere un'applicazione utilizzando il AWS CLI

Per utilizzare il comando AWS CLI per aggiungere un'applicazione per il gruppo di risorse denominata`my-resource-group`, con OpsCenter abilitato a consegnare l'OpsItem creato all'argomento SNS `arn:aws:sns:us-east-1:123456789012:MyTopic` ARN, utilizzare il comando seguente.

```
aws application-insights create-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Aggiungere un'applicazione utilizzando AWS Tools for Windows PowerShell

Da utilizzare AWS Tools for Windows PowerShell per aggiungere un'applicazione per il gruppo di risorse chiamata `my-resource-group` con OpsCenter enabled per consegnare l'OpsItem creato all'argomento SNS `arn:aws:sns:us-east-1:123456789012:MyTopic` ARN, utilizzare il comando seguente.

```
New-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Descrizione di un'applicazione

Descrivi un'applicazione utilizzando il AWS CLI

Per utilizzare il comando AWS CLI per descrivere un'applicazione creata su un gruppo di risorse chiamato`my-resource-group`, utilizzate il comando seguente.

```
aws application-insights describe-application --resource-group-name my-resource-group
```

Descrivi un'applicazione utilizzando AWS Tools for Windows PowerShell

Per utilizzare il comando AWS Tools for Windows PowerShell per descrivere un'applicazione creata su un gruppo di risorse chiamato`my-resource-group`, utilizzare il comando seguente.

```
Get-CWAIApplication -ResourceGroupName my-resource-group
```

Elenco dei componenti in un'applicazione

Elenca i componenti di un'applicazione utilizzando il AWS CLI

Per utilizzare AWS CLI per elencare i componenti creati su un gruppo di risorse chiamato`my-resource-group`, utilizzate il comando seguente.

```
aws application-insights list-components --resource-group-name my-resource-group
```

Elenca i componenti di un'applicazione utilizzando AWS Tools for Windows PowerShell

Per utilizzare AWS Tools for Windows PowerShell per elencare i componenti creati su un gruppo di risorse chiamato`my-resource-group`, utilizzare il comando seguente.

```
Get-CWAIComponentList -ResourceGroupName my-resource-group
```

Descrizione di un componente

Descrivete un componente utilizzando il AWS CLI

È possibile utilizzare il seguente AWS CLI comando per descrivere un componente chiamato `my-component` che appartiene a un'applicazione creata su un gruppo di risorse chiamato`my-resource-group`.

```
aws application-insights describe-component --resource-group-name my-resource-group --
component-name my-component
```

Descrivi un componente usando AWS Tools for Windows PowerShell

È possibile utilizzare il seguente AWS Tools for Windows PowerShell comando per descrivere un componente chiamato `my-component` che appartiene a un'applicazione creata su un gruppo di risorse chiamato`my-resource-group`.

```
Get-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Raggruppamento di risorse simili in un componente personalizzato

Ti consigliamo di raggruppare risorse simili, ad esempio istanze di server Web .NET, in componenti personalizzati per facilitare l'integrazione e migliorare il monitoraggio e gli approfondimenti. Attualmente, CloudWatch Application Insights supporta gruppi personalizzati per le istanze EC2.

Per raggruppare le risorse in un componente personalizzato tramite AWS CLI

Per AWS CLI utilizzarlo per raggruppare tre istanze (`arn:aws:ec2:us-east-1:123456789012:instance/i-11111arn:aws:ec2:us-east-1:123456789012:instance/i-22222`, e `arn:aws:ec2:us-east-1:123456789012:instance/i-33333`) in un componente personalizzato chiamato `my-component` per un'applicazione creata per il gruppo di risorse chiamato`my-resource-group`, usa il seguente comando.

```
aws application-insights create-component --resource-group-name my-
resource-group --component-name my-component --resource-list arn:aws:ec2:us-
east-1:123456789012:instance/i-11111 arn:aws:ec2:us-east-1:123456789012:instance/
i-22222 arn:aws:ec2:us-east-1:123456789012:instance/i-33333
```

Per raggruppare le risorse in un componente personalizzato tramite AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell Per raggruppare tre istanze (`arn:aws:ec2:us-east-1:123456789012:instance/i-11111arn:aws:ec2:us-east-1:123456789012:instance/i-22222`, e`arn:aws:ec2:us-east-1:123456789012:instance/i-33333`) in un componente personalizzato chiamato`my-component`, per un'applicazione creata per il gruppo di risorse chiamato`my-resource-group`, utilizzate il comando seguente.

```
New-CWAIComponent -ResourceGroupName my-resource-group -ComponentName my-component
 -ResourceList arn:aws:ec2:us-east-1:123456789012:instance/i-11111,arn:aws:ec2:us-
```

```
east-1:123456789012:instance/i-22222,arn:aws:ec2:us-east-1:123456789012:instance/
i-33333
```

Annullamento del raggruppamento di un componente personalizzato

Per separare un componente personalizzato utilizzando il AWS CLI

Per utilizzare AWS CLI per separare un componente personalizzato denominato `my-component` in un'applicazione creata nel gruppo di risorse`my-resource-group`, utilizzare il comando seguente.

```
aws application-insights delete-component --resource-group-name my-resource-group --
component-name my-new-component
```

Per separare un componente personalizzato utilizzando AWS Tools for Windows PowerShell

Per utilizzare AWS Tools for Windows PowerShell per separare un componente personalizzato denominato `my-component` in un'applicazione creata nel gruppo di risorse`my-resource-group`, utilizzare il comando seguente.

```
Remove-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Aggiornamento di un'applicazione

Aggiornate un'applicazione utilizzando il AWS CLI

È possibile utilizzare AWS CLI per aggiornare un'applicazione, generare AWS Systems Manager OpsCenter OpsItems per i problemi rilevati con l'applicazione e associare la creazione OpsItems all'argomento SNS `arn:aws:sns:us-east-1:123456789012:MyTopic`, utilizzando il comando seguente.

```
aws application-insights update-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Aggiornare un'applicazione utilizzando AWS Tools for Windows PowerShell

È possibile utilizzare il AWS Tools for Windows PowerShell comando seguente per aggiornare un'applicazione OpsCenter OpsItems per generare AWS SSM per i problemi rilevati con l'applicazione e OpsItems per associare l'applicazione creata all'argomento `arn:aws:sns:us-east-1:123456789012:MyTopic` SNS.

```
Update-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Aggiornamento di un componente personalizzato

Aggiornate un componente personalizzato utilizzando il AWS CLI

È possibile utilizzare il AWS CLI per aggiornare un componente personalizzato chiamato `my-component` con un nuovo nome di componente e un gruppo aggiornato di istanze, utilizzando il comando seguente. `my-new-component`

```
aws application-insights update-component --resource-group-name my-resource-
group --component-name my-component --new-component-name my-new-component --
resource-list arn:aws:ec2:us-east-1:123456789012:instance/i-44444 arn:aws:ec2:us-
east-1:123456789012:instance/i-55555
```

Aggiornare un componente personalizzato utilizzando AWS Tools for Windows PowerShell

È possibile AWS Tools for Windows PowerShell utilizzare il comando seguente per aggiornare un componente personalizzato chiamato `my-component` con un nuovo nome di componente e un gruppo aggiornato di istanze. `my-new-component`

```
Update-CWAIComponent -ComponentName my-component -NewComponentName my-new-
component -ResourceGroupName my-resource-group -ResourceList arn:aws:ec2:us-
east-1:123456789012:instance/i-44444,arn:aws:ec2:us-east-1:123456789012:instance/
i-55555
```

Gestione e aggiornamento del monitoraggio

Puoi gestire e aggiornare il monitoraggio per l'applicazione Application Insights utilizzando la riga di comando.

Argomenti

- Elenco dei problemi con l'applicazione
- Descrizione di un problema dell'applicazione
- Descrizione delle anomalie o degli errori associati a un problema
- Descrizione di un'anomalia o di un errore con l'applicazione

- Descrizione delle configurazioni di monitoraggio di un componente
- Descrizione della configurazione di monitoraggio consigliata di un componente
- Aggiornamento delle configurazioni di monitoraggio per un componente
- Rimozione di un gruppo di risorse specificato dal monitoraggio di Application Insights

Elenco dei problemi con l'applicazione

Elenca i problemi relativi all'applicazione utilizzando il AWS CLI

Per utilizzarlo AWS CLI per elencare i problemi con l'applicazione rilevati tra 1.000 e 10.000 millisecondi a partire da Unix Epoch per un'applicazione creata su un gruppo di risorse chiamato`my-resource-group`, usate il comando seguente.

```
aws application-insights list-problems --resource-group-name my-resource-group --start-time 1000 --end-time 10000
```

Elenca i problemi relativi all'applicazione utilizzando Tools for Windows AWS PowerShell

Per utilizzarlo AWS Tools for Windows PowerShell per elencare i problemi con l'applicazione rilevati tra 1.000 e 10.000 millisecondi a partire da Unix Epoch per un'applicazione creata su un gruppo di risorse chiamato`my-resource-group`, usate il comando seguente.

```
$startDate = "8/6/2019 3:33:00"
$endDate = "8/6/2019 3:34:00"
Get-CWAIProblemList -ResourceGroupName my-resource-group -StartTime $startDate -EndTime $endDate
```

Descrizione di un problema dell'applicazione

Descrivete un problema applicativo utilizzando il AWS CLI

Per utilizzare il comando AWS CLI per descrivere un problema con l'id del problema`p-1234567890`, utilizzate il comando seguente.

```
aws application-insights describe-problem –problem-id p-1234567890
```

Descrivete un problema relativo all'applicazione utilizzando AWS Tools for Windows PowerShell

Per utilizzare il comando AWS Tools for Windows PowerShell per descrivere un problema con l'ID del problemap-1234567890, utilizzate il comando seguente.

```
Get-CWAIProblem -ProblemId p-1234567890
```

Descrizione delle anomalie o degli errori associati a un problema

Descrivi le anomalie o gli errori associati a un problema utilizzando il AWS CLI

Per utilizzare il comando AWS CLI per descrivere le anomalie o gli errori associati a un problema con l'ID del problemap-1234567890, utilizzare il comando seguente.

```
aws application-insights describe-problem-observations --problem-id -1234567890
```

Descrizione delle anomalie o degli errori associati a un problema tramite AWS Tools for Windows PowerShell

Per utilizzare il comando AWS Tools for Windows PowerShell per descrivere le anomalie o gli errori associati a un problema con l'id del problemap-1234567890, utilizzare il comando seguente.

```
Get-CWAIProblemObservation -ProblemId p-1234567890
```

Descrizione di un'anomalia o di un errore con l'applicazione

Descrizione di un'anomalia o di un errore con l'applicazione tramite CLI AWS

Per utilizzare il AWS CLI per descrivere un'anomalia o un errore nell'applicazione con l'ID di osservazioneo-1234567890, utilizzare il comando seguente.

```
aws application-insights describe-observation –observation-id o-1234567890
```

Descrivete un'anomalia o un errore dell'applicazione utilizzando AWS Tools for Windows PowerShell

Per utilizzare il comando AWS Tools for Windows PowerShell per descrivere un'anomalia o un errore nell'applicazione con l'ID di osservazioneo-1234567890, utilizzate il comando seguente.

```
Get-CWAIObservation -ObservationId o-1234567890
```

Descrizione delle configurazioni di monitoraggio di un componente

Descrizione delle configurazioni di monitoraggio di un componente tramite AWS CLI

Per utilizzare il comando AWS CLI per descrivere la configurazione di monitoraggio di un componente chiamato `my-component` in un'applicazione creata nel gruppo di risorse`my-resource-group`, utilizzare il comando seguente.

```
aws application-insights describe-component-configuration –resource-group-name my-resource-group –component-name my-component
```

Descrivi le configurazioni di monitoraggio di un componente utilizzando AWS Tools for Windows PowerShell

Per utilizzare il comando AWS Tools for Windows PowerShell per descrivere la configurazione di monitoraggio di un componente chiamato`my-component`, in un'applicazione creata nel gruppo di risorse`my-resource-group`, utilizzare il comando seguente.

```
Get-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-resource-group
```

Per ulteriori informazioni sulla configurazione del componente e, ad esempio, dei file JSON, consulta Utilizzo delle configurazioni dei componenti.

Descrizione della configurazione di monitoraggio consigliata di un componente

Descrivere la configurazione di monitoraggio consigliata di un componente utilizzando il AWS CLI

Quando il componente fa parte di un'applicazione.NET Worker, è possibile utilizzare AWS CLI per descrivere la configurazione di monitoraggio consigliata di un componente chiamato `my-component` in un'applicazione creata nel gruppo di risorse`my-resource-group`, utilizzando il comando seguente.

```
aws application-insights describe-component-configuration-recommendation --resource-group-name my-resource-group --component-name my-component --tier DOT_NET_WORKER
```

Descrivi la configurazione di monitoraggio consigliata di un componente utilizzando AWS Tools for Windows PowerShell

Quando il componente fa parte di un'applicazione.NET Worker, è possibile utilizzare il AWS Tools for Windows PowerShell per descrivere la configurazione di monitoraggio consigliata di un componente chiamato `my-component` in un'applicazione creata nel gruppo di risorse`my-resource-group`, utilizzando il comando seguente.

```
Get-CWAIComponentConfigurationRecommendation -ComponentName my-component -
ResourceGroupName my-resource-group -Tier DOT_NET_WORKER
```

Per ulteriori informazioni sulla configurazione del componente e, ad esempio, dei file JSON, consulta Utilizzo delle configurazioni dei componenti.

Aggiornamento delle configurazioni di monitoraggio per un componente

Aggiornamento delle configurazioni di monitoraggio per un componente tramite AWS CLI

Per utilizzare il AWS CLI per aggiornare il componente chiamato `my-component` in un'applicazione creata sul gruppo di risorse chiamato`my-resource-group`, utilizzare il comando seguente. Il comando include queste operazioni:

1. Abilita il monitoraggio del componente.
2. Impostare il livello del componente su `.NET Worker`.
3. Aggiornare la configurazione JSON del componente per leggere dal file locale `configuration.txt`.

```
aws application-insights update-component-configuration --resource-group-name my-
resource-group --component-name my-component --tier DOT_NET_WORKER --monitor --
component-configuration "file://configuration.txt"
```

Aggiornamento delle configurazioni di monitoraggio per un componente tramite AWS Tools for Windows PowerShell

Per utilizzare il AWS Tools for Windows PowerShell per aggiornare il componente chiamato `my-component` in un'applicazione creata sul gruppo di risorse chiamato`my-resource-group`, utilizzate il comando seguente. Il comando include queste operazioni:

1. Abilita il monitoraggio del componente.
2. Impostare il livello del componente su `.NET Worker`.

3. Aggiornare la configurazione JSON del componente per leggere dal file locale `configuration.txt`.

```
[string]$config = Get-Content -Path configuration.txt
Update-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-resource-group -Tier DOT_NET_WORKER -Monitor 1 -ComponentConfiguration $config
```

Per ulteriori informazioni sulla configurazione del componente e, ad esempio, dei file JSON, consulta Utilizzo delle configurazioni dei componenti.

Rimozione di un gruppo di risorse specificato dal monitoraggio di Application Insights

Rimuovere un gruppo di risorse specificato dal monitoraggio di Application Insights utilizzando il AWS CLI

Per utilizzare il comando AWS CLI per rimuovere un'applicazione creata sul gruppo di risorse richiamato `my-resource-group` dal monitoraggio, utilizzare il comando seguente.

```
aws application-insights delete-application --resource-group-name my-resource-group
```

Rimuovere un gruppo di risorse specificato dal monitoraggio di Application Insights utilizzando il AWS Tools for Windows PowerShell

Per utilizzare il comando AWS Tools for Windows PowerShell per rimuovere un'applicazione creata sul gruppo di risorse richiamato `my-resource-group` dal monitoraggio, utilizzare il comando seguente.

```
Remove-CWAIApplication -ResourceGroupName my-resource-group
```

Configurazione del monitoraggio per i gruppi di disponibilità SQL Always On

1. Creare un'applicazione per il gruppo di risorse con le istanze EC2 SQL HA.

   ```
   aws application-insights create-application --region <REGION> --resource-group-name
     <RESOURCE_GROUP_NAME>
   ```

2. Definisci le istanze EC2 che rappresentano il cluster SQL HA creando un nuovo componente applicazione.

```
aws application-insights create-component --resource-group-name
 "<RESOURCE_GROUP_NAME>" --component-name SQL_HA_CLUSTER --resource-list
  "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_1_ID>"
 "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_2_ID>
```

3. Configura il componente SQL HA.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "SQL_HA_CLUSTER" --
monitor --tier SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP --monitor  --component-
configuration '{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/
sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-
<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\
\MSSQL\\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
```

```
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}'
```

> **Note**
>
> Application Insights deve includere i log eventi dell'applicazione (a livello di informazioni) per rilevare le attività del cluster, ad esempio il failover.

Configurazione del monitoraggio per MySQL RDS

1. Creare un'applicazione per il gruppo di risorse con l'istanza di database MySQL RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Il log degli errori è abilitato per impostazione predefinita. Il log delle query lente può essere abilitato utilizzando gruppi di parametri di dati. Per ulteriori informazioni, consulta Accesso al log delle query lente e al log generale di MySQL.
   - `set slow_query_log = 1`
   - `set log_output = FILE`
3. Esporta i log da monitorare in CloudWatch log. Per ulteriori informazioni, consulta Pubblicazione dei log MySQL nei log. CloudWatch
4. Configurare il componente RDS MySQL.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier DEFAULT --monitor  --component-configuration "{\"alarmMetrics\":
```

```
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logType\":
\"MYSQL\",\"monitor\":true},{\"logType\": \"MYSQL_SLOW_QUERY\",\"monitor\":false}]}"
```

Configurazione del monitoraggio per MySQL EC2

1. Creare un'applicazione per il gruppo di risorse con le istanze EC2 SQL HA.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Il log degli errori è abilitato per impostazione predefinita. Il log delle query lente può essere abilitato utilizzando gruppi di parametri di dati. Per ulteriori informazioni, consulta Accesso al log delle query lente e al log generale di MySQL.
   - `set slow_query_log = 1`
   - `set log_output = FILE`
3. Configurare il componente MySQL EC2.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier MYSQL --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logGroupName
\":\"<UNIQUE_LOG_GROUP_NAME>\",\"logPath\":\"C:\\\\ProgramData\\\\MySQL\\\\MySQL
 Server **\\\\Data\\\\<FILE_NAME>.err\",\"logType\":\"MYSQL\",\"monitor\":true,
\"encoding\":\"utf-8\"}]}"
```

Configurazione del monitoraggio per PostgreSQL RDS

1. Creare un'applicazione per il gruppo di risorse con l'istanza di database PostgreSQL RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. La pubblicazione dei log di PostgreSQL su non è abilitata per impostazione predefinita. CloudWatch Per abilitare il monitoraggio, apri la console RDS e seleziona il database da monitorare. Scegli Modify (Modifica) nell'angolo in alto a destra e seleziona la casella di controllo con il log PostgreSQL. Scegli Continue (Continua) per salvare questa impostazione.
3. I tuoi log PostgreSQL vengono esportati in. CloudWatch

4. Configurare il componente PostgreSQL RDS.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"POSTGRESQL\",
         \"monitor\": true
      }
   ]
}"
```

Configurazione del monitoraggio per PostgreSQL EC2

1. Creare un'applicazione per il gruppo di risorse con l'istanza PostgreSQL EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Configurare il componente PostgreSQL EC2.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier POSTGRESQL --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
```

```
         \"logPath\":\"/var/lib/pgsql/data/log/\",
         \"logType\":\"POSTGRESQL\",
         \"monitor\":true,
         \"encoding\":\"utf-8\"
      }
   ]
}"
```

Configurare il monitoraggio per Oracle RDS

1. Creare un'applicazione per il gruppo di risorse con l'istanza di database OracleL RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. La pubblicazione dei log Oracle su non è abilitata per impostazione predefinita CloudWatch . Per abilitare il monitoraggio, apri la console RDS e seleziona il database da monitorare. Scegli Modify (Modifica) nell'angolo in alto a destra e seleziona le caselle di controllo con i log Alert (Avviso) e Listener. Scegli Continue (Continua) per salvare questa impostazione.
3. I log Oracle vengono esportati in. CloudWatch
4. Configurare il componente Oracle RDS.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"ORACLE_ALERT\",
         \"monitor\": true
      },
      {
         \"logType\": \"ORACLE_LISTENER\",
         \"monitor\": true
      }
```

```
   ]
}"
```

Configurare il monitoraggio per Oracle EC2

1. Creare un'applicazione per il gruppo di risorse con le istanze EC2 Oracle.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Configurare il componente Oracle EC2.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier ORACLE --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/rdbms/*/*/trace\",
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      },
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/tnslsnr/$HOSTNAME/listener/trace/\",
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      }
   ]
}"
```

CloudWatch Eventi e notifiche di Application Insights per i problemi rilevati

Per ogni applicazione aggiunta ad CloudWatch Application Insights, viene pubblicato un CloudWatch evento per i seguenti eventi nella massima misura possibile:

- Creazione problema. Emesso quando CloudWatch Application Insights rileva un nuovo problema.
  - Tipo di Dettaglio: "Problema di Application Insights rilevato"
  - Dettaglio:
    - `problemId`: l'ID del problema rilevato.
    - `region`: La AWS regione in cui è stato creato il problema.
    - `resourceGroupName`: il gruppo di risorse per l'applicazione registrata per la quale il problema è stato rilevato.
    - `status`: lo stato del problema. Lo stato e le definizioni possibili sono i seguenti:
      - `In progress`: è stato identificato un nuovo problema. Il problema continua a ricevere osservazioni.
      - `Recovering`: il problema si sta stabilizzando. Puoi risolvere manualmente il problema quando si trova in questo stato.
      - `Resolved`: il problema è stato risolto. Non ci sono nuove osservazioni relative a questo problema.
      - `Recurring`: il problema è stato risolto nelle ultime 24 ore. È stato riaperto a seguito di ulteriori osservazioni.
    - `severity`: la gravità del problema.
    - `problemUrl`: l'URL della console per il problema.

- Aggiornamento problema. Emesso quando il problema viene aggiornato con una nuova osservazione o quando un'osservazione esistente viene aggiornata e il problema viene successivamente aggiornato; gli aggiornamenti includono una risoluzione o una chiusura del problema.
  - Tipo di dettaglio: "Problema di Application Insights aggiornato"
  - Dettaglio:
    - `problemId`: l'ID del problema creato.
    - `region`: La AWS regione in cui è stato creato il problema.

- `resourceGroupName`: il gruppo di risorse per l'applicazione registrata per la quale il problema è stato rilevato.
- `status`: lo stato del problema.
- `severity`: la gravità del problema.
- `problemUrl`: l'URL della console per il problema.

Come ricevere una notifica per eventi di problema generati da un'applicazione

Dalla CloudWatch console, seleziona Regole in Eventi nel riquadro di navigazione a sinistra. Nella pagina Rules (Regole) seleziona Create rule (Crea regola). Scegli Amazon CloudWatch Application Insights dall'elenco a discesa Service Name e scegli il tipo di evento. Quindi, scegli Add target (Aggiungi destinazione) e seleziona la destinazione e i parametri, ad esempio un argomento SNS o una funzione Lambda.

Azioni tramite AWS Systems Manager. CloudWatch Application Insights offre l'integrazione integrata con Systems Manager OpsCenter. Se si sceglie di utilizzare questa integrazione per l'applicazione, OpsItem viene creata una sulla OpsCenter console per ogni problema rilevato con l'applicazione. Dalla OpsCenter console, è possibile visualizzare informazioni riepilogative sul problema rilevato da CloudWatch Application Insights e scegliere un runbook di Systems Manager Automation per intraprendere azioni correttive o identificare ulteriormente i processi Windows che causano problemi di risorse nell'applicazione.

Osservabilità tra account di Approfondimenti sulle applicazioni

Con CloudWatch l'osservabilità tra più account di Application Insights, puoi monitorare e risolvere i problemi delle applicazioni che si estendono su più account all'interno di una singola regione. AWS

Puoi utilizzare Amazon CloudWatch Observability Access Manager per configurare uno o più AWS account come account di monitoraggio. Fornirai all'account di monitoraggio la possibilità di visualizzare i dati nel tuo account di origine creando un sink all'interno di esso. Il sink viene utilizzato per creare un collegamento dal tuo account di origine al tuo account di monitoraggio. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Risorse obbligatorie

Per il corretto funzionamento dell'osservabilità tra account di CloudWatch Application Insights, assicurati che i seguenti tipi di telemetria siano condivisi tramite Observability Access Manager. CloudWatch

- Applicazioni in Application Insights CloudWatch
- Metriche in Amazon CloudWatch
- Gruppi di log in Amazon CloudWatch Logs
- Tracce in AWS X-Ray

Utilizzo delle configurazioni dei componenti

La configurazione di un componente è un file di testo in formato JSON che descrive le impostazioni di configurazione del componente. Questa sezione mostra un frammento di modello di esempio, descrizioni delle sezioni di configurazione dei componenti e configurazioni di componenti di esempio.

Argomenti

- Frammento del modello di configurazione dei componenti
- Sezioni di configurazione dei componenti
- Esempi di configurazione dei componenti

Frammento del modello di configurazione dei componenti

Il seguente esempio mostra un frammento di modello in formato JSON.

```
{
  "alarmMetrics" : [
    list of alarm metrics
  ],
  "logs" : [
    list of logs
  ],
  "processes" : [
   list of processes
  ],
  "windowsEvents" : [
    list of windows events channels configurations
  ],
  "alarms" : [
    list of CloudWatch alarms
  ],
  "jmxPrometheusExporter": {
    JMX Prometheus Exporter configuration
```

```
  },
  "hanaPrometheusExporter": {
      SAP HANA Prometheus Exporter configuration
  },
  "haClusterPrometheusExporter": {
      HA Cluster Prometheus Exporter configuration
  },
  "netWeaverPrometheusExporter": {
      SAP NetWeaver Prometheus Exporter configuration
  },
  "subComponents" : [
    {
      "subComponentType" : "AWS::EC2::Instance" ...
      component nested instances configuration
    },
    {
      "subComponentType" : "AWS::EC2::Volume" ...
      component nested volumes configuration
    }
  ]
}
```

Sezioni di configurazione dei componenti

La configurazione di un componente include diverse sezioni principali. Le sezioni nella configurazione di un componente possono essere elencate in qualsiasi ordine.

- alarmMetrics (facoltativo)

  Un elenco di parametri da monitorare per il componente. Tutti i tipi di componente possono avere una sezione alarmMetrics.

- logs (facoltativo)

  Un elenco di log da monitorare per il componente. Solo le istanze EC2 possono avere una sezione Logs.

- processes (facoltativo)

  Un elenco di processi da monitorare per il componente. Solo le istanze EC2 possono avere una sezione processes.

- subComponents (facoltativo)

Configurazione dell'istanza nidificata e del volume subComponent per il componente. I seguenti tipi di componenti possono avere istanze nidificate e una sezione subComponents: ELB, ASG, istanze EC2 raggruppate personalizzate e istanze EC2.

- allarmi (facoltativo)

  Un elenco di allarmi da monitorare per il componente. Tutti i tipi di componente possono avere una sezione di allarme.

- WindowsEvents (facoltativo)

  Un elenco di eventi di Windows da monitorare per il componente. Solo le istanze di Windows su EC2 hanno una sezione `windowsEvents`.

- JMX PrometheusExporter (opzionale)

  Configurazione di JMXPromeTheus Exporter.

- hanaPrometheusExporter (opzionale)

  Configurazione SAP HANA Prometheus Exporter.

- haClusterPrometheusEsportatore (opzionale)

  Configurazione HA Cluster Prometheus Exporter.

- netWeaverPrometheusEsportatore (opzionale)

  Configurazione SAP NetWeaver Prometheus Exporter.

- sapAsePrometheusExporter (opzionale)

  Configurazione dell'esportatore Prometheus per SAP ASE.

L'esempio seguente mostra la sintassi del frammento di sezione subComponent in formato JSON.

```
[
  {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [
      list of alarm metrics
    ],
    "logs" : [
      list of logs
    ],
```

```
    "processes": [
      list of processes
    ],
    "windowsEvents" : [
      list of windows events channels configurations
    ]
  },
  {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [
      list of alarm metrics
    ]
  }
]
```

Proprietà della sezione di configurazione dei componenti

Questa sezione le proprietà di ciascuna sezione di configurazione dei componenti.

Sections

- Parametro
- Log
- Processo
- JMX Prometheus Exporter
- HANA Prometheus Exporter
- HA Cluster Prometheus Exporter
- NetWeaver Prometheus Exporter
- Esportatore Prometheus per SAP ASE
- Eventi Windows
- Allarme

Parametro

Definisce un parametro da monitorare per il componente.

JSON

```
{
```

```
  "alarmMetricName" : "monitoredMetricName",
  "monitor" : true/false
}
```

Proprietà

- alarmMetricName (richiesto)

  Il nome del parametro da monitorare per il componente. Per i parametri supportati da Application Insights, consulta Log e parametri supportati da Amazon Application Insights CloudWatch .
- monitor (facoltativo)

  Valore booleano per indicare se monitorare il parametro. Il valore predefinito è `true`.

Log

Definisce un log da monitorare per il componente.

JSON

```
{
  "logGroupName" : "logGroupName",
  "logPath" : "logPath",
  "logType" : "logType",
  "encoding" : "encodingType",
  "monitor" : true/false
}
```

Proprietà

- logGroupName (richiesto)

  Il nome del gruppo di CloudWatch log da associare al registro monitorato. Per i vincoli relativi al nome del gruppo di log, vedere. CreateLogGroup
- LogPath (richiesto per i componenti delle istanze EC2; non richiesto per i componenti che non utilizzano CloudWatch Agent, come) AWS Lambda

  Il percorso dei log da monitorare. Il percorso dei log deve essere un percorso assoluto del file di sistema di Windows. Per ulteriori informazioni, consulta la sezione File di configurazione CloudWatch dell'agente: log.

- logType (obbligatorio)

  Il tipo di log determina i modelli di log in base ai quali Application Insights analizza il log. Il tipo di log è selezionato tra i seguenti:

  - SQL_SERVER
  - MYSQL
  - MYSQL_SLOW_QUERY
  - POSTGRESQL
  - ORACLE_ALERT
  - ORACLE_LISTENER
  - IIS
  - APPLICATION
  - WINDOWS_EVENTS
  - WINDOWS_EVENTS_ACTIVE_DIRECTORY
  - WINDOWS_EVENTS_DNS
  - WINDOWS_EVENTS_IIS
  - WINDOWS_EVENTS_SHAREPOINT
  - SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP
  - SQL_SERVER_FAILOVER_CLUSTER_INSTANCE
  - DEFAULT
  - CUSTOM
  - STEP_FUNCTION
  - API_GATEWAY_ACCESS
  - API_GATEWAY_EXECUTION
  - SAP_HANA_LOGS
  - SAP_HANA_TRACE
  - SAP_HANA_HIGH_AVAILABILITY
  - SAP_NETWEAVER_DEV_TRACE_LOGS
  - PACEMAKER_HIGH_AVAILABILITY

- encoding (facoltativo)

Il tipo di codifica dei log da monitorare. La codifica specificata deve essere inclusa nell'elenco delle codifiche supportate dall'CloudWatch agente. Se non viene fornita, CloudWatch Application Insights utilizza la codifica predefinita di tipo utf-8, ad eccezione di:

- `SQL_SERVER`: codifica utf-16
- `IIS`: codifica ascii

- monitor (facoltativo)

  Valore booleano che indica se monitorare i log. Il valore predefinito è `true`.

Processo

Definisce un processo da monitorare per il componente.

JSON

```
{
  "processName" : "monitoredProcessName",
  "alarmMetrics" : [
      list of alarm metrics
  ]
}
```

Proprietà

- processName (obbligatorio)

  Il nome del processo da monitorare per il componente. Il nome del processo non deve contenere una radice del processo, ad esempio `sqlservr` o `sqlservr.exe`.

- alarmMetrics (obbligatorio)

  Un elenco di parametri da monitorare per questo processo. Per visualizzare le metriche di processo supportate da CloudWatch Application Insights, vedere. Amazon Elastic Compute Cloud (EC2)

JMX Prometheus Exporter

Definisce le impostazioni di JMX Prometheus Exporter.

JSON

```
"JMXPrometheusExporter": {
  "jmxURL" : "JMX URL",
  "hostPort" : "The host and port",
  "prometheusPort" : "Target port to emit Prometheus metrics"
}
```

Proprietà

- jmxURL (facoltativo)

  L'URL JMX completo a cui connettersi.

- hostPort (facoltativo)

  L'host e la porta a cui connettersi tramite il JMX remoto. È possibile specificare solo `jmxURL` o `hostPort`.

- prometheusPort (facoltativo)

  La porta di destinazione a cui inviare i parametri di Prometheus. Se non viene specificata, sarà utilizzata la porta predefinita 9404.

HANA Prometheus Exporter

Definisce le impostazioni di HANA Prometheus Exporter.

JSON

```
"hanaPrometheusExporter": {
    "hanaSid": "SAP HANA  SID",
    "hanaPort": "HANA database port",
    "hanaSecretName": "HANA secret name",
    "prometheusPort": "Target port to emit Prometheus metrics"
}
```

Proprietà

- hanaSid

  L'ID di sistema SAP (SID) a tre caratteri del sistema SAP HANA.

- hanaPort

La porta del database HANA con la quale l'esportatore eseguirà una query sui parametri HANA.

- hanaSecretName

  Il AWS Secrets Manager segreto che memorizza le credenziali degli utenti di monitoraggio HANA. L'HANA Prometheus exporter utilizza queste credenziali per connettersi al database e eseguire query sui parametri HANA.

- prometheusPort (facoltativo)

  La porta di destinazione a cui inviare i parametri di Prometheus. Se non viene specificata, sarà utilizzata la porta di default 9668.

HA Cluster Prometheus Exporter

Definisce le impostazioni di HA Cluster Prometheus Exporter.

JSON

```
"haClusterPrometheusExporter": {
    "prometheusPort": "Target port to emit Prometheus metrics"
}
```

Proprietà

- prometheusPort (facoltativo)

  La porta di destinazione a cui inviare i parametri di Prometheus. Se non viene specificata, sarà utilizzata la porta di default 9664.

NetWeaver Prometheus Exporter

Definisce le impostazioni di NetWeaver Prometheus Exporter.

JSON

```
"netWeaverPrometheusExporter": {
    "sapSid": "SAP NetWeaver  SID",
    "instanceNumbers": [ "Array of instance Numbers of SAP NetWeaver system "],
"prometheusPort": "Target port to emit Prometheus metrics"
```

```
}
```

Proprietà

- sapSid

  L'ID di sistema SAP (SID) a 3 caratteri del sistema SAP. NetWeaver
- instanceNumbers

  Matrice dei numeri di istanza del sistema SAP. NetWeaver

  Esempio: `"instanceNumbers": [ "00", "01"]`
- prometheusPort (facoltativo)

  La porta di destinazione a cui inviare i parametri di Prometheus. Se non viene specificata, sarà utilizzata la porta predefinita `9680`.

Esportatore Prometheus per SAP ASE

Definisce le impostazioni dell'esportatore Prometheus per SAP ASE.

JSON

```
"sapASEPrometheusExporter": {
    "sapAseSid": "SAP ASE SID",
    "sapAsePort": "SAP ASE database port",
    "sapAseSecretName": "SAP ASE secret name",
    "prometheusPort": "Target port to emit Prometheus metrics",
    "agreeToEnableASEMonitoring": true
}
```

Proprietà

- sapAseSid

  L'ID di sistema SAP (SID) a tre caratteri del sistema SAP ASE.
- sapAsePort

  La porta del database ASE con la quale l'esportatore eseguirà una query sui parametri ASE.
- sapAseSecretNome

Il AWS Secrets Manager segreto che memorizza le credenziali degli utenti di monitoraggio ASE. L'esportatore Prometheus di SAP ASE utilizza queste credenziali per connettersi al database ed eseguire query sui parametri ASE.

- prometheusPort (facoltativo)

  La porta di destinazione a cui inviare i parametri di Prometheus. Se non specificata, viene utilizzata la porta predefinita 9399. Se esiste un altro database ASE che utilizza la porta predefinita, viene utilizzata la porta 9499.

Eventi Windows

Definisce gli eventi Windows da registrare.

JSON

```
{
  "logGroupName" : "logGroupName",
  "eventName" : "eventName",
  "eventLevels" : ["ERROR","WARNING","CRITICAL","INFORMATION","VERBOSE"],
  "monitor" : true/false
}
```

Proprietà

- logGroupName (obbligatorio)

  Il nome del gruppo di CloudWatch log da associare al registro monitorato. Per i vincoli relativi al nome del gruppo di log, vedere. CreateLogGroup

- eventName (obbligatorio)

  Il tipo di eventi Windows da registrare. Ciò equivale al nome del canale del log degli eventi Windows. Ad esempio, Sistema, Sicurezza CustomEventName, ecc. Questo campo è obbligatorio per ogni tipo di evento Windows da registrare.

- eventLevels (obbligatorio)

  I livelli dell'evento da registrare. È necessario specificare ciascun livello da registrare. I valori possibili sono `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` e `VERBOSE`. Questo campo è obbligatorio per ogni tipo di evento Windows da registrare.

- monitor (facoltativo)

  Valore booleano che indica se monitorare i log. Il valore predefinito è `true`.

Allarme

Definisce un CloudWatch allarme da monitorare per il componente.

JSON

```
{
  "alarmName" : "monitoredAlarmName",
  "severity" : HIGH/MEDIUM/LOW
}
```

Proprietà

- alarmName (obbligatorio)

  Il nome dell' CloudWatch allarme da monitorare per il componente.
- gravità (facoltativo)

  Indica il grado di interruzione quando l'allarme si spegne.

Esempi di configurazione dei componenti

I seguenti esempi mostrano le configurazioni dei componenti in formato JSON per i servizi pertinenti.

Esempio di configurazioni dei componenti

- Tabella Amazon DynamoDB
- Amazon EC2 Auto Scaling (ASG)
- Cluster Amazon EKS
- Istanza Amazon Elastic Compute Cloud (EC2)
- Amazon Elastic Container Service (Amazon ECS)
- Servizi Amazon ECS
- Attività di Amazon ECS
- Amazon Elastic File System (Amazon EFS)

- Amazon FSx
- Amazon Relational Database Service (RDS) Aurora MySQL
- Istanza Amazon Relational Database Service (RDS)
- Controllo dell'integrità di Amazon Route 53
- Zona ospitata di Amazon Route 53
- Amazon Route 53 Resolver endpoint
- Amazon Route 53 Resolver configurazione della registrazione delle interrogazioni
- Bucket Amazon S3
- Amazon Simple Queue Service (SQS)
- Argomento Amazon SNS
- Amazon Virtual Private Cloud (Amazon VPC) (Amazon VPC)
- Gateway Network Address Translation (NAT) di Amazon VPC
- Fasi REST API di API Gateway
- Application Elastic Load Balancing
- Funzione AWS Lambda
- AWS Network Firewall gruppo di regole
- AWS Network Firewall associazione di gruppi di regole
- AWS Step Functions
- Istanze Amazon EC2 raggruppate per clienti
- Sistema di bilanciamento del carico elastico
- Java
- Kubernetes su Amazon EC2
- RDS MariaDB e RDS MySQL
- RDS Oracle
- RDS PostgreSQL
- SAP ASE su Amazon EC2
- SAP ASE High Availability su Amazon EC2
- SAP HANA su Amazon EC2
- Alta disponibilità SAP HANA su Amazon EC2
- SAP NetWeaver su Amazon EC2

- NetWeaver Alta disponibilità SAP su Amazon EC2
- Gruppo di disponibilità SQL Always On
- Istanza cluster di failover SQL

Tabella Amazon DynamoDB

Il seguente esempio mostra una configurazione di componente in formato JSON per la tabella Amazon DynamoDB.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "SystemErrors",
      "monitor": false
    },
    {
      "alarmMetricName": "UserErrors",
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedReadCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedWriteCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ReadThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "WriteThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "ConditionalCheckFailedRequests",
      "monitor": false
    },
    {
      "alarmMetricName": "TransactionConflict",
      "monitor": false
```

```
    }
  ],
  "logs": []
}
```

Amazon EC2 Auto Scaling (ASG)

Il seguente esempio mostra una configurazione di componente in formato JSON per (ASG) di Amazon EC2 Auto Scaling .

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUCreditBalance"
      }, {
        "alarmMetricName" : "EBSIOBalance%"
      }
    ],
    "subComponents" : [
      {
        "subComponentType" : "AWS::EC2::Instance",
        "alarmMetrics" : [
          {
            "alarmMetricName" : "CPUUtilization"
          }, {
            "alarmMetricName" : "StatusCheckFailed"
          }
        ],
        "logs" : [
          {
            "logGroupName" : "my_log_group",
            "logPath" : "C:\\LogFolder\\*",
            "logType" : "APPLICATION"
          }
        ],
        "processes" : [
          {
            "processName" : "my_process",
            "alarmMetrics" : [
              {
                  "alarmMetricName" : "procstat cpu_usage",
                  "monitor" : true
              }, {
```

```
                "alarmMetricName" : "procstat memory_rss",
                "monitor" : true
            }
        ]
    }
],
      "windowsEvents" : [
        {
          "logGroupName" : "my_log_group_2",
          "eventName" : "Application",
          "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ]
        }
      ]
    }, {
      "subComponentType" : "AWS::EC2::Volume",
      "alarmMetrics" : [
        {
          "alarmMetricName" : "VolumeQueueLength"
        }, {
          "alarmMetricName" : "BurstBalance"
        }
      ]
    }
  ],
  "alarms" : [
    {
      "alarmName" : "my_asg_alarm",
      "severity" : "LOW"
    }
  ]
}
```

Cluster Amazon EKS

Il seguente esempio mostra una configurazione di componente in formato JSON per il cluster Amazon EKS.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName": "cluster_failed_node_count",
          "monitor":true
       },
```

```
        {
           "alarmMetricName": "node_cpu_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName": "node_cpu_utilization",
           "monitor":true
        },
        {
           "alarmMetricName": "node_filesystem_utilization",
           "monitor":true
        },
        {
           "alarmMetricName": "node_memory_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName": "node_memory_utilization",
           "monitor":true
        },
        {
           "alarmMetricName": "node_network_total_bytes",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_cpu_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_cpu_utilization",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_cpu_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_memory_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_memory_utilization",
           "monitor":true
        },
```

```
        {
           "alarmMetricName": "pod_memory_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_network_rx_bytes",
           "monitor":true
        },
        {
           "alarmMetricName": "pod_network_tx_bytes",
           "monitor":true
        }
     ],
     "logs":[
        {
           "logGroupName": "/aws/containerinsights/kubernetes/application",
           "logType":"APPLICATION",
           "monitor":true,
           "encoding":"utf-8"
        }
     ],
     "subComponents":[
        {
           "subComponentType":"AWS::EC2::Instance",
           "alarmMetrics":[
              {
                 "alarmMetricName":"CPUUtilization",
                 "monitor":true
              },
              {
                 "alarmMetricName":"StatusCheckFailed",
                 "monitor":true
              },
              {
                 "alarmMetricName":"disk_used_percent",
                 "monitor":true
              },
              {
                 "alarmMetricName":"mem_used_percent",
                 "monitor":true
              }
           ],
           "logs":[
              {
```

```
            "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
            "logPath":"",
            "logType":"APPLICATION",
            "monitor":true,
            "encoding":"utf-8"
         }
      ],
      "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ],
      "windowsEvents":[
         {
            "logGroupName":"my_log_group_2",
            "eventName":"Application",
            "eventLevels":[
               "ERROR",
               "WARNING",
               "CRITICAL"
            ],
            "monitor":true
         }
      ]
   },
   {
      "subComponentType":"AWS::AutoScaling::AutoScalingGroup",
      "alarmMetrics":[
         {
            "alarmMetricName":"CPUCreditBalance",
            "monitor":true
         },
         {
            "alarmMetricName":"EBSIOBalance%",
            "monitor":true
```

```
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Volume",
         "alarmMetrics":[
            {
               "alarmMetricName":"VolumeReadBytes",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeWriteBytes",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeReadOps",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeWriteOps",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeQueueLength",
               "monitor":true
            },
            {
               "alarmMetricName":"BurstBalance",
               "monitor":true
            }
         ]
      }
   ]
}
```

Note

- La sezione `subComponents` di `AWS::EC2::Instance`, `AWS::EC2::Volume` e `AWS::AutoScaling::AutoScalingGroup` si applica solo ai cluster di Amazon EKS in esecuzione sul tipo di avvio EC2.

- La sezione `windowsEvents` di `AWS::EC2::Instance` in `subComponents` si applica solo a Windows in esecuzione su istanze Amazon EC2.

Istanza Amazon Elastic Compute Cloud (EC2)

Il seguente esempio mostra una configurazione di componente in formato JSON per un'istanza Amazon EC2.

Important

Quando un'istanza Amazon EC2 entra in uno stato `stopped`, viene rimossa dal monitoraggio. Quando torna a uno `running` stato, viene aggiunto all'elenco dei componenti non monitorati nella pagina dei dettagli dell'applicazione della console di CloudWatch Application Insights. Se è abilitato il monitoraggio automatico di nuove risorse per l'applicazione, l'istanza viene aggiunta all'elenco diComponenti monitorati. Tuttavia, i registri e i parametri sono impostati sul valore predefinito per il carico di lavoro. La configurazione precedente del log e dei parametri non viene salvata.

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUUtilization",
        "monitor" : true
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
        "logType" : "APPLICATION",
        "monitor" : true
      },
      {
        "logGroupName" : "my_log_group_2",
        "logPath" : "C:\\LogFolder2\\*",
        "logType" : "IIS",
```

```
        "encoding" : "utf-8"
      }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ],
    "windowsEvents" : [
      {
        "logGroupName" : "my_log_group_3",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
      }, {
        "logGroupName" : "my_log_group_4",
        "eventName" : "System",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
    }],
     "alarms" : [
      {
        "alarmName" : "my_instance_alarm_1",
        "severity" : "HIGH"
      },
      {
        "alarmName" : "my_instance_alarm_2",
        "severity" : "LOW"
      }
    ],
     "subComponents" : [
     {
       "subComponentType" : "AWS::EC2::Volume",
       "alarmMetrics" : [
       {
```

```
        "alarmMetricName" : "VolumeQueueLength",
        "monitor" : "true"
      },
      {
        "alarmMetricName" : "VolumeThroughputPercentage",
        "monitor" : "true"
      },
      {
        "alarmMetricName" : "BurstBalance",
        "monitor" : "true"
      }
    }]
  }
```

Amazon Elastic Container Service (Amazon ECS)

Il seguente esempio mostra una configurazione di componente in formato JSON per Amazon Elastic Container Service (Amazon ECS).

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CpuUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkRxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkTxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"RunningTaskCount",
          "monitor":true
       },
       {
          "alarmMetricName":"PendingTaskCount",
```

```
            "monitor":true
         },
         {
            "alarmMetricName":"StorageReadBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageWriteBytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/ecs/my-task-definition",
            "logType":"APPLICATION",
            "monitor":true
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
            "alarmMetrics":[
               {
                  "alarmMetricName":"HTTPCode_Backend_4XX",
                  "monitor":true
               },
               {
                  "alarmMetricName":"HTTPCode_Backend_5XX",
                  "monitor":true
               },
               {
                  "alarmMetricName":"Latency",
                  "monitor":true
               },
               {
                  "alarmMetricName":"SurgeQueueLength",
                  "monitor":true
               },
               {
                  "alarmMetricName":"UnHealthyHostCount",
                  "monitor":true
               }
            ]
         },
```

```
        {
           "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
           "alarmMetrics":[
              {
                 "alarmMetricName":"HTTPCode_Target_4XX_Count",
                 "monitor":true
              },
              {
                 "alarmMetricName":"HTTPCode_Target_5XX_Count",
                 "monitor":true
              },
              {
                 "alarmMetricName":"TargetResponseTime",
                 "monitor":true
              },
              {
                 "alarmMetricName":"UnHealthyHostCount",
                 "monitor":true
              }
           ]
        },
        {
           "subComponentType":"AWS::EC2::Instance",
           "alarmMetrics":[
              {
                 "alarmMetricName":"CPUUtilization",
                 "monitor":true
              },
              {
                 "alarmMetricName":"StatusCheckFailed",
                 "monitor":true
              },
              {
                 "alarmMetricName":"disk_used_percent",
                 "monitor":true
              },
              {
                 "alarmMetricName":"mem_used_percent",
                 "monitor":true
              }
           ],
           "logs":[
              {
                 "logGroupName":"my_log_group",
```

```
                "logPath":"/mylog/path",
                "logType":"APPLICATION",
                "monitor":true
             }
          ],
          "processes" : [
             {
                "processName" : "my_process",
                "alarmMetrics" : [
                    {
                        "alarmMetricName" : "procstat cpu_usage",
                        "monitor" : true
                    }, {
                        "alarmMetricName" : "procstat memory_rss",
                        "monitor" : true
                    }
                ]
             }
          ],
          "windowsEvents":[
             {
                "logGroupName":"my_log_group_2",
                "eventName":"Application",
                "eventLevels":[
                   "ERROR",
                   "WARNING",
                   "CRITICAL"
                ],
                "monitor":true
             }
          ]
       },
       {
          "subComponentType":"AWS::EC2::Volume",
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":"true"
             },
             {
                "alarmMetricName":"VolumeThroughputPercentage",
                "monitor":"true"
             },
             {
```

```
                "alarmMetricName":"BurstBalance",
                "monitor":"true"
             }
          ]
       }
    ]
 }
```

> **Note**
>
> - La sezione `subComponents` di `AWS::EC2::Instance` e `AWS::EC2::Volume` si applica solo ai cluster Amazon ECS con servizio ECS o attività ECS in esecuzione sul tipo di avvio EC2.
> - La sezione `windowsEvents` di `AWS::EC2::Instance` in `subComponents` si applica solo a Windows in esecuzione su istanze Amazon EC2.

Servizi Amazon ECS

I seguenti esempi mostrano le configurazioni dei componenti in formato JSON per un servizio Amazon ECS.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CPUUtilization",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilization",
          "monitor":true
       },
       {
          "alarmMetricName":"CpuUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilized",
          "monitor":true
       },
       {
```

```
            "alarmMetricName":"NetworkRxBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"NetworkTxBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"RunningTaskCount",
            "monitor":true
         },
         {
            "alarmMetricName":"PendingTaskCount",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageReadBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageWriteBytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/ecs/my-task-definition",
            "logType":"APPLICATION",
            "monitor":true
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
            "alarmMetrics":[
               {
                  "alarmMetricName":"HTTPCode_Backend_4XX",
                  "monitor":true
               },
               {
                  "alarmMetricName":"HTTPCode_Backend_5XX",
                  "monitor":true
               },
               {
```

```
                "alarmMetricName":"Latency",
                "monitor":true
            },
            {
                "alarmMetricName":"SurgeQueueLength",
                "monitor":true
            },
            {
                "alarmMetricName":"UnHealthyHostCount",
                "monitor":true
            }
        ]
    },
    {
        "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
        "alarmMetrics":[
            {
                "alarmMetricName":"HTTPCode_Target_4XX_Count",
                "monitor":true
            },
            {
                "alarmMetricName":"HTTPCode_Target_5XX_Count",
                "monitor":true
            },
            {
                "alarmMetricName":"TargetResponseTime",
                "monitor":true
            },
            {
                "alarmMetricName":"UnHealthyHostCount",
                "monitor":true
            }
        ]
    },
    {
        "subComponentType":"AWS::EC2::Instance",
        "alarmMetrics":[
            {
                "alarmMetricName":"CPUUtilization",
                "monitor":true
            },
            {
                "alarmMetricName":"StatusCheckFailed",
                "monitor":true
```

```
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"my_log_group",
               "logPath":"/mylog/path",
               "logType":"APPLICATION",
               "monitor":true
            }
         ],
         "processes" : [
            {
               "processName" : "my_process",
               "alarmMetrics" : [
               {
                   "alarmMetricName" : "procstat cpu_usage",
                   "monitor" : true
               }, {
                   "alarmMetricName" : "procstat memory_rss",
                   "monitor" : true
               }
           ]
       }
    ],
         "windowsEvents":[
            {
               "logGroupName":"my_log_group_2",
               "eventName":"Application",
               "eventLevels":[
                  "ERROR",
                  "WARNING",
                  "CRITICAL"
               ],
               "monitor":true
            }
         ]
```

```
      },
      {
         "subComponentType":"AWS::EC2::Volume",
         "alarmMetrics":[
            {
               "alarmMetricName":"VolumeQueueLength",
               "monitor":"true"
            },
            {
               "alarmMetricName":"VolumeThroughputPercentage",
               "monitor":"true"
            },
            {
               "alarmMetricName":"BurstBalance",
               "monitor":"true"
            }
         ]
      }
   ]
}
```

> **Note**
>
> - La sezione `subComponents` di `AWS::EC2::Instance` e `AWS::EC2::Volume` si applica solo ad Amazon ECS in esecuzione sul tipo di avvio EC2.
> - La sezione `windowsEvents` di `AWS::EC2::Instance` in `subComponents` si applica solo a Windows in esecuzione su istanze Amazon EC2.

Attività di Amazon ECS

Il seguente esempio mostra una configurazione di componente in formato JSON per un'attività Amazon ECS.

```
{
    "logs":[
       {
          "logGroupName":"/ecs/my-task-definition",
          "logType":"APPLICATION",
          "monitor":true
       }
```

```
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ]
}
```

Amazon Elastic File System (Amazon EFS)

Il seguente esempio mostra una configurazione di componente in formato JSON per Amazon EFS.

```
{
   "alarmMetrics": [
     {
       "alarmMetricName": "BurstCreditBalance",
       "monitor": true
     },
     {
       "alarmMetricName": "PercentIOLimit",
       "monitor": true
     },
     {
       "alarmMetricName": "PermittedThroughput",
       "monitor": true
     },
     {
       "alarmMetricName": "MeteredIOBytes",
       "monitor": true
     },
     {
       "alarmMetricName": "TotalIOBytes",
       "monitor": true
     },
     {
```

```
        "alarmMetricName": "DataWriteIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "DataReadIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "MetadataIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "ClientConnections",
        "monitor": true
      },
      {
        "alarmMetricName": "TimeSinceLastSync",
        "monitor": true
      },
      {
        "alarmMetricName": "Throughput",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentageOfPermittedThroughputUtilization",
        "monitor": true
      },
      {
        "alarmMetricName": "ThroughputIOPS",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentThroughputDataReadIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentThroughputDataWriteIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentageOfIOPSDataReadIOBytes",
        "monitor": true
      },
      {
```

```
      "alarmMetricName": "PercentageOfIOPSDataWriteIOBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "AverageDataReadIOBytesSize",
      "monitor": true
    },
    {
      "alarmMetricName": "AverageDataWriteIOBytesSize",
      "monitor": true
    }
  ],
  "logs": [
   {
   "logGroupName": "/aws/efs/utils",
   "logType": "EFS_MOUNT_STATUS",
   "monitor": true,
   }
  ]
}
```

Amazon FSx

Il seguente esempio mostra una configurazione di componente in formato JSON per Amazon FSx.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "DataReadBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataReadOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteOperations",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "MetadataOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "FreeStorageCapacity",
      "monitor": true
    }
  ]
}
```

Amazon Relational Database Service (RDS) Aurora MySQL

Il seguente esempio mostra una configurazione di componente in formato JSON per Amazon RDS Aurora MySQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    },
    {
      "alarmMetricName": "CommitLatency",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
}
```

Istanza Amazon Relational Database Service (RDS)

Il seguente esempio mostra una configurazione di componente in formato JSON per un'istanza Amazon RDS.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "BurstBalance",
      "monitor" : true
    }, {
      "alarmMetricName" : "WriteThroughput",
      "monitor" : false
    }
  ],

  "alarms" : [
    {
      "alarmName" : "my_rds_instance_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Controllo dell'integrità di Amazon Route 53

Il seguente esempio mostra una configurazione del componente in formato JSON per i controlli dell'integrità di Amazon Route 53.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ChildHealthCheckHealthyCount",
      "monitor": true
    },
    {
      "alarmMetricName": "ConnectionTime",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckPercentageHealthy",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckStatus",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "SSLHandshakeTime",
      "monitor": true
    },
    {
      "alarmMetricName": "TimeToFirstByte",
      "monitor": true
    }
  ]
}
```

Zona ospitata di Amazon Route 53

Il seguente esempio mostra una configurazione del componente in formato JSON per la zona ospitata di Amazon Route 53.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "DNSQueries",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECInternalFailure",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeysNeedingAction",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyMaxNeedingActionAge",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyAge",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logGroupName":"/hosted-zone/logs",
      "logType": "ROUTE53_DNS_PUBLIC_QUERY_LOGS",
      "monitor": true
```

```
    }
  ]
}
```

Amazon Route 53 Resolver endpoint

L'esempio seguente mostra una configurazione dei componenti in formato JSON per Amazon Route 53 Resolver endpoint.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "EndpointHealthyENICount",
      "monitor": true
    },
    {
      "alarmMetricName": "EndpointUnHealthyENICount",
      "monitor": true
    },
    {
      "alarmMetricName": "InboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryAggregateVolume",
      "monitor": true
    }
  ]
}
```

Amazon Route 53 Resolver configurazione della registrazione delle interrogazioni

Il seguente esempio mostra una configurazione del componente in formato JSON per la configurazione della registrazione delle query Amazon Route 53 Resolver .

```
{
  "logs": [
    {
```

```
      "logGroupName": "/resolver-query-log-config/logs",
      "logType": "ROUTE53_RESOLVER_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Bucket Amazon S3

I seguenti esempi mostrano le configurazioni dei componenti in formato JSON per il bucket Amazon S3.

```
{
    "alarmMetrics" : [
        {
            "alarmMetricName" : "ReplicationLatency",
            "monitor" : true
        }, {
            "alarmMetricName" : "5xxErrors",
            "monitor" : true
        }, {
            "alarmMetricName" : "BytesDownloaded"
            "monitor" : true
        }
    ]
}
```

Amazon Simple Queue Service (SQS)

I seguenti esempi mostrano una configurazione del componente in formato JSON per il servizio Amazon Simple Queue.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "ApproximateAgeOfOldestMessage"
    }, {
      "alarmMetricName" : "NumberOfEmptyReceives"
    }
  ],
  "alarms" : [
    {
```

```
      "alarmName" : "my_sqs_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Argomento Amazon SNS

Il seguente esempio mostra una configurazione di componente in formato JSON per l'argomento Amazon SNS.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NumberOfNotificationsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-InvalidAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-NoMessageAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFailedToRedriveToDlq",
      "monitor": true
    }
  ]
}
```

Amazon Virtual Private Cloud (Amazon VPC) (Amazon VPC)

Il seguente esempio mostra una configurazione del componente in formato JSON per Amazon VPC.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NetworkAddressUsage",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "NetworkAddressUsagePeered",
      "monitor": true
    },
    {
      "alarmMetricName": "VPCFirewallQueryVolume",
      "monitor": true
    }
  ]
}
```

Gateway Network Address Translation (NAT) di Amazon VPC

Il seguente esempio mostra una configurazione del componenti in formato JSON per i gateway NAT.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ErrorPortAllocation",
      "monitor": true
    },
    {
      "alarmMetricName": "IdleTimeoutCount",
      "monitor": true
    }
  ]
}
```

Fasi REST API di API Gateway

L'esempio seguente mostra una configurazione di componente in formato JSON per gli stadi REST API di API Gateway.

```
{
     "alarmMetrics" : [
         {
             "alarmMetricName" : "4XXError",
             "monitor" : true
         },
         {
             "alarmMetricName" : "5XXError",
             "monitor" : true
         }
     ],
```

```
    "logs" : [
        {
            "logType" : "API_GATEWAY_EXECUTION",
            "monitor" : true
        },
        {
            "logType" : "API_GATEWAY_ACCESS",
            "monitor" : true
        }
    ]
}
```

Application Elastic Load Balancing

Il seguente esempio mostra una configurazione di componente in formato JSON per Application Elastic Load Balancing.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ActiveConnectionCount",
    }, {
      "alarmMetricName": "TargetResponseTime"
    }
  ],
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
        }, {
          "alarmMetricName": "StatusCheckFailed"
        }
      ],
      "logs": [
        {
          "logGroupName": "my_log_group",
          "logPath": "C:\\LogFolder\\*",
          "logType": "APPLICATION",
        }
      ],
      "windowsEvents": [
```

```
        {
          "logGroupName": "my_log_group_2",
          "eventName": "Application",
          "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
        }
      ]
    }, {
      "subComponentType": "AWS::EC2::Volume",
      "alarmMetrics": [
        {
          "alarmMetricName": "VolumeQueueLength",
        }, {
          "alarmMetricName": "BurstBalance"
        }
      ]
    }
  ],

  "alarms": [
    {
      "alarmName": "my_alb_alarm",
      "severity": "LOW"
    }
  ]
}
```

Funzione AWS Lambda

Il seguente esempio mostra una configurazione di componente in formato JSON per la funzione AWS Lambda .

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "Errors",
      "monitor": true
    },
    {
      "alarmMetricName": "Throttles",
      "monitor": true
    },
    {
      "alarmMetricName": "IteratorAge",
```

```
      "monitor": true
    },
    {
      "alarmMetricName": "Duration",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "DEFAULT",
      "monitor": true
    }
  ]
}
```

AWS Network Firewall gruppo di regole

Il seguente esempio mostra una configurazione del componente in formato JSON per il gruppo di regole AWS Network Firewall .

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

AWS Network Firewall associazione di gruppi di regole

I seguenti esempi mostrano una configurazione del componente in formato JSON per l'associazione del gruppo di regole AWS Network Firewall .

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

AWS Step Functions

I seguenti esempi mostrano le configurazioni dei componenti in formato JSON per AWS Step Functions.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ExecutionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "LambdaFunctionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "ProvisionedRefillRate",
      "monitor": true
    }
  ],
  "logs": [
    {
     "logGroupName": "/aws/states/HelloWorld-Logs",
      "logType": "STEP_FUNCTION",
      "monitor": true,
    }
  ]
}
```

Istanze Amazon EC2 raggruppate per clienti

L'esempio seguente mostra una configurazione di componenti in formato JSON per le istanze Amazon EC2 raggruppate per cliente.

```
{
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization",
          },
          {
```

```
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION",
          }
        ],
        "processes": [
            {
                "processName": "my_process",
                "alarmMetrics": [
                    {
                        "alarmMetricName": "procstat cpu_usage",
                        "monitor": true
                    }, {
                        "alarmMetricName": "procstat memory_rss",
                        "monitor": true
                    }
                ]
            }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength",
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
      }
    ],
    "alarms": [
      {
```

```
        "alarmName": "my_alarm",
        "severity": "MEDIUM"
      }
    ]
  }
```

Sistema di bilanciamento del carico elastico

Il seguente esempio mostra una configurazione di componente in formato JSON per Elastic Load Balancing.

```
{
    "alarmMetrics": [
      {
        "alarmMetricName": "EstimatedALBActiveConnectionCount"
      }, {
        "alarmMetricName": "HTTPCode_Backend_5XX"
      }
    ],
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization"
          }, {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION"
          }
        ],
        "processes": [
          {
            "processName": "my_process",
            "alarmMetrics": [
                {
                  "alarmMetricName": "procstat cpu_usage",
                  "monitor": true
```

```
                }, {
                  "alarmMetricName": "procstat memory_rss",
                  "monitor": true
                }
              ]
          }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ],
            "monitor": true
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength"
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
      }
    ],

    "alarms": [
      {
        "alarmName": "my_elb_alarm",
        "severity": "HIGH"
      }
    ]
  }
```

Java

I seguenti esempi mostrano le configurazioni dei componenti in formato JSON per Java.

```
{
  "alarmMetrics": [ {
    "alarmMetricName": "java_lang_threading_threadcount",
    "monitor": true
  },
```

```
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_used",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_committed",
    "monitor": true
  }],
  "logs": [ ],
  "JMXPrometheusExporter": {
      "hostPort": "8686",
      "prometheusPort": "9404"
  }
}
```

> **Note**
>
> Application Insights non supporta la configurazione dell'autenticazione per Prometheus JMX Exporter. Per informazioni su come configurare l'autenticazione, consulta la Configurazione di esempio di Prometheus JMX Exporter..

Kubernetes su Amazon EC2

L'esempio seguente mostra una configurazione di componente in formato JSON per Kubernetes su Amazon EC2.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_utilization",
          "monitor":true
       },
       {
```

```
            "alarmMetricName":"node_filesystem_utilization",
            "monitor":true
        },
        {
            "alarmMetricName":"node_memory_reserved_capacity",
            "monitor":true
        },
        {
            "alarmMetricName":"node_memory_utilization",
            "monitor":true
        },
        {
            "alarmMetricName":"node_network_total_bytes",
            "monitor":true
        },
        {
            "alarmMetricName":"pod_cpu_reserved_capacity",
            "monitor":true
        },
        {
            "alarmMetricName":"pod_cpu_utilization",
            "monitor":true
        },
        {
            "alarmMetricName":"pod_cpu_utilization_over_pod_limit",
            "monitor":true
        },
        {
            "alarmMetricName":"pod_memory_reserved_capacity",
            "monitor":true
        },
        {
            "alarmMetricName":"pod_memory_utilization",
            "monitor":true
        },
        {
            "alarmMetricName":"pod_memory_utilization_over_pod_limit",
            "monitor":true
        },
        {
            "alarmMetricName":"pod_network_rx_bytes",
            "monitor":true
        },
        {
```

```
            "alarmMetricName":"pod_network_tx_bytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/aws/containerinsights/kubernetes/application",
            "logType":"APPLICATION",
            "monitor":true,
            "encoding":"utf-8"
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::EC2::Instance",
            "alarmMetrics":[
               {
                  "alarmMetricName":"CPUUtilization",
                  "monitor":true
               },
               {
                  "alarmMetricName":"StatusCheckFailed",
                  "monitor":true
               },
               {
                  "alarmMetricName":"disk_used_percent",
                  "monitor":true
               },
               {
                  "alarmMetricName":"mem_used_percent",
                  "monitor":true
               }
            ],
            "logs":[
               {
                  "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
                  "logPath":"",
                  "logType":"APPLICATION",
                  "monitor":true,
                  "encoding":"utf-8"
               }
            ],
            "processes" : [
              {
```

```
                "processName" : "my_process",
                "alarmMetrics" : [
                    {
                        "alarmMetricName" : "procstat cpu_usage",
                        "monitor" : true
                    }, {
                        "alarmMetricName" : "procstat memory_rss",
                        "monitor" : true
                    }
                ]
            }
        ]
       },
       {
          "subComponentType":"AWS::EC2::Volume",
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeReadBytes",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeWriteBytes",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeReadOps",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeWriteOps",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":true
             },
             {
                "alarmMetricName":"BurstBalance",
                "monitor":true
             }
          ]
       }
    ]
```

```
 }
```

RDS MariaDB e RDS MySQL

I seguenti esempi mostrano le configurazioni dei componenti in formato JSON per RDS MariaDB e RDS MySQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
}
```

RDS Oracle

Il seguente esempio mostra una configurazione di componente in formato JSON per RDS Oracle.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "ORACLE_ALERT",
      "monitor": true,
    },
    {
```

```
      "logType": "ORACLE_LISTENER",
      "monitor": false
    }
  ]
}
```

RDS PostgreSQL

I seguenti esempi mostrano le configurazioni dei componenti in formato JSON per RDS PostgreSQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "POSTGRESQL",
      "monitor": true
    }
  ]
}
```

SAP ASE su Amazon EC2

L'esempio seguente mostra una configurazione di componenti in formato JSON per SAP ASE su Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_log_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_locked_login",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_data_cache_hit_ratio",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
```

```
  ],
  "sapAsePrometheusExporter": {
    "sapAseSid": "ASE",
    "sapAsePort": "4901",
    "sapAseSecretName": "ASE_DB_CREDS",
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

SAP ASE High Availability su Amazon EC2

L'esempio seguente mostra una configurazione di componenti in formato JSON per SAP ASE High Availability su Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "asedb_ha_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_mode",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_latency_in_minutes",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_REP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/repservername/repservername.log",
          "logType": "SAP_ASE_REP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_RMA_AGENT_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/RMA-*/instances/AgentContainer/logs/",
          "logType": "SAP_ASE_RMA_AGENT_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_FAULT_MANAGER_LOGS-my-resource-group",
```

```
          "logPath": "/opt/sap/FaultManager/dev_sybdbfm",
          "logType": "SAP_ASE_FAULT_MANAGER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
  ],
  "sapAsePrometheusExporter": {
    "sapAseSid": "ASE",
    "sapAsePort": "4901",
    "sapAseSecretName": "ASE_DB_CREDS",
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

SAP HANA su Amazon EC2

L'esempio seguente mostra una configurazione di componenti in formato JSON per SAP HANA su Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_max_trigger_read_ratio_percent",
          "monitor": true
```

```
        },
        {
          "alarmMetricName": "hanadb_table_allocation_limit_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_cpu_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_plan_cache_hit_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_last_data_backup_age_days",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_LOGS-my-resource-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.log",
          "logType": "SAP_HANA_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  }
}
```

Alta disponibilità SAP HANA su Amazon EC2

L'esempio seguente mostra una configurazione di componenti in formato JSON per SAP HANA High Availability su Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_stonith_enabled",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_HIGH_AVAILABILITY-my-resource-group",
          "logPath": "/var/log/pacemaker/pacemaker.log",
          "logType": "SAP_HANA_HIGH_AVAILABILITY",
          "monitor": true,
```

```
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  },
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  }
}
```

SAP NetWeaver su Amazon EC2

L'esempio seguente mostra una configurazione dei componenti in formato JSON per SAP NetWeaver su Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
          "monitor": true
        },
        {
          "alarmMetricName": "StatusCheckFailed",
          "monitor": true
        },
        {
          "alarmMetricName": "disk_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "mem_used_percent",
          "monitor": true
        },
        {
```

```
            "alarmMetricName": "sap_alerts_ResponseTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_ResponseTimeDialog",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_DBRequestTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_LongRunners",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_AbortedJobs",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_BasisSystem",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Database",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Security",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_System",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_QueueTime",
            "monitor": true
          },
          {
```

```
          "alarmMetricName": "sap_alerts_Availability",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_start_service_processes",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_dispatcher_queue_now",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_dispatcher_queue_max",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_max",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_now",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-ML4",
          "logPath": "/usr/sap/ML4/*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "netWeaverPrometheusExporter": {
```

```
    "sapSid": "ML4",
    "instanceNumbers": [
      "00",
      "11"
    ],
    "prometheusPort": "9680"
  }
}
```

NetWeaver Alta disponibilità SAP su Amazon EC2

L'esempio seguente mostra una configurazione dei componenti in formato JSON per SAP NetWeaver High Availability su Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "ha_cluster_corosync_ring_errors",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_fail_count",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_HA_check_failover_config_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_HA_get_failover_config_HAActive",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_AbortedJobs",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Availability",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "sap_alerts_BasisSystem",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_DBRequestTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Database",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_FrontendResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_LongRunners",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_QueueTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialog",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Security",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Shortdumps",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "sap_alerts_SqlError",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_System",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_start_service_processes",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-PR1",
          "logPath": "/usr/sap/<SID>/D*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  },
  "netWeaverPrometheusExporter": {
    "sapSid": "PR1",
    "instanceNumbers": [
      "11",
      "12"
    ],
    "prometheusPort": "9680"
  }
}
```

Gruppo di disponibilità SQL Always On

Nell'esempio seguente viene illustrata una configurazione di componente in formato JSON per SQL Always On Availability Group.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
```

```
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
```

```
  } ]
}
```

Istanza cluster di failover SQL

Il seguente esempio mostra una configurazione di componente in formato JSON per l'istanza di cluster di failover SQL.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Normal Messages Queue Length/sec",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "Urgent Message Queue Length/se",
      "monitor" : true
    }, {
      "alarmMetricName" : "Reconnect Count",
      "monitor" : true
    }, {
      "alarmMetricName" : "Unacknowledged Message Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Outstanding",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Sent/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Database Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Flushes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster API Calls/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource API Calls/sec",
      "monitor" : true
    }, {
```

```
      "alarmMetricName" : "Cluster Handles/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource Handles/sec",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL"],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_FAILOVER_CLUSTER_INSTANCE-<RESOURCE_GROUP_NAME>",
      "logPath" : "\\\\amznfsxjmzbykwn.mydomain.aws\\SQLDB\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
```

```
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

Crea e configura il monitoraggio di CloudWatch Application Insights utilizzando modelli CloudFormation

Puoi aggiungere il monitoraggio di Application Insights, comprese le metriche chiave e la telemetria, all'applicazione, al database e al server Web, direttamente dai modelli. AWS CloudFormation

Questa sezione fornisce AWS CloudFormation modelli di esempio in formato JSON e YAML per aiutarti a creare e configurare il monitoraggio di Application Insights.

Per visualizzare il riferimento alle risorse e alle proprietà di Application Insights nella Guida per l'AWS CloudFormation utente, consulta il riferimento al tipo di ApplicationInsights risorsa.

Modelli di esempio

- Crea un'applicazione Application Insights per l'intero AWS CloudFormation stack
- Creazione di un'applicazione Application Insights con impostazioni dettagliate
- Creazione di un'applicazione Application Insights con configurazione del componente in modalità CUSTOM
- Creazione di un'applicazione Application Insights con configurazione del componente in modalità DEFAULT
- Creazione di un'applicazione Application Insights con configurazione del componente in modalità DEFAULT_WITH_OVERWRITE

Crea un'applicazione Application Insights per l'intero AWS CloudFormation stack

Per applicare il seguente modello, è necessario creare AWS risorse e uno o più gruppi di risorse da cui creare applicazioni Application Insights per monitorare tali risorse. Per ulteriori informazioni, consulta Nozioni di base su AWS Resource Groups.

Le prime due parti del modello seguente specificano una risorsa e un gruppo di risorse. L'ultima parte del modello crea un'applicazione Application Insights per il gruppo di risorse, ma non configura l'applicazione né applica il monitoraggio. Per ulteriori informazioni, consulta i dettagli del CreateApplicationcomando nell'Amazon CloudWatch Application Insights API Reference.

Modello in formato JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Resource Group stack",
    "Resources": {
        "EC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId" : "ami-abcd1234efgh5678i",
                "SecurityGroupIds" : ["sg-abcd1234"]
            }
        },
        ...
        "ResourceGroup": {
            "Type": "AWS::ResourceGroups::Group",
            "Properties": {
                "Name": "my_resource_group"
            }
        },
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
                "ResourceGroupName": "my_resource_group"
            },
            "DependsOn" : "ResourceGroup"
        }
    }
}
```

Modello in formato YAML

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Resource Group stack
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-abcd1234efgh5678i
      SecurityGroupIds:
      - sg-abcd1234
  ...
  ResourceGroup:
    Type: AWS::ResourceGroups::Group
    Properties:
      Name: my_resource_group
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
    DependsOn: ResourceGroup
```

La sezione del modello seguente applica la configurazione di monitoraggio predefinita all'applicazione Application Insights. Per ulteriori informazioni, consulta i dettagli del CreateApplicationcomando nell'Amazon CloudWatch Application Insights API Reference.

Quando `AutoConfigurationEnabled` è impostato su `true`, tutti i componenti dell'applicazione vengono configurati con le impostazioni di monitoraggio consigliate per il livello `DEFAULT`. Per ulteriori informazioni su queste impostazioni e livelli, consulta DescribeComponentConfigurationRecommendatione UpdateComponentConfigurationconsulta l'Amazon CloudWatch Application Insights API Reference.

Modello in formato JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Application Insights Application stack",
    "Resources": {
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
                "ResourceGroupName": "my_resource_group",
                "AutoConfigurationEnabled": true
```

```
            }
        }
    }
}
```

Modello in formato YAML

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Application Insights Application stack
Resources:
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
      AutoConfigurationEnabled: true
```

Creazione di un'applicazione Application Insights con impostazioni dettagliate

Questo modello di esempio esegue le seguenti operazioni:

- Crea un'applicazione Application Insights con notifica CloudWatch degli eventi e OpsCenter abilitata. Per ulteriori informazioni, consulta i dettagli del CreateApplicationcomando nell'Amazon CloudWatch Application Insights API Reference.
- Applica tag all'applicazione con due tag, uno dei quali non ha valori di tag. Per ulteriori informazioni, consulta TagResourceAmazon CloudWatch Application Insights API Reference.
- Crea due componenti del gruppo di istanze personalizzate. Per ulteriori informazioni, consulta CreateComponentAmazon CloudWatch Application Insights API Reference.
- Crea due set di modelli di log. Per ulteriori informazioni, consulta CreateLogPatternAmazon CloudWatch Application Insights API Reference.
- Imposta `AutoConfigurationEnabled` su `true`, che configura tutti i componenti dell'applicazione con le impostazioni di monitoraggio consigliate per il livello `DEFAULT`. Per ulteriori informazioni, consulta DescribeComponentConfigurationRecommendationAmazon CloudWatch Application Insights API Reference.

Modello in formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
```

```
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "CWEMonitorEnabled": true,
        "OpsCenterEnabled": true,
        "OpsItemSNSTopicArn": "arn:aws:sns:us-east-1:123456789012:my_topic",
        "AutoConfigurationEnabled": true,
        "Tags": [
            {
                "Key": "key1",
                "Value": "value1"
            },
            {
                "Key": "key2",
                "Value": ""
            }
        ],
        "CustomComponents": [
            {
                "ComponentName": "test_component_1",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            },
            {
                "ComponentName": "test_component_2",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i",
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            }
        ],
        "LogPatternSets": [
            {
                "PatternSetName": "pattern_set_1",
                "LogPatterns": [
                    {
                        "PatternName": "deadlock_pattern",
                        "Pattern": ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))",
                        "Rank": 1
                    }
                ]
            },
            {
                "PatternSetName": "pattern_set_2",
```

```
                "LogPatterns": [
                    {
                        "PatternName": "error_pattern",
                        "Pattern": ".*[\\s\\[]ERROR[\\s\\]].*",
                        "Rank": 1
                    },
                    {
                        "PatternName": "warning_pattern",
                        "Pattern": ".*[\\s\\[]WARN(ING)?[\\s\\]].*",
                        "Rank": 10
                    }
                ]
            }
        ]
    }
}
```

Modello in formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  CWEMonitorEnabled: true
  OpsCenterEnabled: true
  OpsItemSNSTopicArn: arn:aws:sns:us-east-1:123456789012:my_topic
  AutoConfigurationEnabled: true
  Tags:
  - Key: key1
    Value: value1
  - Key: key2
    Value: ''
  CustomComponents:
  - ComponentName: test_component_1
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  - ComponentName: test_component_2
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  LogPatternSets:
  - PatternSetName: pattern_set_1
    LogPatterns:
```

```
    - PatternName: deadlock_pattern
      Pattern: ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))"
      Rank: 1
  - PatternSetName: pattern_set_2
    LogPatterns:
    - PatternName: error_pattern
      Pattern: ".*[\\s\\[]ERROR[\\s\\]].*"
      Rank: 1
    - PatternName: warning_pattern
      Pattern: ".*[\\s\\[]WARN(ING)?[\\s\\]].*"
      Rank: 10
```

Creazione di un'applicazione Application Insights con configurazione del componente in modalità **`CUSTOM`**

Questo modello di esempio esegue le seguenti operazioni:

- Crea un'applicazione Application Insights. Per ulteriori informazioni, consulta CreateApplicationAmazon CloudWatch Application Insights API Reference.
- Il componente `my_component` imposta `ComponentConfigurationMode` su `CUSTOM`, che causa la configurazione di questo componente con la configurazione specificata in `CustomComponentConfiguration`. Per ulteriori informazioni, consulta UpdateComponentConfigurationAmazon CloudWatch Application Insights API Reference.

Modello in formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group,
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "CUSTOM",
                "CustomComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            },
```

```
                        ...
                    ],
                    "Logs": [
                        {
                            "LogGroupName": "my_log_group_1",
                            "LogPath": "C:\\LogFolder_1\\*",
                            "LogType": "DOT_NET_CORE",
                            "Encoding": "utf-8",
                            "PatternSet": "my_pattern_set_1"
                        },
                        ...
                    ],
                    "WindowsEvents": [
                        {
                            "LogGroupName": "my_windows_event_log_group_1",
                            "EventName": "Application",
                            "EventLevels": [
                                "ERROR",
                                "WARNING",
                                ...
                            ],
                            "Encoding": "utf-8",
                            "PatternSet": "my_pattern_set_2"
                        },
                        ...
                    ],
                    "Alarms": [
                        {
                            "AlarmName": "my_alarm_name",
                            "Severity": "HIGH"
                        },
                        ...
                    ]
                },
                "SubComponentTypeConfigurations": [
                    {
                        "SubComponentType": "EC2_INSTANCE",
                        "SubComponentConfigurationDetails": {
                            "AlarmMetrics": [
                                {
                                    "AlarmMetricName": "DiskReadOps"
                                },
                                ...
                            ],
```

```
                            "Logs": [
                                {
                                    "LogGroupName": "my_log_group_2",
                                    "LogPath": "C:\\LogFolder_2\\*",
                                    "LogType": "IIS",
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set_3"
                                },
                                ...
                            ],
                            "processes" : [
                                {
                                    "processName" : "my_process",
                                    "alarmMetrics" : [
                                {
                                    "alarmMetricName" : "procstat cpu_usage",
                                    "monitor" : true
                                }, {
                                    "alarmMetricName" : "procstat memory_rss",
                                    "monitor" : true
                                }
                            ]
                        }
                    ],
                            "WindowsEvents": [
                                {
                                    "LogGroupName": "my_windows_event_log_group_2",
                                    "EventName": "Application",
                                    "EventLevels": [
                                        "ERROR",
                                        "WARNING",
                                        ...
                                    ],
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set_4"
                                },
                                ...
                            ]
                        }
                    }
                ]
            }
        }
    ]
```

```
    }
}
```

Modello in formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: CUSTOM
    CustomComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
        ...
        Logs:
        - LogGroupName: my_log_group_1
          LogPath: C:\LogFolder_1\*
          LogType: DOT_NET_CORE
          Encoding: utf-8
          PatternSet: my_pattern_set_1
        ...
        WindowsEvents:
        - LogGroupName: my_windows_event_log_group_1
          EventName: Application
          EventLevels:
          - ERROR
          - WARNING
          ...
          Encoding: utf-8
          PatternSet: my_pattern_set_2
        ...
        Alarms:
        - AlarmName: my_alarm_name
          Severity: HIGH
        ...
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          AlarmMetrics:
```

```
          - AlarmMetricName: DiskReadOps
          ...
          Logs:
          - LogGroupName: my_log_group_2
            LogPath: C:\LogFolder_2\*
            LogType: IIS
            Encoding: utf-8
            PatternSet: my_pattern_set_3
          ...
          Processes:
          - ProcessName: my_process
            AlarmMetrics:
            - AlarmMetricName: procstat cpu_usage
              ...
            ...
          WindowsEvents:
          - LogGroupName: my_windows_event_log_group_2
            EventName: Application
            EventLevels:
            - ERROR
            - WARNING
            ...
            Encoding: utf-8
            PatternSet: my_pattern_set_4
          ...
```

Creazione di un'applicazione Application Insights con configurazione del componente in modalità **`DEFAULT`**

Questo modello di esempio esegue le seguenti operazioni:

- Crea un'applicazione Application Insights. Per ulteriori informazioni, consulta CreateApplicationAmazon CloudWatch Application Insights API Reference.
- Il componente `my_component` imposta `ComponentConfigurationMode` su `DEFAULT` e `Tier` su `SQL_SERVER`, il che causa la configurazione del componente con le impostazioni di configurazione consigliate da Application Insights per il livello `SQL_Server`. Per ulteriori informazioni, consulta DescribeComponentConfiguratione UpdateComponentConfigurationconsulta l'Amazon CloudWatch Application Insights API Reference.

Modello in formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "DEFAULT"
            }
        ]
    }
}
```

Modello in formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: DEFAULT
```

Creazione di un'applicazione Application Insights con configurazione del componente in modalità **`DEFAULT_WITH_OVERWRITE`**

Questo modello di esempio esegue le seguenti operazioni:

- Crea un'applicazione Application Insights. Per ulteriori informazioni, consulta CreateApplicationAmazon CloudWatch Application Insights API Reference.
- Il componente `my_component` imposta `ComponentConfigurationMode` su `DEFAULT_WITH_OVERWRITE` e `tier` su `DOT_NET_CORE`, il che causa la configurazione del componente con le impostazioni di configurazione consigliate da Application Insights per il livello `DOT_NET_CORE`. Le impostazioni di configurazione sovrascritte sono specificate in `DefaultOverwriteComponentConfiguration`:
  - A livello di componente, le impostazioni `AlarmMetrics` vengono sovrascritte.

- A livello di componente secondario, per i componenti secondari di tipo `EC2_Instance`, le impostazioni `Logs` vengono sovrascritte.

Per ulteriori informazioni, consulta [UpdateComponentConfiguration](#)Amazon CloudWatch Application Insights API Reference.

Modello in formato JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentName": "my_component",
                "Tier": "DOT_NET_CORE",
                "ComponentConfigurationMode": "DEFAULT_WITH_OVERWRITE",
                "DefaultOverwriteComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            }
                        ]
                    },
                    "SubComponentTypeConfigurations": [
                        {
                            "SubComponentType": "EC2_INSTANCE",
                            "SubComponentConfigurationDetails": {
                                "Logs": [
                                    {
                                        "LogGroupName": "my_log_group",
                                        "LogPath": "C:\\LogFolder\\*",
                                        "LogType": "IIS",
                                        "Encoding": "utf-8",
                                        "PatternSet": "my_pattern_set"
                                    }
                                ]
                            }
                        }
                    ]
                }
```

```
            }
        ]
    }
}
```

Modello in formato YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentName: my_component
    Tier: DOT_NET_CORE
    ComponentConfigurationMode: DEFAULT_WITH_OVERWRITE
    DefaultOverwriteComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          Logs:
          - LogGroupName: my_log_group
            LogPath: C:\LogFolder\*
            LogType: IIS
            Encoding: utf-8
            PatternSet: my_pattern_set
```

Esercitazione: configurazione del monitoraggio per SAP ASE

Questo tutorial dimostra come configurare CloudWatch Application Insights per configurare il monitoraggio per i database SAP ASE. È possibile utilizzare i dashboard automatici di CloudWatch Application Insights per visualizzare i dettagli dei problemi, accelerare la risoluzione dei problemi e facilitare il tempo medio di risoluzione (MTTR) per i database SAP ASE.

Approfondimenti sulle applicazioni per argomenti SAP ASE

- Ambienti supportati
- Sistemi operativi supportati
- Funzionalità

- Prerequisiti
- Configurazione del monitoraggio sul database SAP ASE
- Gestione del monitoraggio del database SAP ASE
- Configurazione della soglia dell'allarme
- Visualizzazione e risoluzione dei problemi di SAP ASE rilevati da Approfondimenti sulle applicazioni
- Risoluzione dei problemi relativi ad Approfondimenti sulle applicazioni per SAP ASE

Ambienti supportati

CloudWatch Application Insights supporta l'implementazione di AWS risorse per i seguenti sistemi e modelli. È possibile fornire e installare il software di database SAP ASE e il software applicativo SAP supportato.

- Uno o più database SAP ASE su una singola istanza Amazon EC2: SAP ASE in un'architettura dimensionabile a nodo singolo.
- Configurazione dei database SAP ASE con disponibilità elevata tra AZ: SAP ASE con disponibilità elevata tra configurata su due zone di disponibilità utilizzando il clustering SUSE/RHEL.

> **Note**
>
> CloudWatch Application Insights supporta solo ambienti ASE HA con ID di sistema SAP (SID) singolo. Se sono collegati più SID HA ASE, il monitoraggio verrà configurato solo per il primo SID rilevato.

Sistemi operativi supportati

CloudWatch Application Insights for SAP ASE supporta l'architettura x86-64 sui seguenti sistemi operativi:

- SuSE Linux 12 SP4
- SuSE Linux 12 SP5
- SuSE Linux 15
- Usa Linux 15 SP1

- Utilizzare Linux 15 SP2
- SuSE Linux 15 SP3
- SuSE Linux 15 SP4
- SuSE Linux 15 SP1 per SAP
- SuSE Linux 15 SP2 per SAP
- SuSE Linux 15 SP3 per SAP
- SuSE Linux 15 SP4 per SAP
- SuSE Linux 12 SP4 per SAP
- SuSE Linux 12 SP5 per SAP
- RedHat Linux 7.6
- RedHat Linux 7.7
- RedHat Linux 7.9
- RedHat Linux 8.1
- RedHat Linux 8.4
- RedHat Linux 8.6

Funzionalità

CloudWatch Application Insights for SAP ASE offre le seguenti funzionalità:

- Rilevamento automatico dei carichi di lavoro SAP ASE
- Creazione automatica di allarmi SAP ASE in base alla soglia statica
- Creazione automatica di allarmi SAP ASE in base al rilevamento di anomalie
- Riconoscimento automatico del modello di log SAP ASE
- Pannello di controllo dell'integrità per SAP ASE
- Pannello di controllo dei problemi per SAP ASE

Prerequisiti

È necessario soddisfare i seguenti prerequisiti per configurare un database SAP ASE con CloudWatch Application Insights:

- Parametri di configurazione SAP ASE: i seguenti parametri di configurazione devono essere abilitati sul database ASE: `"enable monitoring"`, `"sql text pipe max messages"` e `"sql text pipe active"`. Ciò consente ad CloudWatch Application Insights di fornire funzionalità di monitoraggio complete per il database. Se queste impostazioni non sono abilitate nel database ASE, Approfondimenti sulle applicazioni le abiliterà automaticamente a raccogliere i parametri necessari per consentire il monitoraggio.
- Utente del database SAP ASE: l'utente del database fornito durante l'onboarding di Approfondimenti sulle applicazioni deve disporre dell'autorizzazione per accedere a quanto segue:
  - Tabelle di sistema nel database master e nei database degli utenti (tenant)
  - Monitoraggio delle tabelle
- SAP HostCtrl: installa e configura SAP HostCtrl sulla tua istanza Amazon EC2.
- CloudWatch Agente Amazon: assicurati di non eseguire un CloudWatch agente preesistente sulla tua istanza Amazon EC2. Se hai installato un CloudWatch agente, assicurati di rimuovere la configurazione delle risorse che stai utilizzando in CloudWatch Application Insights dal file di configurazione dell' CloudWatch agente esistente per evitare un conflitto di fusione. Per ulteriori informazioni, consulta Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.
- AWS Abilitazione di Systems Manager: installa SSM Agent sulle tue istanze e abilita le istanze abilitate per SSM. Per ulteriori informazioni su SSM Agent, consulta Utilizzo di SSM Agent nella Guida per l'utente di Systems Manager AWS .
- Ruoli delle istanze Amazon EC2: è necessario collegare i seguenti ruoli di istanza Amazon EC2 per configurare il database.
  - È necessario allegare il ruolo `AmazonSSMManagedInstanceCore` per abilitare Systems Manager. Per ulteriori informazioni, consulta Esempi di policy basate sull'identità nella AWS Systems Manager.
  - È necessario allegare il `CloudWatchAgentServerPolicy` per consentire l'emissione dei parametri e dei log dell'istanza. CloudWatch Per ulteriori informazioni, consulta Creare ruoli e utenti IAM da utilizzare con l' CloudWatch agente Amazon.
  - È necessario allegare la seguente policy inline IAM al ruolo dell'istanza Amazon EC2 per leggere la password memorizzata in AWS Secrets Manager. Per ulteriori informazioni sulle policy inline, consulta Policy gestite e policy inline nella Guida per l'utente di IAM AWS Identity and Access Management .

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- AWS Resource Groups— È necessario creare un gruppo di risorse che includa tutte le AWS risorse associate utilizzate dallo stack di applicazioni per l'onboarding delle applicazioni in CloudWatch Application Insights. Sono incluse le istanze Amazon EC2 e i volumi Amazon EBS che eseguono il database SAP ASE. Se sono presenti più database per account, si consiglia di creare un gruppo di risorse che includa le AWS risorse per ogni sistema di database SAP ASE.
- Autorizzazioni IAM: per gli utenti non amministratori:
  - È necessario creare una policy AWS Identity and Access Management (IAM) che consenta ad Application Insights di creare un ruolo collegato al servizio e collegarlo alla propria identità utente. Per le fasi di collegamento della policy, consulta Policy IAM.
  - L'utente deve disporre dell'autorizzazione per creare un account segreto per AWS Secrets Manager archiviare le credenziali utente del database. Per ulteriori informazioni, consulta la sezione Esempio: autorizzazione alla creazione di segreti.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Ruolo collegato al servizio: Application Insights utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato al servizio viene creato per te quando crei la tua

prima applicazione Application Insights nella console Application Insights. Per ulteriori informazioni, consulta la pagina Utilizzo di ruoli collegati ai servizi per CloudWatch Application Insights.

Configurazione del monitoraggio sul database SAP ASE

Per configurare il monitoraggio per il database SAP ASE, completa i seguenti passaggi.

1. Apri la CloudWatch console.
2. Nel pannello di navigazione a sinistra, scegli Application Insights sotto Application Insights (Informazioni dettagliate applicazione).
3. La pagina Application Insights visualizza l'elenco delle applicazioni che sono monitorate con Application Insights e lo stato di monitoraggio per ciascuna applicazione. Nell'angolo in alto a destra, scegli Aggiungere un'applicazione.
4. Nella pagina Specifica i dettagli dell'applicazione, dall'elenco a discesa sotto Gruppo di risorse, seleziona il gruppo di risorse AWS contenente le risorse del database SAP ASE. Se non hai creato un gruppo di risorse per l'applicazione, puoi crearne uno scegliendo Crea nuovo gruppo di risorse sotto il menu a discesa Resource Group (Gruppo di risorse). Per ulteriori informazioni sui resource groups, consulta la Guida per l'utente di Resource Groups AWS .
5. In Monitor CloudWatch Events, seleziona la casella di controllo per integrare il monitoraggio di Application Insights con CloudWatch Events per ottenere informazioni da Amazon EBS, Amazon EC2 AWS CodeDeploy, Amazon ECS AWS Health , API e notifiche, Amazon RDS, Amazon S3 e. AWS Step Functions
6. In Integra con AWS Systems Manager OpsCenter, seleziona la casella di controllo accanto a Genera AWS Systems Manager OpsCenter OpsItems per azioni correttive per visualizzare e ricevere notifiche quando vengono rilevati problemi per le applicazioni selezionate. Per tenere traccia delle operazioni eseguite per risolvere gli elementi di lavoro operativi, denominati OpsItems, correlati alle AWS risorse, fornisci un argomento SNS ARN.
7. Facoltativamente, puoi inserire tag per aiutarti a identificare e organizzare le tue risorse. CloudWatch Application Insights supporta gruppi di risorse AWS CloudFormation basati su tag e stack, ad eccezione dei gruppi. Application Auto Scaling Per ulteriori informazioni, consulta Tag Editor nella Guida per l'utente di AWS Resource Groups .
8. Scegli Successivo per continuare a impostare il monitoraggio.
9. Nella pagina Rivedi i componenti rilevati, sono elencati i componenti monitorati e i relativi carichi di lavoro rilevati automaticamente da CloudWatch Application Insights.

> **Note**
>
> I componenti che contengono un carico di lavoro SAP ASE High Availability rilevato supportano solo un carico di lavoro su un componente. I componenti che contengono un carico di lavoro SAP ASE a nodo singolo rilevato supportano più carichi di lavoro, ma non è possibile aggiungere o rimuovere carichi di lavoro. Tutti i carichi di lavoro rilevati automaticamente verranno monitorati.

10. Seleziona Successivo.
11. Sulla pagina Specifica i dettagli dei componenti, immetti il nome utente e la password dei tuoi database SAP ASE.
12. Esamina la configurazione del monitoraggio delle applicazioni e scegli Invia.
13. Si apre la pagina dei dettagli dell'applicazione, in cui è possibile visualizzare riepilogo dell'applicazione, l'elenco di componenti e carichi di lavoro monitorati e i componenti e i carichi di lavoro non monitorati. Se si seleziona il pulsante di opzione accanto a un componente o a un carico di lavoro, è anche possibile visualizzare la cronologia della configurazione, i modelli di log e qualsiasi Tag creato. Quando si invia la configurazione, l'account implementa tutti i parametri e gli allarmi per il sistema SAP ASE; l'operazione può richiedere fino a 2 ore.

Gestione del monitoraggio del database SAP ASE

È possibile gestire le credenziali utente, i parametri e i percorsi di log per il database SAP ASE completando i seguenti passaggi:

1. Apri la CloudWatch console.
2. Nel pannello di navigazione a sinistra, scegli Application Insights sotto Application Insights (Informazioni dettagliate applicazione).
3. La pagina Application Insights visualizza l'elenco delle applicazioni che sono monitorate con Application Insights e lo stato di monitoraggio per ciascuna applicazione.
4. In Componenti monitorati, seleziona il pulsante di opzione accanto al nome del componente. Quindi, scegli Gestisci il monitoraggio.
5. In Log dei gruppi di istanze EC2, è possibile aggiornare il percorso di registro esistente, il set di pattern di log e il nome del gruppo di log. Inoltre, è possibile aggiungere fino a tre altri Log di applicazioni.

6. In Parametri, è possibile scegliere i parametri SAP ASE in base alle proprie esigenze. I nomi dei parametri SAP ASE hanno il prefisso `asedb`. È possibile aggiungere fino a 60 parametri per componente.
7. In Configurazione ASE, immetti la password e il nome utente per il database SAP ASE. Si tratta del nome utente e della password che CloudWatch l'agente Amazon utilizza per connettersi al database SAP ASE.
8. In Allarmi personalizzati, puoi aggiungere altri allarmi da CloudWatch monitorare con Application Insights.
9. Esamina la configurazione del monitoraggio dell'applicazione e scegli Invia. Quando si invia la configurazione, l'account aggiorna tutti i parametri e gli allarmi per il sistema SAP HANA, che possono richiedere fino a 2 ore.

Configurazione della soglia dell'allarme

CloudWatch Application Insights crea automaticamente una CloudWatch metrica Amazon per l'allarme da monitorare, insieme alla soglia per tale metrica. L'allarme passa nello stato `ALARM` quando il parametro supera la soglia per un numero specificato di periodi di valutazione. Si noti che queste impostazioni non vengono mantenute da Application Insights.

Per modificare un allarme per un singolo parametro, procedere come segue:

1. Apri la CloudWatch console.
2. Nel pannello di navigazione, seleziona Alarms (Allarmi), All alarms (Tutti gli allarmi).
3. Seleziona il pulsante di opzione accanto all'allarme creato automaticamente da CloudWatch Application Insights. Quindi scegli Operazioni e seleziona Modifica dal menu a discesa.
4. Modificare i seguenti parametri sotto Parametro.
    a. In Statistic (Statistica), scegli una delle statistiche o percentili predefiniti oppure specifichi un percentile personalizzato. Ad esempio, `p95.45`.
    b. In Period (Periodo), scegli il periodo di valutazione per l'allarme. Durante la valutazione dell'allarme, ogni periodo è aggregato in un punto dati.
5. Modificare i seguenti parametri sotto Condizioni.
    a. Specificare se il parametro deve essere maggiore di, minore di o uguale alla soglia.
    b. Specificare il valore della soglia.
6. In Configurazione aggiuntiva, modificare i seguenti parametri.

a. In Datapoint per l'allarme, specificare il numero di punti dati, o periodi di valutazione, che devono essere inclusi nello stato `ALARM` per avviare l'allarme. Quando i due valori corrispondono, viene creato un allarme che entra nello stato `ALARM` se il numero designato di periodi consecutivi viene superato. Per creare un allarme `m` di `n`, specificare un numero minore per il primo valore rispetto a quello specificato per il secondo valore. Per ulteriori informazioni sulla valutazione degli allarmi, consulta Valutazione di un allarme.

b. Per Missing data treatment (Trattamento dati mancanti), scegli la modalità di comportamento dell'allarme quando mancano alcuni punti dati. Per ulteriori informazioni sul trattamento dei dati mancanti, vedi Configurazione del modo in cui gli CloudWatch allarmi trattano i dati mancanti.

c. Se l'allarme utilizza un percentile come statistica monitorata, viene visualizzata una casella Percentiles with low samples (Percentili con campioni ridotti). Scegli se valutare o ignorare casi con bassa frequenza di campionamento. Se scegli ignore (maintain alarm state) (ignora (mantieni stato dell'allarme)), lo stato corrente dell'allarme viene sempre mantenuto quando la dimensione dell'esempio è troppo bassa. Per ulteriori informazioni sui percentili con campioni bassi, consulta CloudWatch Allarmi basati su percentili ed esempi di dati limitati.

7. Seleziona Successivo.
8. In Notification (Notifica), seleziona un argomento SNS per segnalare quando l'allarme è nello stato `ALARM`, `OK` o `INSUFFICIENT_DATA`.
9. Seleziona Update Alarm (Aggiorna allarme).

Visualizzazione e risoluzione dei problemi di SAP ASE rilevati da Approfondimenti sulle applicazioni

Questa sezione fornisce istruzioni per risolvere i problemi comuni che si possono verificare durante la configurazione del monitoraggio per SAP ASE su Approfondimenti sulle applicazioni.

Errori del server di backup SAP ASE

È possibile identificare il messaggio di errore esaminando il pannello di controllo creato dinamicamente. Il pannello di controllo mostra il messaggio di errore riportato nel server di backup SAP ASE. Per ulteriori dettagli sui log del server di backup SAP ASE, consulta la pagina Backup Server Error Logging nella documentazione di SAP.

Transazioni SAP ASE di lunga durata

Identifica la transazione di lunga durata e verifica se può essere interrotta o se l'esecuzione è intenzionale. Per ulteriori dettagli, consulta la pagina 2180410 — How to display transaction log records for long running transactions? — SAP ASE.

Connessioni utente SAP ASE

Verifica se il tuo database SAP ASE è dimensionato in modo adeguato al carico di lavoro che intendi eseguire sul database. Per ulteriori dettagli, consulta Configuring User Connections nella documentazione SAP.

Spazio su disco SAP ASE

È possibile identificare il livello di database che causa il problema esaminando il pannello di controllo creato dinamicamente. Il pannello di controllo mostra i parametri correlati e i frammenti di file di log. È importante comprendere la causa dell'aumento dello spazio su disco e, laddove possibile, aumentare le dimensioni fisiche del disco, lo spazio su disco allocato o entrambi questi fattori. Per ulteriori dettagli, consulta SAP Documentation disk resize nella documentazione SAP.

Risoluzione dei problemi relativi ad Approfondimenti sulle applicazioni per SAP ASE

Questa sezione fornisce i passaggi per aiutarti a risolvere gli errori comuni restituiti dal pannello di controllo di Application Insights.

Errore	Errore restituito	Causa principale	Risoluzione
Impossibile aggiungere più di 60 parametri del monitor.	`Component cannot have more than 60 monitored metric`	Attualmente, il limite dei parametri è di 60 parametri monitorati per componente.	Rimuovi i parametri non necessari per rispettare il limite.
Non vengono visualizzati parametri o allarmi SAP dopo il processo di integrazione	Il comando `run` relativo a `AWS-ConfigureAWSPackage` non è riuscito in AWS Systems Manager. L'output mostra il seguente errore: `CT-LIBRARY error:ct_connec`	Il nome utente e la password potrebbero non essere corretti.	Verifica che il nome utente e la password siano validi ed esegui nuovamente il processo di integrazione.

Errore	Errore restituito	Causa principale	Risoluzione
	`t(): protocol specific layer: external error: The attempt to connect to the server failed`		

Esercitazione: Configurare il monitoraggio per SAP HANA

Questo tutorial dimostra come configurare CloudWatch Application Insights per configurare il monitoraggio per i database SAP HANA. Puoi utilizzare i dashboard automatici di CloudWatch Application Insights per visualizzare i dettagli dei problemi, accelerare la risoluzione dei problemi e facilitare il tempo medio di risoluzione (MTTR) per i tuoi database SAP HANA.

Application Insights per argomenti SAP HANA

- Ambienti supportati
- Sistemi operativi supportati
- Funzionalità
- Prerequisiti
- Configurare il database SAP HANA per il monitoraggio
- Gestire il monitoraggio del tuo database SAP HANA
- Visualizza e risolvi i problemi di SAP HANA rilevati da Application Insights CloudWatch
- Rilevamento delle anomalie per SAP HANA
- Risoluzione dei problemi relativi ad Application Insights per SAP HANA

Ambienti supportati

CloudWatch Application Insights supporta l'implementazione di AWS risorse per i seguenti sistemi e modelli. È possibile fornire e installare il software di database SAP HANA e il software applicativo SAP supportato.

- Database SAP HANA su una singola istanza Amazon EC2— SAP HANA in un'architettura scalabile a nodo singolo, con fino a 24 TB di memoria.

- Database SAP HANA su più istanze Amazon EC2— SAP HANA in un'architettura multi-nodo con scalabilità orizzontale.
- Configurazione ad alta disponibilità del database SAP HANA cross-AZ— SAP HANA con elevata disponibilità configurata su due zone di disponibilità utilizzando il clustering SUSE/RHEL.

> **Note**
>
> CloudWatch Application Insights supporta solo ambienti HANA con SID singolo. Se sono collegati più SID HANA, il monitoraggio verrà configurato solo per il primo SID rilevato.

Sistemi operativi supportati

CloudWatch Application Insights per SAP HANA supporta l'architettura x86-64 sui seguenti sistemi operativi:

- SuSE Linux 12 SP4 per SAP
- SuSE Linux 12 SP5 per SAP
- SuSE Linux 15
- Usa Linux 15 SP1
- Utilizzare Linux 15 SP2
- SuSE Linux 15 per SAP
- SuSE Linux 15 SP1 per SAP
- SuSE Linux 15 SP2 per SAP
- SuSE Linux 15 SP3 per SAP
- SuSE Linux 15 SP4 per SAP
- SuSE Linux 15 SP5 per SAP
- RedHat Linux 8.6 per SAP con elevata disponibilità e servizi di aggiornamento
- RedHat Linux 8.5 per SAP con alta disponibilità e servizi di aggiornamento
- RedHat Linux 8.4 per SAP con alta disponibilità e servizi di aggiornamento
- RedHat Linux 8.3 per SAP con alta disponibilità e servizi di aggiornamento
- RedHat Linux 8.2 per SAP con alta disponibilità e servizi di aggiornamento
- RedHat Linux 8.1 per SAP con alta disponibilità e servizi di aggiornamento

- RedHat Linux 7.9 per SAP con alta disponibilità e servizi di aggiornamento

Funzionalità

CloudWatch Application Insights per SAP HANA offre le seguenti funzionalità:

- Rilevamento automatico del carico di lavoro SAP HANA
- Creazione automatica di allarmi SAP HANA in base alla soglia statica
- Creazione automatica di allarmi SAP HANA in base al rilevamento di anomalie
- Riconoscimento automatico del pattern di log SAP H
- Pannello di controllo della Health per SAP HANA
- Pannello di controllo dei problemi per SAP HANA

Prerequisiti

È necessario soddisfare i seguenti prerequisiti per configurare un database SAP HANA con Application Insights: CloudWatch

- SAP HANA: installa un database SAP HANA 2.0 SPS05 funzionante e raggiungibile su un'istanza Amazon EC2.
- Utente del database SAP HANA: è necessario creare un utente del database con ruoli di monitoraggio nel database SYSTEM e in tutti i tenant.

  Esempio

  I seguenti comandi SQL consentono di creare un utente con ruoli di monitoraggio.

  ```
  su - <sid>adm
  hdbsql -u SYSTEM -p <SYSTEMDB password> -d SYSTEMDB
  CREATE USER CW_HANADB_EXPORTER_USER PASSWORD <Monitoring user password> NO
   FORCE_FIRST_PASSWORD_CHANGE;
  CREATE ROLE CW_HANADB_EXPORTER_ROLE;
  GRANT MONITORING TO CW_HANADB_EXPORTER_ROLE;
  GRANT CW_HANADB_EXPORTER_ROLE TO CW_HANADB_EXPORTER_USER;
  ```

- Python 3.8 — Installa Python 3.8 o versioni successive sul tuo sistema operativo. Utilizza l'ultima versione di Python. Se Python3 non viene rilevato sul sistema operativo, verrà installato Python 3.6.

Per ulteriori informazioni, consulta installation example.

> **Note**
>
> L'installazione manuale di Python 3.8 o versioni successive è richiesta per i sistemi operativi SuSE Linux 15 SP4, RedHat Linux 8.6 e versioni successive.

- Pip3 — Installa il programma di installazione, pip3, sul tuo sistema operativo. Se pip3 non viene rilevato sul sistema operativo, verrà installato.
- hdbclient — CloudWatch Application Insights utilizza il driver python per connettersi al database SAP HANA. Se il client non è installato in python3, assicurati di avere la versione del file tar hdbclient sotto. `2.10 or later` `/hana/shared/SID/hdbclient/`
- CloudWatch Agente Amazon: assicurati di non eseguire un CloudWatch agente preesistente sulla tua istanza Amazon EC2. Se hai installato un CloudWatch agente, assicurati di rimuovere la configurazione delle risorse che stai utilizzando in CloudWatch Application Insights dal file di configurazione dell' CloudWatch agente esistente per evitare un conflitto di fusione. Per ulteriori informazioni, consulta Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.
- AWS Abilitazione di Systems Manager: installa SSM Agent sulle tue istanze e le istanze devono essere abilitate per SSM. Per informazioni su come installare l'agente SSM, vedere Working with SSM Agent nella AWS Systems Manager User Guide.
- Ruoli delle istanze Amazon EC2: è necessario collegare i seguenti ruoli di istanza Amazon EC2 per configurare il database.
  - È necessario allegare il ruolo `AmazonSSMManagedInstanceCore` per abilitare Systems Manager. Per ulteriori informazioni, consulta Esempi di policy basate sull'identità nella AWS Systems Manager.
  - È necessario allegare il `CloudWatchAgentServerPolicy` per consentire l'emissione delle metriche e dei log dell'istanza. CloudWatch Per ulteriori informazioni, consulta Creare ruoli e utenti IAM da utilizzare con l'agente. CloudWatch
  - È necessario allegare la seguente policy inline IAM al ruolo dell'istanza Amazon EC2 per leggere la password memorizzata in AWS Secrets Manager. Per ulteriori informazioni sulle policy inline, consulta Policy gestite e policy inline nella Guida per l'utente di IAM AWS Identity and Access Management .

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- AWS gruppi di risorse: è necessario creare un gruppo di risorse che includa tutte le AWS risorse associate utilizzate dallo stack di applicazioni per l'onboarding delle applicazioni in CloudWatch Application Insights. Sono incluse le istanze Amazon EC2 e i volumi Amazon EBS che eseguono il database SAP HANA. Se ci sono più database per account, ti consigliamo di creare un gruppo di risorse che includa le AWS risorse per ogni sistema di database SAP HANA.
- Autorizzazioni IAM: per gli utenti non amministratori:
  - È necessario creare una policy AWS Identity and Access Management (IAM) che consenta ad Application Insights di creare un ruolo collegato al servizio e collegarlo alla propria identità utente. Per le fasi di collegamento della policy, consulta Policy IAM.
  - L'utente deve disporre dell'autorizzazione per creare un account segreto per AWS Secrets Manager archiviare le credenziali utente del database. Per ulteriori informazioni, consulta la sezione Esempio: autorizzazione alla creazione di segreti.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Ruolo collegato al servizio: Application Insights utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato al servizio viene creato per te quando crei la tua

prima applicazione Application Insights nella console Application Insights. Per ulteriori informazioni, consulta la pagina Utilizzo di ruoli collegati ai servizi per CloudWatch Application Insights.

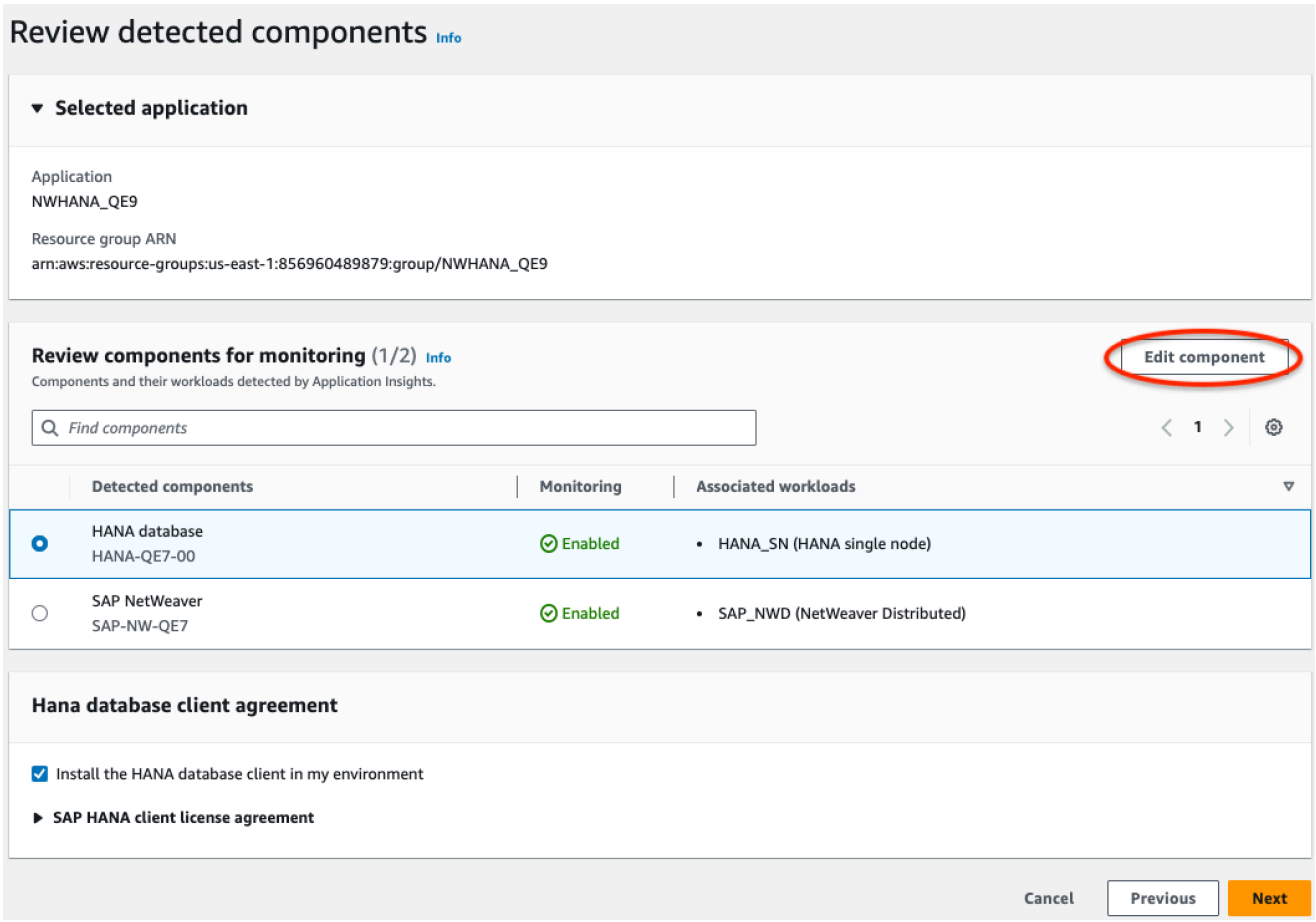
Configurare il database SAP HANA per il monitoraggio

Per configurare il monitoraggio per il database SAP HANA, completa i seguenti passaggi.

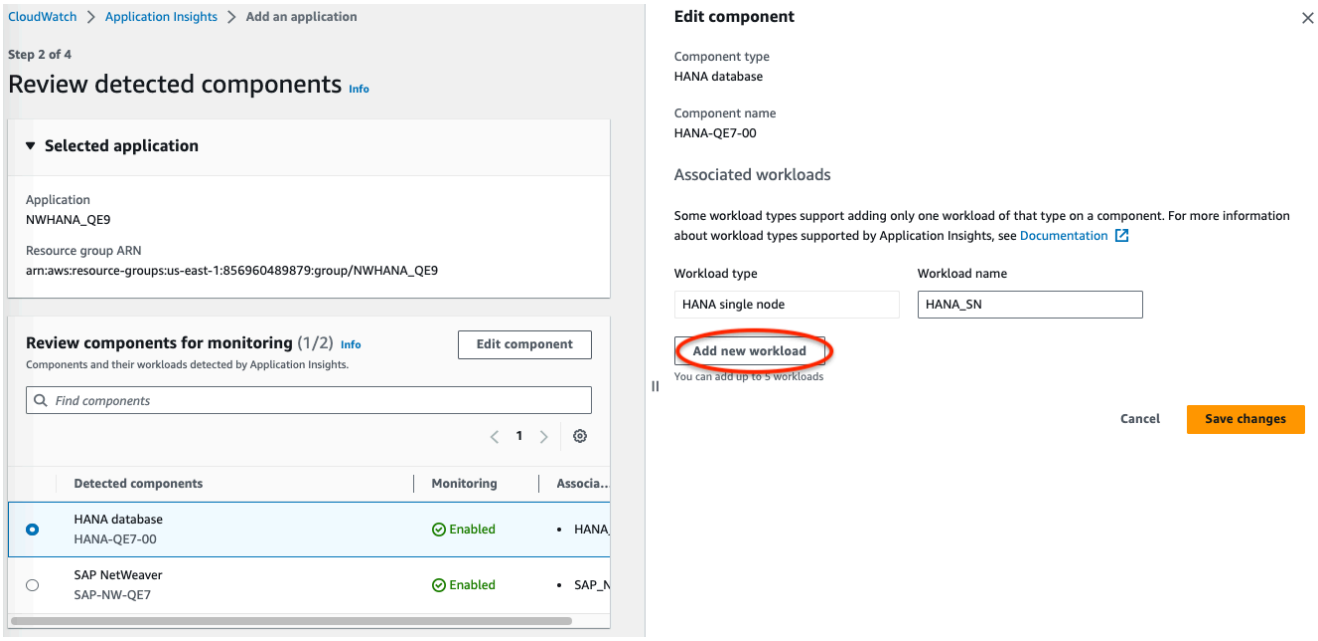
1. Apri la CloudWatch console.
2. Nel pannello di navigazione a sinistra, scegli Application Insights sotto Application Insights (Informazioni dettagliate applicazione).
3. La pagina Application Insights visualizza l'elenco delle applicazioni che sono monitorate con Application Insights e lo stato di monitoraggio per ciascuna applicazione. Nell'angolo in alto a destra, scegli Aggiungere un'applicazione.
4. Nella pagina Specifica i dettagli dell'applicazione, dall'elenco a discesa sotto Resource group, seleziona il resource group AWS contenente le risorse del database SAP HANA. Se non hai creato un gruppo di risorse per l'applicazione, puoi crearne uno scegliendo Crea nuovo gruppo di risorse sotto il menu a discesa Resource Group (Gruppo di risorse). Per ulteriori informazioni sui resource groups, consulta la Guida per l'utente di Resource Groups AWS .
5. In Monitor CloudWatch Events, seleziona la casella di controllo per integrare il monitoraggio di Application Insights con CloudWatch Events per ottenere informazioni da Amazon EBS, Amazon EC2 AWS CodeDeploy, Amazon ECS AWS Health , API e notifiche, Amazon RDS, Amazon S3 e. AWS Step Functions
6. In Integra con AWS Systems Manager OpsCenter, seleziona la casella di controllo accanto a Genera AWS Systems Manager OpsCenter OpsItems per azioni correttive per visualizzare e ricevere notifiche quando vengono rilevati problemi per le applicazioni selezionate. Per tenere traccia delle operazioni eseguite per risolvere gli elementi di lavoro operativi, denominati OpsItems, correlati alle AWS risorse, fornisci un argomento SNS ARN.
7. Facoltativamente, puoi inserire tag per aiutarti a identificare e organizzare le tue risorse. CloudWatch Application Insights supporta gruppi di risorse AWS CloudFormation basati su tag e stack, ad eccezione dei gruppi. Application Auto Scaling Per ulteriori informazioni, consulta Tag Editor nella Guida per l'utente di AWS Resource Groups .
8. Scegli Successivo per continuare a impostare il monitoraggio.
9. Nella pagina Rivedi i componenti rilevati, sono elencati i componenti monitorati e i relativi carichi di lavoro rilevati automaticamente da CloudWatch Application Insights.

a. Per aggiungere carichi di lavoro a un componente che contiene un carico di lavoro a nodo singolo SAP HANA rilevato, seleziona il componente, quindi scegli Modifica componente.

> **Note**
>
> I componenti che contengono un carico di lavoro multi-nodo SAP HANA o HANA High Availability rilevato supportano solo un carico di lavoro su un componente.



b. Per aggiungere un nuovo carico di lavoro, scegli Aggiungi nuovo carico di lavoro.



c. Al termine della modifica dei carichi di lavoro, scegli Salva modifiche.

10. Seleziona Successivo.
11. Sulla pagina Specifica i dettagli dei componenti, immetti il nome utente e la password.
12. Esamina la configurazione del monitoraggio delle applicazioni e scegli Invia.
13. Si apre la pagina dei dettagli dell'applicazione, in cui è possibile visualizzare riepilogo dell'applicazione, l'elenco di componenti e carichi di lavoro monitorati e i componenti e i carichi di lavoro non monitorati. Se si seleziona il pulsante di opzione accanto a un componente o a un carico di lavoro, è anche possibile visualizzare la cronologia della configurazione, i modelli di log e qualsiasi Tag creato. Quando si invia la configurazione, l'account distribuisce tutti i parametri e gli allarmi per il sistema SAP HANA, che possono richiedere fino a 2 ore.

Gestire il monitoraggio del tuo database SAP HANA

È possibile gestire le credenziali utente, i parametri e i percorsi di log per il database SAP HANA seguendo i seguenti passaggi:

1. Apri la CloudWatch console.
2. Nel pannello di navigazione a sinistra, scegli Application Insights sotto Application Insights (Informazioni dettagliate applicazione).
3. La pagina Application Insights visualizza l'elenco delle applicazioni che sono monitorate con Application Insights e lo stato di monitoraggio per ciascuna applicazione.

4. In Componenti monitorati, seleziona il pulsante di opzione accanto al nome del componente. Quindi, scegli Gestisci il monitoraggio.
5. In Log dei gruppi di istanze EC2, è possibile aggiornare il percorso di registro esistente, il set di pattern di log e il nome del gruppo di log. Inoltre, è possibile aggiungere fino a tre altri Log di applicazioni.
6. In Parametri, è possibile scegliere i parametri SAP HANA in base alle proprie esigenze. I nomi dei parametri SAP HANA hanno il prefisso `hanadb`. È possibile aggiungere fino a 40 parametri per componente.
7. In Configurazione HANA, immetti la password e il nome utente per il database SAP HANA. Si tratta del nome utente e della password che CloudWatch l'agente Amazon utilizza per connettersi al database SAP HANA.
8. In Allarmi personalizzati, puoi aggiungere altri allarmi che verranno monitorati da Application Insights. CloudWatch
9. Esamina la configurazione del monitoraggio dell'applicazione e scegli Invia. Quando si invia la configurazione, l'account aggiorna tutti i parametri e gli allarmi per il sistema SAP HANA, che possono richiedere fino a 2 ore.

Visualizza e risolvi i problemi di SAP HANA rilevati da Application Insights CloudWatch

Nelle sezioni seguenti vengono forniti passaggi che consentono di risolvere gli scenari di risoluzione dei problemi comuni che si verificano durante la configurazione del monitoraggio per SAP HANA su Application Insights.

Argomenti sulla risoluzione dei problemi

- Il database SAP HANA raggiunge il limite di allocazione della memoria
- Evento disco pieno
- Il backup SAP HANA ha smesso di funzionare

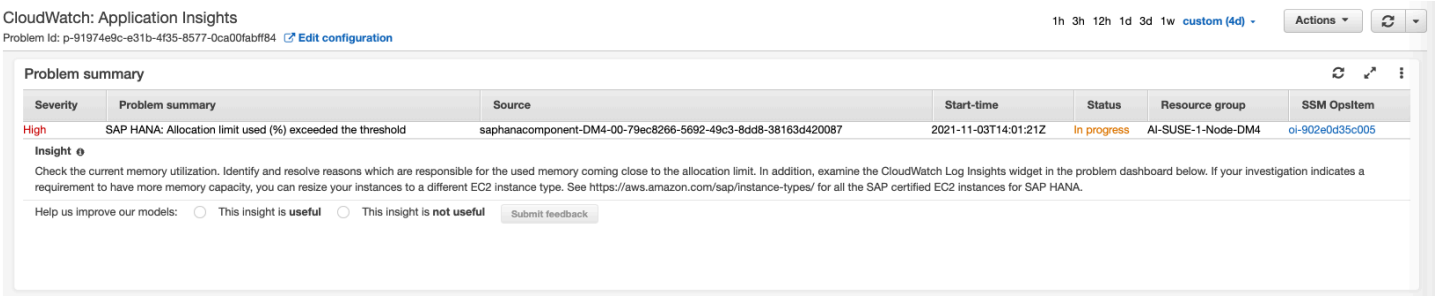
Il database SAP HANA raggiunge il limite di allocazione della memoria

Descrizione

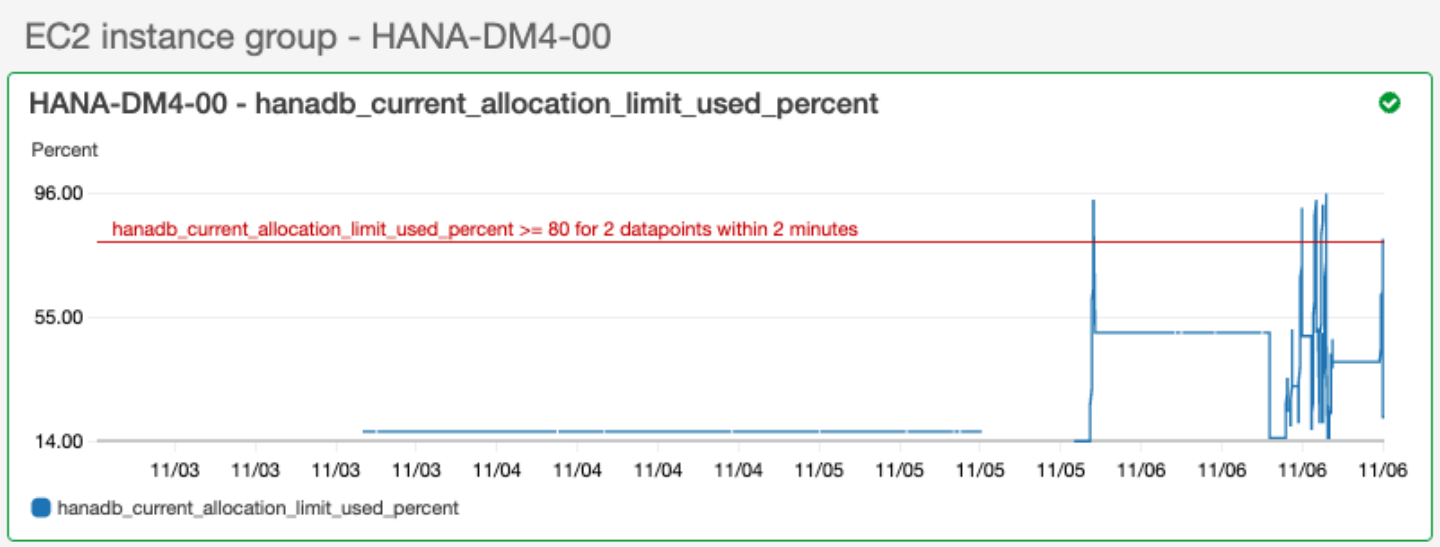
L'applicazione SAP supportata da un database SAP HANA non funziona a causa dell'elevata pressione della memoria, con conseguente riduzione delle prestazioni delle applicazioni.

Risoluzione

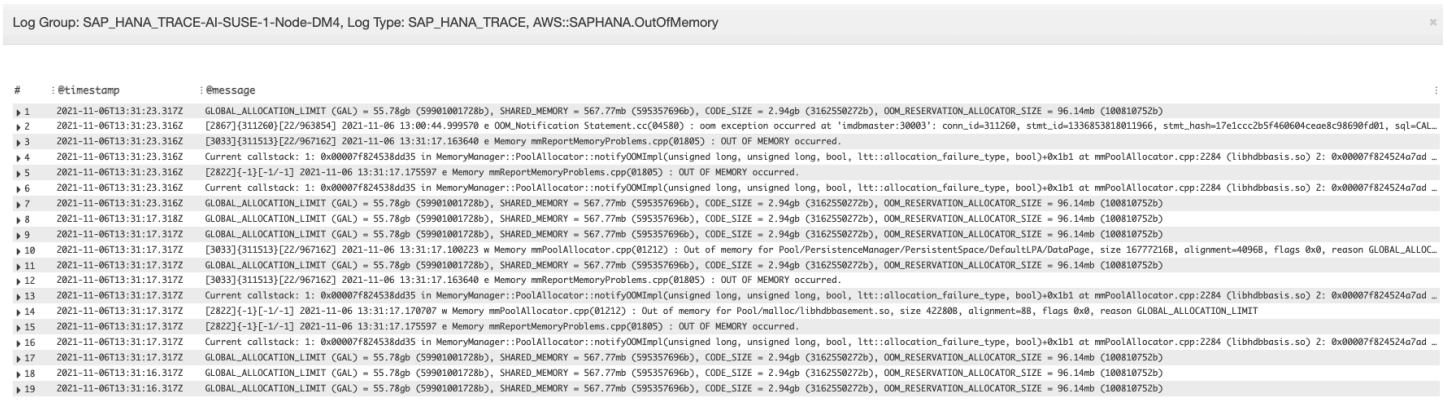
È possibile identificare il livello di applicazione che causa il problema controllando il pannello di controllo creato dinamicamente che mostra i parametri correlati e i frammenti del file di log. Nell'esempio seguente, il problema potrebbe essere dovuto a un elevato carico di dati nel sistema SAP HANA.



L'allocazione della memoria utilizzata supera la soglia dell'80% del limite totale di allocazione della memoria.



Il gruppo di log mostra che lo schema `BNR-DATA` e la tabella `IMDBMASTER_30003` hanno esaurito la memoria insufficiente. Inoltre, il gruppo di log mostra l'ora esatta del problema, il limite di posizione globale corrente, la memoria condivisa, la dimensione del codice e la dimensione dell'allocazione della prenotazione OOM.



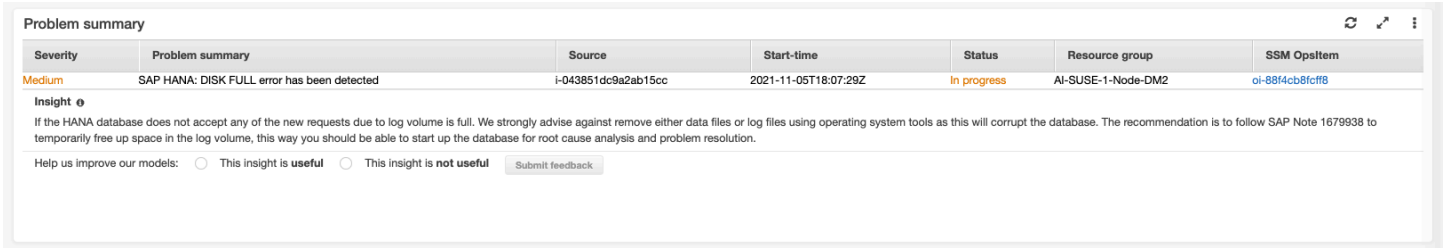
Evento disco pieno

Descrizione

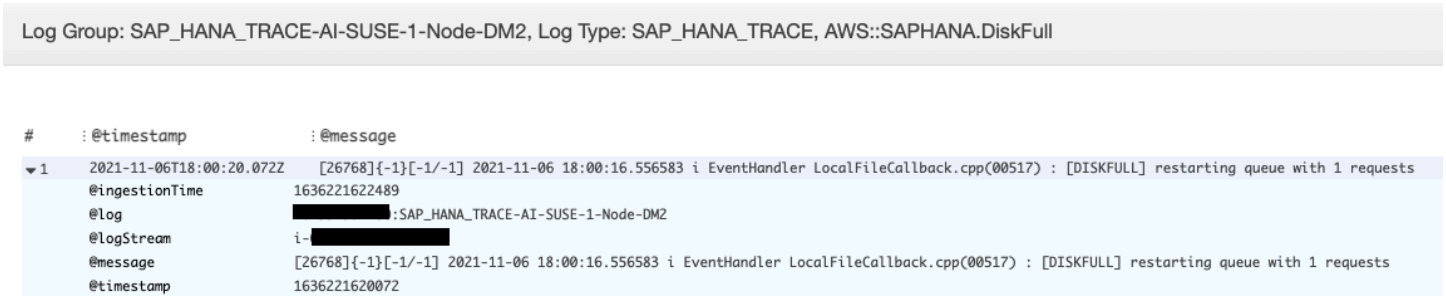
L'applicazione SAP supportata da un database SAP HANA smette di rispondere, il che comporta l'impossibilità di accedere al database.

Risoluzione

È possibile identificare il livello di database che causa il problema controllando il pannello di controllo creato dinamicamente che mostra i parametri correlati e i frammenti del file di log. Nell'esempio seguente, il problema potrebbe essere che l'amministratore non ha abilitato il backup automatico del registro, che ha causato il riempimento della directory sap/hana/log.



Il widget del gruppo di log nel pannello di controllo dei problemi mostra l'evento `DISKFULL`.



Il backup SAP HANA ha smesso di funzionare

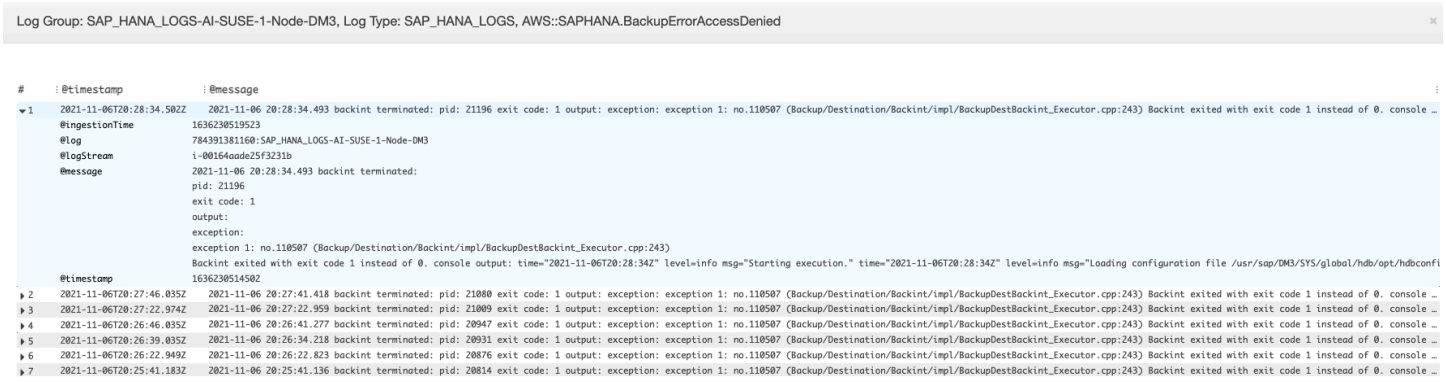
Descrizione

L'applicazione SAP supportata da un database SAP HANA ha smesso di funzionare.

Risoluzione

È possibile identificare il livello di database che causa il problema controllando il pannello di controllo creato dinamicamente che mostra i parametri correlati e i frammenti del file di log.

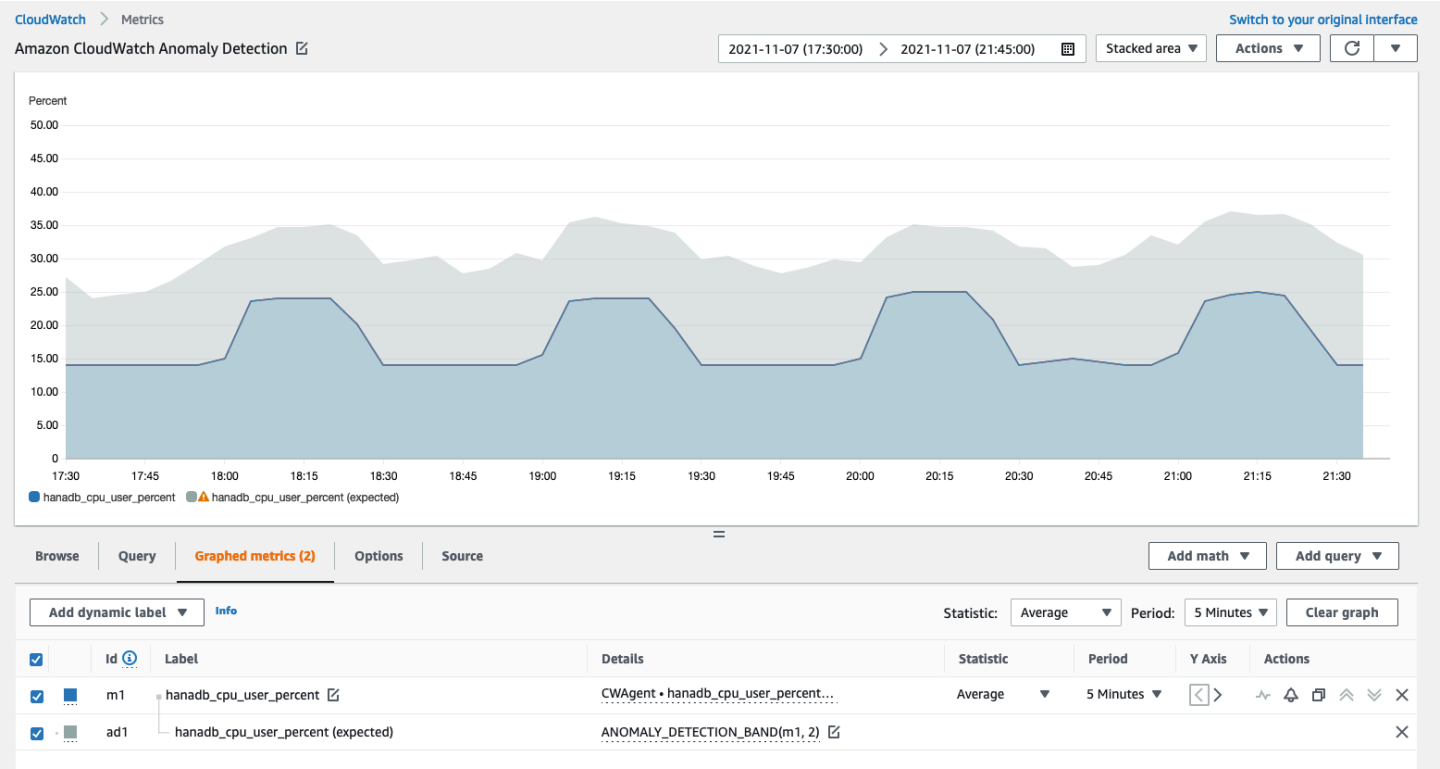
Il widget del gruppo di log nel pannello di controllo dei problemi mostra l'evento `ACCESS DENIED`. Ciò include informazioni aggiuntive, ad esempio il bucket S3, la cartella del bucket S3 e la regione del bucket S3.



Rilevamento delle anomalie per SAP HANA

Per metriche specifiche di SAP HANA, come il numero di thread, CloudWatch applica algoritmi statistici e di apprendimento automatico per definire la soglia. Questi algoritmi analizzano continuamente i parametri del database SAP HANA, determinano le normali linee di base e le anomalie superficiali con un intervento minimo dell'utente. Gli algoritmi generano un modello di rilevamento delle anomalie, che genera un intervallo di valori previsti che rappresentano il normale comportamento del parametro.

Gli algoritmi di rilevamento delle anomalie tengono conto delle variazioni di stagionalità e di tendenza dei parametri. Le variazioni di stagionalità potrebbero essere orarie, giornaliere o settimanali, come mostrato negli esempi seguenti dell'utilizzo della CPU di SAP HANA.



Dopo aver creato un modello, il rilevamento delle CloudWatch anomalie valuta continuamente il modello e lo aggiusta per garantire che sia il più preciso possibile. Ciò include la riqualificazione del modello da regolare se i valori dei parametri si evolvono nel tempo o subiscono cambiamenti improvvisi. Include anche predittori per migliorare i modelli per i parametri stagionali, spikey o scarsi.

Risoluzione dei problemi relativi ad Application Insights per SAP HANA

Questa sezione fornisce i passaggi per aiutarti a risolvere gli errori comuni restituiti dal pannello di controllo di Application Insights.

Impossibile aggiungere più di 60 metriche monitorate

L'output mostra il seguente errore.

```
Component cannot have more than 60 monitored metrics
```

Causa principale: il limite metrico attuale è di 60 metriche monitorate per componente.

Risoluzione: per rimanere al di sotto del limite, rimuovi le metriche non necessarie.

Dopo il processo di onboarding non viene visualizzata alcuna SAP metrica

Utilizza le seguenti informazioni per scoprire perché le metriche SAP non vengono visualizzate nella dashboard dopo il processo di onboarding. Il primo passaggio consiste nel risolvere il motivo per cui i parametri SAP non vengono visualizzati utilizzando i log AWS Management Console o Exporter di un'istanza Amazon EC2. Successivamente, esamina l'output dell'errore per trovare una soluzione.

Risolvi il motivo per cui le metriche SAP non vengono visualizzate dopo l'onboarding

Puoi utilizzare i log AWS Management Console o exporter di un'istanza Amazon EC2 per la risoluzione dei problemi.

AWS Management Console

Risolvi i problemi: nessuna metrica SAP viene visualizzata dopo l'onboarding tramite la console

1. Apri la console all'indirizzo https://console.aws.amazon.com/systems-manager/. AWS Systems Manager
2. Nel riquadro di navigazione a sinistra, scegli State Manager.
3. In Associazioni, controlla lo stato del documento`AWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure`. Se lo stato è`Failed`, in Execution id, seleziona l'ID fallito e visualizza l'output.
4. In Associazioni, controlla lo stato del documento`AWS-ConfigureAWSPackage`. Se lo stato è`Failed`, in Execution id, seleziona l'ID fallito e visualizza l'output.

Exporter logs from Amazon EC2 instance

Risolvi i problemi: nessuna metrica SAP viene visualizzata dopo l'onboarding utilizzando i log di esportazione

1. Connect all'istanza Amazon EC2 su cui è in esecuzione il database SAP HANA.
2. Trova la convenzione di denominazione corretta per `WORKLOAD_SHORT_NAME` utilizzare il seguente comando. Utilizzerai questo nome breve nei due passaggi seguenti.

```
sudo systemctl | grep exporter
```

> **Note**
>
> Application Insights aggiunge un suffisso `WORKLOAD_SHORT_NAME` al nome del servizio a seconda del carico di lavoro in esecuzione. I nomi brevi per le implementazioni SAP HANA a nodo singolo, a più nodi e ad alta disponibilità sono, e. `HANA_SN` `HANA_MN` `HANA_HA`

3. Per verificare la presenza di errori nei log dei servizi di Exporter Manager, esegui il comando seguente sostituendolo `WORKLOAD_SHORT_NAME` con il nome breve che hai trovato in. Step 2

```
sudo journalctl -e --unit=prometheus-
hanadb_exporter_manager_WORKLOAD_SHORT_NAME.service
```

4. Se i registri di servizio di Exporter Manager non mostrano errori, verificate la presenza di errori nei log del servizio di esportazione eseguendo il comando seguente.

```
sudo journalctl -e --unit=prometheus-hanadb_exporter_WORKLOAD_SHORT_NAME.service
```

Risoluzione delle cause principali comuni della mancata visualizzazione delle metriche SAP dopo l'onboarding

Gli esempi seguenti descrivono come risolvere le cause principali comuni della mancata visualizzazione delle metriche SAP dopo l'onboarding.

- L'output mostra il seguente errore.

```
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-
cloudwatch-agent.d/default ...
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/
amazon-cloudwatch-agent.d/ssm_AmazonCloudWatch-ApplicationInsights-
SSMParameterForTESTCWEC2INSTANCEi0d88867f1f3e36285.tmp ...
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Under path : /metrics/append_dimensions | Error : Different
 values are specified for append_dimensions
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/disk | Error : Different
 values are specified for disk
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/mem | Error : Different
 values are specified for mem
```

```
2023/11/30 22:25:17 Configuration validation first phase failed. Agent version: 1.0.
 Verify the JSON input is only using features supported by this version.
```

Risoluzione: Application Insights sta tentando di configurare le stesse metriche preconfigurate come parte del file di configurazione dell' CloudWatch agente esistente. Rimuovi i file esistenti dal file di configurazione dell' CloudWatch agente esistente `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/` o rimuovi le metriche che causano il conflitto.

- L'output mostra il seguente errore.

```
Unable to find a host with system database, for more info rerun using -v
```

Risoluzione: il nome utente, la password o la porta del database potrebbero non essere corretti. Verifica che il nome utente, la password e la porta siano validi, quindi esegui nuovamente il processo di onboarding.

- L'output mostra il seguente errore.

```
This hdbcli installer is not compatible with your Python interpreter
```

Risoluzione — Aggiorna pip3 e wheel come mostrato nell'esempio seguente per Python 3.6.

```
python3.6 -m pip install --upgrade pip setuptools wheel
```

- L'output mostra il seguente errore.

```
Unable to install hdbcli using pip3. Please try to install it
```

Risoluzione: assicurati di aver seguito i `hdbclient` prerequisiti o esegui l'installazione `hdbclient` manualmente in pip3.

- L'output mostra il seguente errore.

```
Package 'boto3' requires a different Python: 3.6.15 not in '>= 3.7'
```

Risoluzione: per questa versione del sistema operativo è richiesto Python 3.8 o versione successiva. Controlla i prerequisiti di Python 3.8 e installalo.

- L'output mostra uno dei seguenti errori di installazione.

```
Can not execute `setup.py` since setuptools is not available in the build environment
```

oppure

```
[SSL: CERTIFICATE_VERIFY_FAILED]
```

Risoluzione: installa Python utilizzando i comandi SUSE Linux come mostrato nell'esempio seguente. L'esempio seguente installa l'ultima versione di Python 3.8.

```
wget https://www.python.org/ftp/python/3.8.<LATEST_RELEASE>/
Python-3.8.<LATEST_RELEASE>.tgz
tar xf Python-3.*
cd Python-3.*/
sudo zypper install make gcc-c++ gcc automake autoconf libtool
sudo zypper install zlib-devel
sudo zypper install libopenssl-devel libffi-devel
./configure --with-ensurepip=install
sudo make
sudo make install
sudo su
python3.8 -m pip install --upgrade pip setuptools wheel
```

Tutorial: configurare il monitoraggio per SAP NetWeaver

Questo tutorial dimostra come configurare Amazon CloudWatch Application Insights per configurare il monitoraggio per NetWeaver SAP. Puoi utilizzare i dashboard automatici di CloudWatch Application Insights per visualizzare i dettagli dei problemi, accelerare la risoluzione dei problemi e ridurre il tempo medio di risoluzione (MTTR) per i tuoi server di applicazioni SAP. NetWeaver

CloudWatch Argomenti di Application Insights per SAP NetWeaver

- Ambienti supportati
- Sistemi operativi supportati
- Funzionalità
- Prerequisiti
- Configura i server delle NetWeaver applicazioni SAP per il monitoraggio

- Gestisci il monitoraggio dei tuoi server di applicazioni SAP NetWeaver
- Visualizza e risolvi i problemi NetWeaver SAP rilevati da Application Insights CloudWatch
- Risoluzione dei problemi relativi a Application Insights per SAP NetWeaver

Ambienti supportati

CloudWatch Application Insights supporta l'implementazione di AWS risorse per i seguenti sistemi e modelli.

- Implementazione di sistemi NetWeaver standard SAP.
- Implementazioni NetWeaver distribuite SAP su più istanze Amazon EC2.
- Configurazione SAP NetWeaver ad alta disponibilità Cross-AZ: SAP NetWeaver con alta disponibilità configurato su due zone di disponibilità utilizzando il clustering SUSE/RHEL.

Sistemi operativi supportati

CloudWatch Application Insights for NetWeaver SAP è supportato sui seguenti sistemi operativi:

- Oracle Linux 8
- Red Hat Enterprise Linux 7.6
- Red Hat Enterprise Linux 7.7
- Red Hat Enterprise Linux 7.9
- Red Hat Enterprise Linux 8.1
- Red Hat Enterprise Linux 8.2
- Red Hat Enterprise Linux 8.4
- Red Hat Enterprise Linux 8.6
- SUSE Linux Enterprise Server 15 per SAP
- SUSE Linux Enterprise Server 15 SP1 per SAP
- SUSE Linux Enterprise Server 15 SP2 per SAP
- SUSE Linux Enterprise Server 15 SP3 per SAP
- SUSE Linux Enterprise Server 15 SP4 per SAP
- SUSE Linux Enterprise Server 12 SP4 per SAP

- SUSE Linux Enterprise Server 12 SP5 per SAP
- SUSE Linux Enterprise Server 15 tranne modelli ad alta disponibilità
- SUSE Linux Enterprise Server 15 SP1 tranne modelli ad alta disponibilità
- SUSE Linux Enterprise Server 15 SP2 tranne modelli ad alta disponibilità
- SUSE Linux Enterprise Server 15 SP3 tranne modelli ad alta disponibilità
- SUSE Linux Enterprise Server 15 SP4 tranne modelli ad alta disponibilità
- SUSE Linux Enterprise Server 12 SP4 tranne modelli ad alta disponibilità
- SUSE Linux Enterprise Server 12 SP5 tranne modelli ad alta disponibilità

Funzionalità

CloudWatch Application Insights for SAP NetWeaver 7.0x—7.5x (inclusa la piattaforma ABAP) offre le seguenti funzionalità:

- NetWeaver Rilevamento automatico del carico di lavoro SAP
- Creazione automatica di NetWeaver allarmi SAP basati su soglie statiche
- Riconoscimento automatico dei pattern di log SAP NetWeaver
- Dashboard Health per SAP NetWeaver
- Dashboard dei problemi per SAP NetWeaver

Prerequisiti

È necessario soddisfare i seguenti prerequisiti per configurare SAP NetWeaver con CloudWatch Application Insights:

- AWS Abilitazione di Systems Manager: installa SSM Agent sulle tue istanze Amazon EC2 e abilita le istanze per SSM. Per informazioni su come installare l'Agente SSM, consulta Configurazione di AWS Systems Manager nella Guida per l'utente di AWS Systems Manager.
- Ruoli delle istanze Amazon EC2: devi collegare i seguenti ruoli di istanza Amazon EC2 per configurare il monitoraggio SAP. NetWeaver
  - È necessario allegare il ruolo `AmazonSSMManagedInstanceCore` per abilitare Systems Manager. Per ulteriori informazioni, consulta Esempi di policy basate sull'identità nella AWS Systems Manager.

  - È necessario allegare la `CloudWatchAgentServerPolicy` policy per consentire la trasmissione dei parametri e dei log dell'istanza. CloudWatch Per ulteriori informazioni, consulta Creare ruoli e utenti IAM da utilizzare con l'agente. CloudWatch
- AWS gruppi di risorse: è necessario creare un gruppo di risorse che includa tutte le AWS risorse associate utilizzate dallo stack di applicazioni per l'onboarding delle applicazioni in CloudWatch Application Insights. Ciò include istanze Amazon EC2, Amazon EFS e volumi Amazon EBS che eseguono i tuoi server di applicazioni SAP. NetWeaver Se ci sono più NetWeaver sistemi SAP per account, ti consigliamo di creare un gruppo di risorse che includa AWS le risorse per ogni sistema SAP. NetWeaver Per ulteriori informazioni sulla creazione di gruppi di risorse, consulta la Guida per l'utente di AWS Resource Groups e tag.
- Autorizzazioni IAM: per gli utenti che non dispongono di accesso amministrativo, è necessario creare una policy AWS Identity and Access Management (IAM) che consenta ad Application Insights di creare un ruolo collegato al servizio e associarlo all'identità dell'utente. Per ulteriori informazioni su come creare la policy IAM, consulta Policy IAM.
- Ruolo collegato ai servizi: Application Insights utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato al servizio viene creato per te quando crei la tua prima applicazione Application Insights nella console Application Insights. Per ulteriori informazioni, consulta la pagina Utilizzo di ruoli collegati ai servizi per CloudWatch Application Insights.
- CloudWatch Agente Amazon: Application Insights installa e configura l' CloudWatch agente. Se hai installato un CloudWatch agente, Application Insights mantiene la tua configurazione. Per evitare un conflitto di unione, rimuovete la configurazione delle risorse che desiderate utilizzare in Application Insights dal file di configurazione dell' CloudWatch agente esistente. Per ulteriori informazioni, consulta Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.

Configura i server delle NetWeaver applicazioni SAP per il monitoraggio

Utilizza i seguenti passaggi per configurare il monitoraggio per i tuoi server di NetWeaver applicazioni SAP.

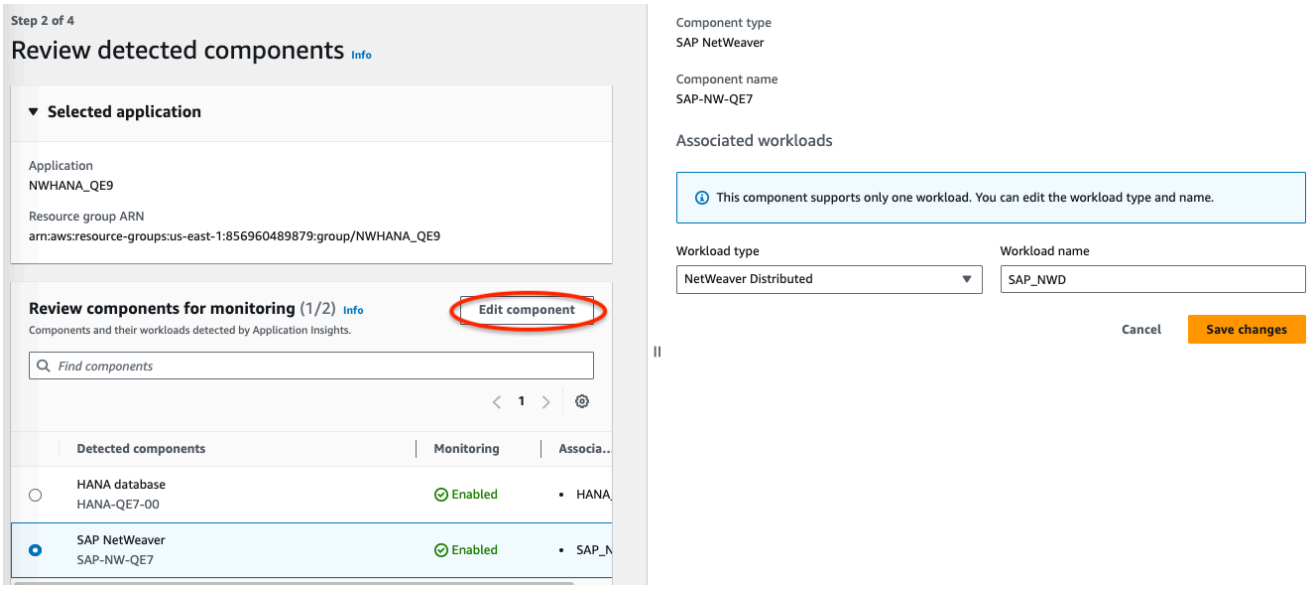
Per configurare il monitoraggio

1. Apri la CloudWatch console.
2. Nel pannello di navigazione a sinistra, in Insights (Approfondimenti) scegli Application Insights (Approfondimenti sulle applicazioni).

3. La pagina Application Insights visualizza l'elenco delle applicazioni che sono monitorate con Application Insights e lo stato di monitoraggio per ciascuna applicazione. Nell'angolo in alto a destra, seleziona Add an application (Aggiungi un'applicazione).
4. Nella pagina Specificare i dettagli dell'applicazione, dall'elenco a discesa in Gruppo di risorse, seleziona il gruppo di AWS risorse che hai creato che contiene le tue risorse NetWeaver SAP. Se non hai creato un gruppo di risorse per l'applicazione, puoi crearne uno scegliendo Create new resource group (Crea nuovo gruppo di risorse) sotto il menu a discesa Resource Group (Gruppo di risorse).
5. In Automatic monitoring of new resources (Monitoraggio automatico delle nuove risorse), seleziona la casella di controllo per consentire ad Application Insights di monitorare automaticamente le risorse aggiunte al gruppo di risorse dell'applicazione dopo l'onboarding.
6. In Monitora EventBridge eventi, seleziona la casella di controllo per integrare il monitoraggio di Application Insights con CloudWatch Events per ottenere informazioni da Amazon EBS, Amazon EC2 AWS CodeDeploy, Amazon ECS AWS Health , API e notifiche, Amazon RDS, Amazon S3 e. AWS Step Functions
7. In Integra con AWS Systems Manager OpsCenter, seleziona la casella di controllo accanto a Genera AWS Systems Manager OpsCenter OpsItems per azioni correttive per visualizzare e ricevere notifiche quando vengono rilevati problemi per le applicazioni selezionate. Per tenere traccia delle operazioni eseguite per risolvere gli elementi di lavoro operativi, denominati OpsItems, correlati alle AWS risorse, fornisci un argomento SNS ARN.
8. Facoltativamente, puoi inserire tag per aiutarti a identificare e organizzare le tue risorse. CloudWatch Application Insights supporta gruppi di risorse AWS CloudFormation basati su tag e stack, ad eccezione dei gruppi. Application Auto Scaling Per ulteriori informazioni, consulta Tag Editor nella Guida per l'utente di AWS Resource Groups .
9. Per rivedere i componenti rilevati, scegli Avanti.
10. Nella pagina Rivedi i componenti rilevati, sono elencati i componenti monitorati e i relativi carichi di lavoro rilevati automaticamente da CloudWatch Application Insights.

    - Per modificare il tipo e il nome del carico di lavoro, scegli Modifica componente.

      > **Note**
      >
      > I componenti che contengono un carico di lavoro NetWeaver distribuito o NetWeaver ad alta disponibilità rilevato supportano solo un carico di lavoro su un componente.



11. Seleziona Successivo.
12. Nella pagina Specify component details (Specifica i dettagli del componente), seleziona Next (Avanti).
13. Rivedi la configurazione del monitoraggio delle applicazioni e scegli Invia.
14. Si apre la pagina dei dettagli dell'applicazione, in cui puoi visualizzare le sezioni Riepilogo applicazione, Pannello di controllo, Componenti e Carichi di lavoro. Puoi visualizzare inoltre la Configuration history (Cronologia delle configurazioni), i Log patterns (Modelli di log) e i Tags (Tag) creati. Dopo aver inviato la domanda, CloudWatch Application Insights implementa tutte le metriche e gli allarmi per il NetWeaver sistema SAP, operazione che può richiedere fino a un'ora.

Gestisci il monitoraggio dei tuoi server di applicazioni SAP NetWeaver

Utilizza i seguenti passaggi per gestire il monitoraggio dei tuoi server di NetWeaver applicazioni SAP.

Per gestire il monitoraggio

1. Apri la CloudWatch console.
2. Nel pannello di navigazione a sinistra, in Insights (Approfondimenti) scegli Application Insights (Approfondimenti sulle applicazioni).
3. Scegli la scheda List view (Visualizzazione elenco).
4. La pagina Application Insights visualizza l'elenco delle applicazioni che sono monitorate con Application Insights e lo stato di monitoraggio per ciascuna applicazione.

5. Seleziona l'applicazione.
6. Scegli la scheda Components (Componenti).
7. In Componenti monitorati, seleziona il pulsante di opzione accanto al nome del componente. Quindi, seleziona Manage monitoring (Gestisci monitoraggio).
8. In Instance logs (Registri delle istanze), puoi aggiornare il percorso di log esistente, il set di modelli di log e il nome del gruppo di log. Inoltre, è possibile aggiungere fino a tre altri Log di applicazioni.
9. In Metriche, puoi selezionare le metriche SAP in base alle NetWeaver tue esigenze. I nomi delle NetWeaver metriche SAP hanno il prefisso. `sap` È possibile aggiungere fino a 40 parametri per componente.
10. In Allarmi personalizzati, puoi aggiungere altri allarmi che verranno monitorati da Application Insights. CloudWatch
11. Esamina la configurazione del monitoraggio dell'applicazione e scegli Save (Salva). Quando invii la configurazione, il tuo account aggiorna tutte le metriche e gli allarmi per i tuoi sistemi SAP. NetWeaver

Visualizza e risolvi i problemi NetWeaver SAP rilevati da Application Insights CloudWatch

Le seguenti sezioni forniscono i passaggi per aiutarti a risolvere gli scenari di risoluzione dei problemi più comuni che si verificano quando si configura il monitoraggio per SAP NetWeaver su Application Insights.

Argomenti sulla risoluzione dei problemi

- Problemi di connettività del NetWeaver database SAP
- Problemi di disponibilità delle NetWeaver applicazioni SAP

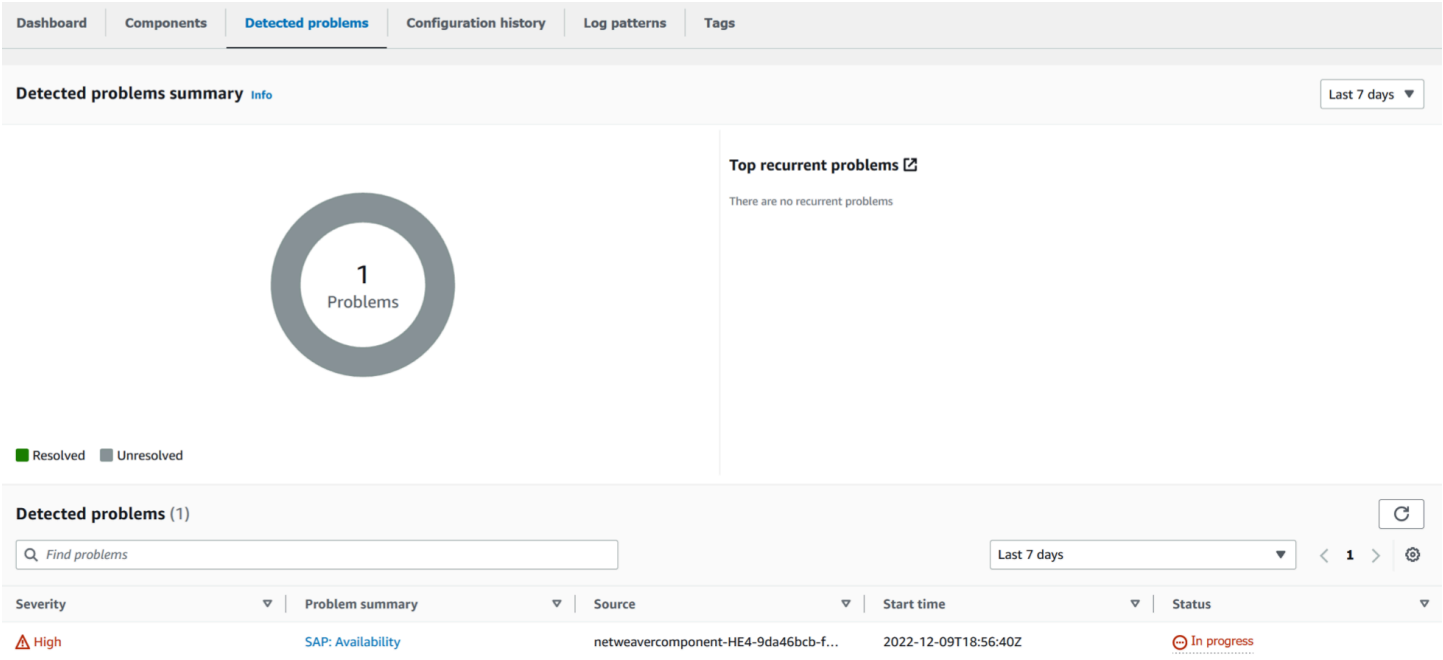
Problemi di connettività del NetWeaver database SAP

Descrizione

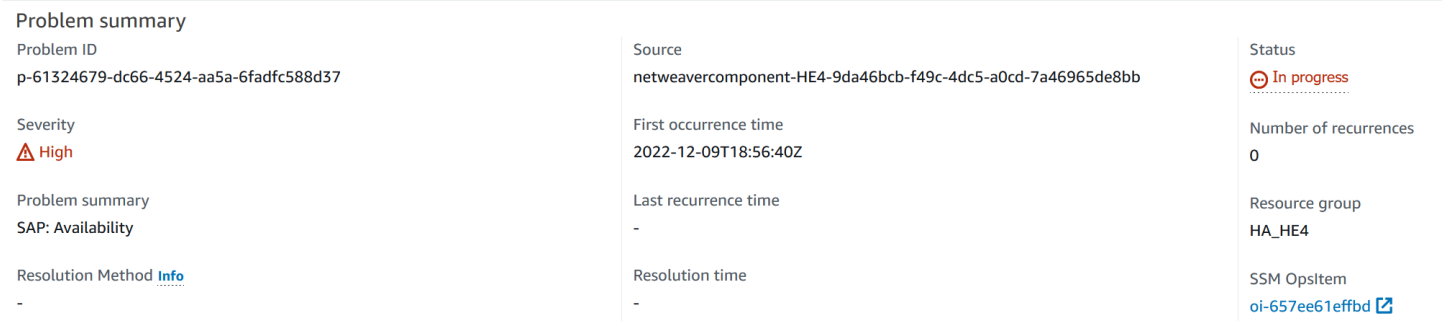
La tua NetWeaver applicazione SAP presenta problemi di connettività del database.

Causa

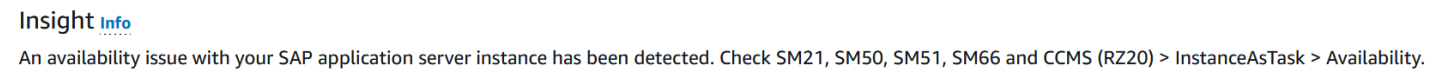
Puoi identificare il problema di connettività accedendo alla console di CloudWatch Application Insights e controllando il pannello di controllo dei problemi di SAP NetWeaver Application Insights. Seleziona il link in Problem summary (Riepilogo dei problemi) per visualizzare il problema specifico.



Nell'esempio seguente, in Problem summary (Riepilogo dei problemi), il problema è rappresentato da SAP: Availability (SAP: disponibilità).

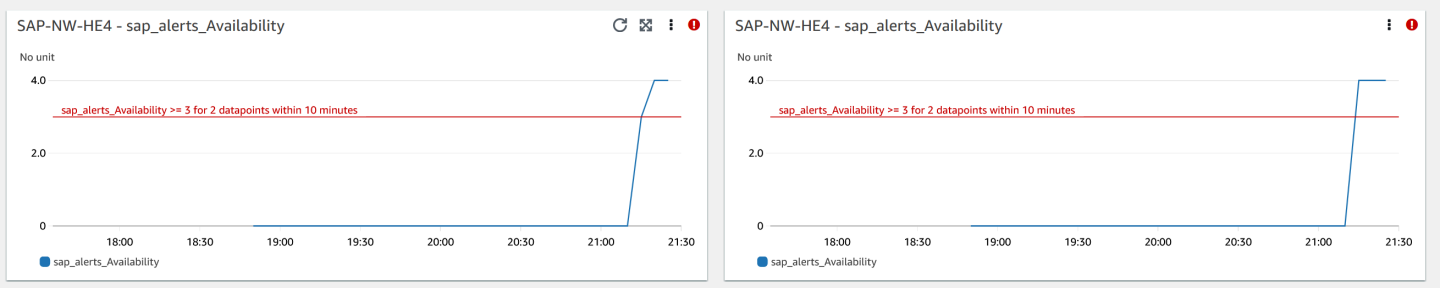


Subito dopo Problem summary (Riepilogo dei problemi), la sezione Insight (Approfondimenti) offre ulteriori informazioni sull'errore e su dove è possibile visualizzare altri dettagli relativi alle cause del problema.

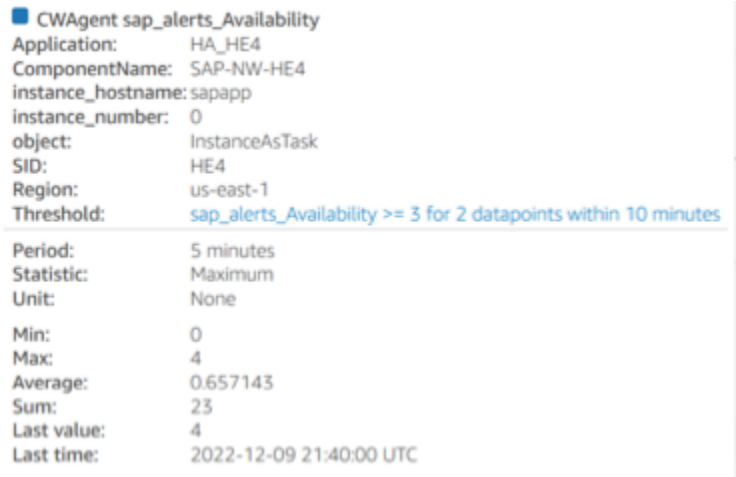


Nello stesso pannello di controllo dei problemi puoi visualizzare i log e le metriche correlati che sono stati raggruppati mediante il rilevamento dei problemi per consentirti di isolare la causa dell'errore.

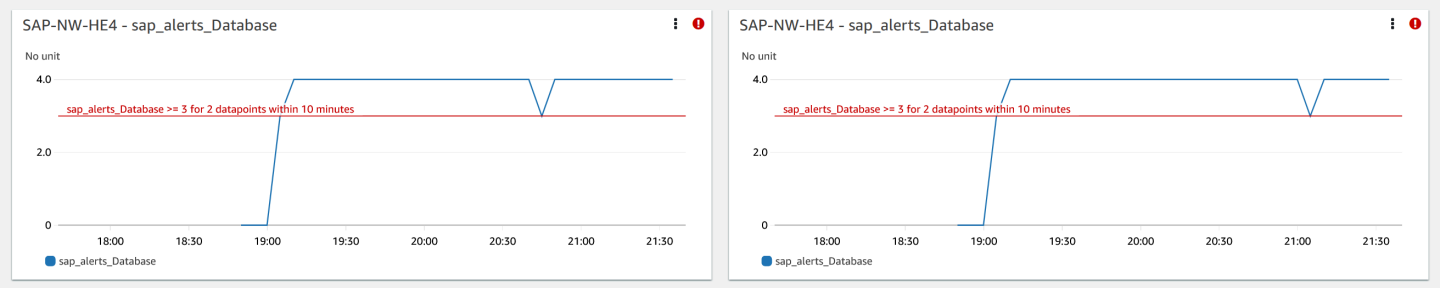
La `sap_alerts_Availability` metrica tiene traccia della disponibilità del NetWeaver sistema SAP nel tempo. Puoi utilizzare il tracciamento cronologico per mettere in relazione il momento in cui il parametro ha avviato uno stato di errore o superato la soglia di allarme. Nell'esempio seguente, c'è un problema di disponibilità con il sistema NetWeaver SAP. L'esempio mostra due allarmi perché sono presenti due istanze del server di applicazioni SAP ed è stato creato un allarme per ogni istanza.



Per ulteriori informazioni su ciascun allarme, passa con il mouse sul nome del parametro `sap_alerts_Availability`.



Nell'esempio seguente, il parametro `sap_alerts_Database` mostra che il livello del database presenta un problema o un errore. Questo allarme indica che SAP NetWeaver ha avuto problemi di connessione o comunicazione con il proprio database.



Poiché il database è una risorsa chiave per SAP NetWeaver, potresti ricevere molti allarmi correlati quando il database presenta un problema o un errore. Nell'esempio seguente, il database non è disponibile e di conseguenza vengono avviate le metriche `sap_alerts_FrontendResponseTime` e `sap_alerts_LongRunners`.

Risoluzione

Approfondimenti sulle applicazioni monitora il problema rilevato ogni ora. Se non ci sono nuove voci di registro correlate nei file di NetWeaver registro SAP, le voci di registro precedenti verranno considerate risolte. È necessario correggere eventuali condizioni di errore relative agli CloudWatch allarmi. Dopo aver corretto le condizioni di errore, gli allarmi e i log vengono ripristinati e l'allarme viene risolto. Quando tutti gli errori di CloudWatch registro e gli allarmi vengono risolti, Application Insights smette di rilevare gli errori e il problema viene risolto automaticamente entro un'ora. Ti consigliamo di risolvere tutte le condizioni di errore e gli allarmi del log in modo da visualizzare i problemi più recenti nel relativo pannello di controllo.

Nell'esempio seguente, il problema della disponibilità SAP è stato risolto.

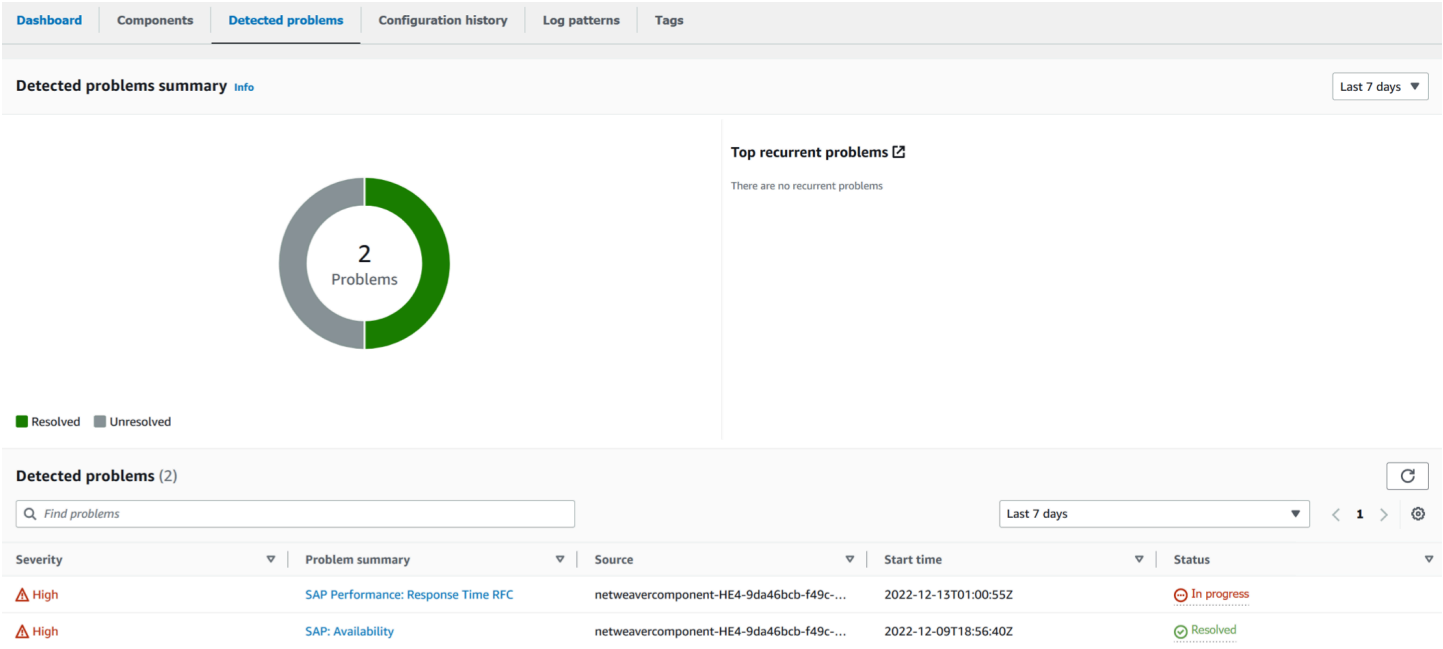
Problemi di disponibilità delle NetWeaver applicazioni SAP

Descrizione

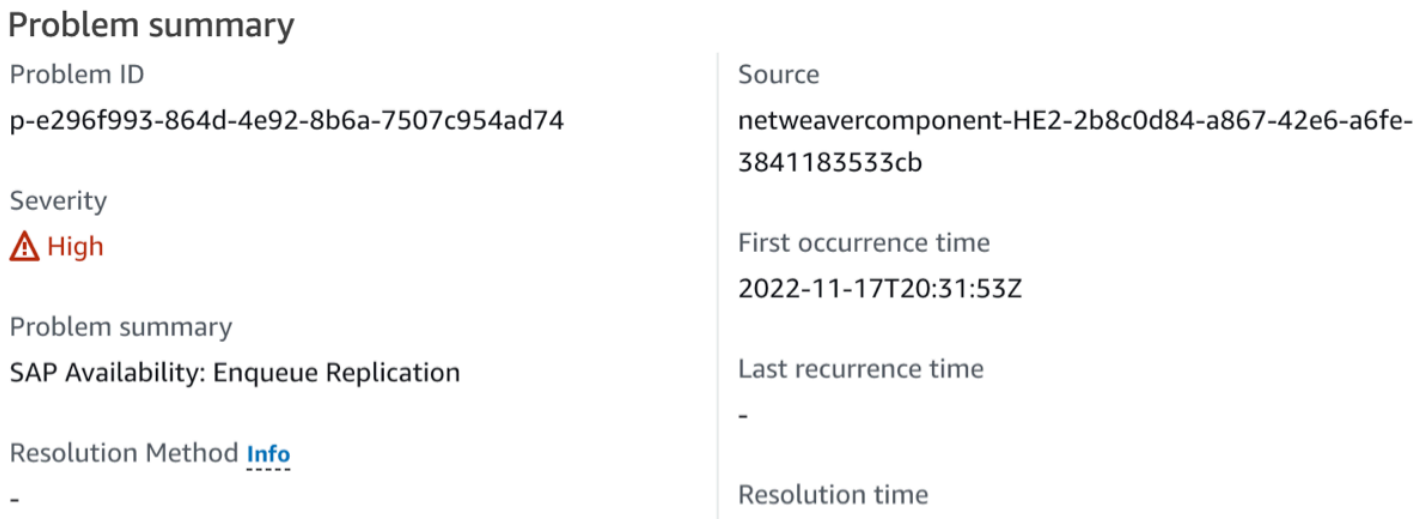
La replica di SAP NetWeaver High Availability Enqueue ha smesso di funzionare.

Causa

Puoi identificare il problema di connettività accedendo alla console di CloudWatch Application Insights e controllando la dashboard dei problemi di SAP NetWeaver Application Insights. Seleziona il link in Problem summary (Riepilogo dei problemi) per visualizzare il problema specifico.



Nell'esempio seguente, in Problem summary (Riepilogo dei problemi), viene riportato il problema High Availability Enqueue Replication (Replica accodamento a disponibilità elevata).

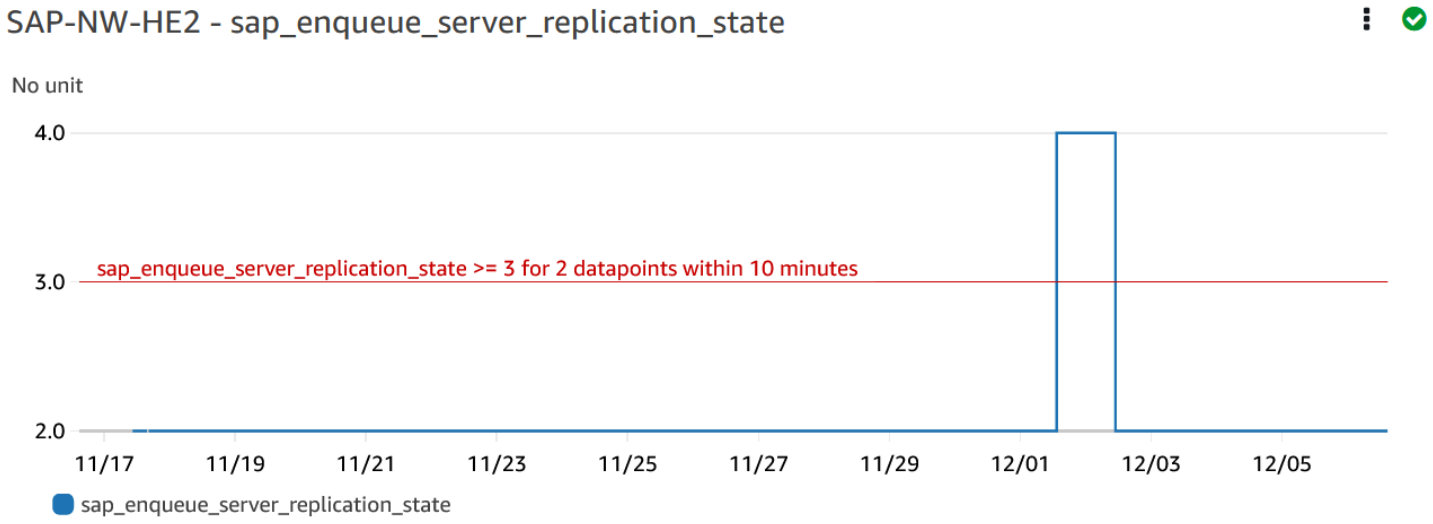


Subito dopo Problem summary (Riepilogo dei problemi), la sezione Insight (Approfondimenti) offre ulteriori informazioni sull'errore e su dove è possibile visualizzare altri dettagli relativi alle cause del problema.

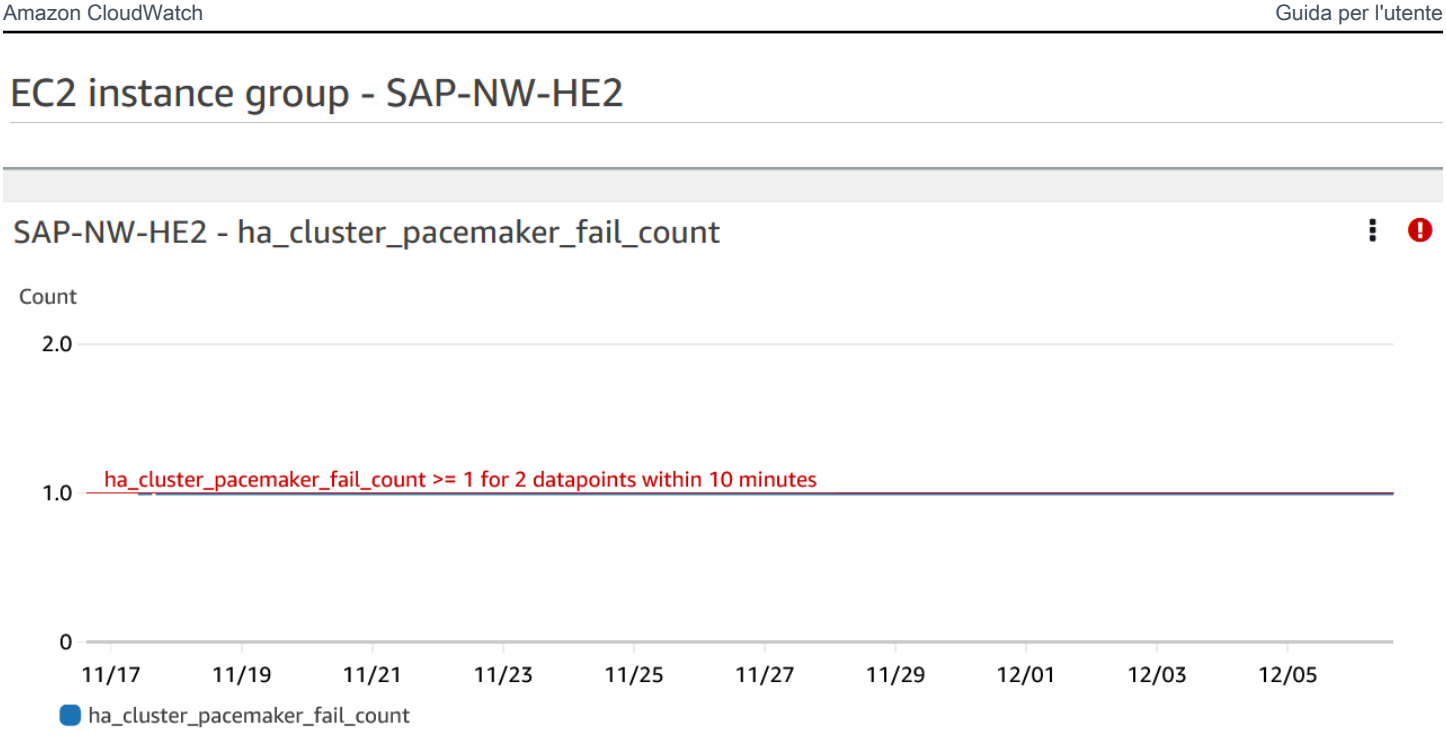
Insight Info

An issue with your SAP enqueue replication (ERS) state has been detected. Check that your enqueue replication is working with SAP transactions, such as SMENQ or the ensmon command.

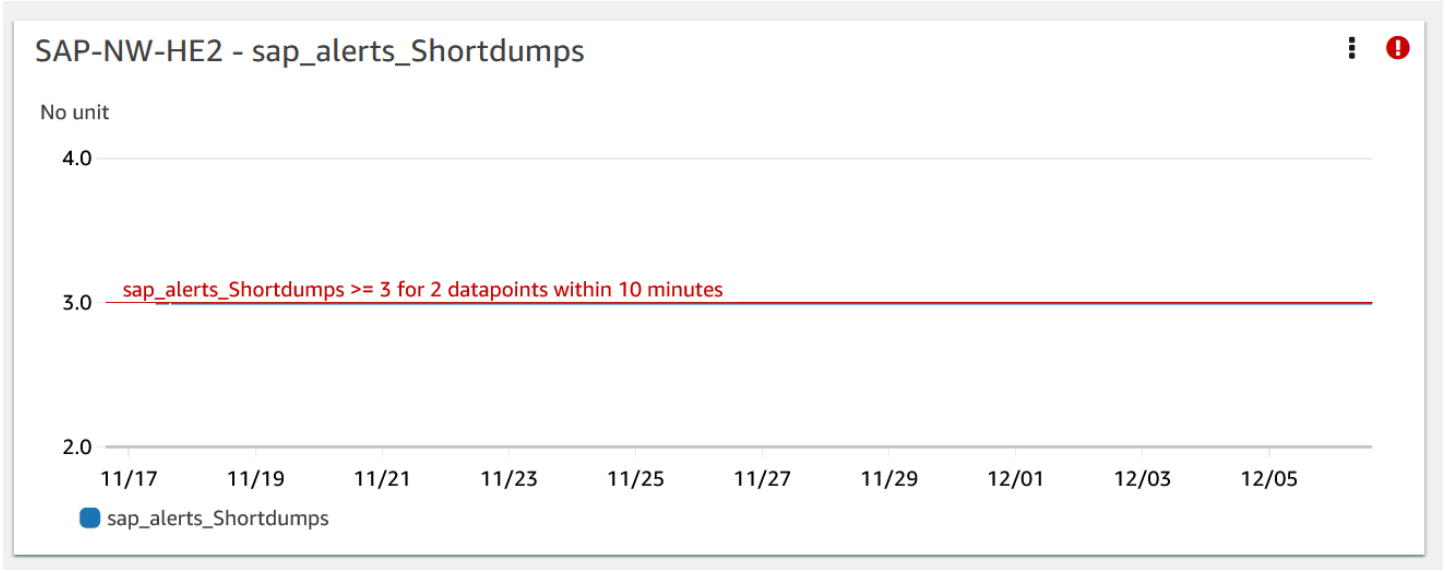
L'esempio seguente mostra il pannello di controllo dei problemi in cui vengono raggruppati i log e le metriche per consentirti di isolare le cause dell'errore. Il parametro `sap_enqueue_server_replication_state` tiene traccia del valore nel tempo. Puoi utilizzare il tracciamento cronologico per mettere in relazione il momento in cui il parametro ha avviato uno stato di errore o superato la soglia di allarme.



Nell'esempio seguente, il parametro `ha_cluster_pacemaker_fail_count` mostra che nel cluster pacemaker a disponibilità elevata si è verificato un errore di risorsa. Le risorse specifiche del pacemaker con un numero di errori maggiore o uguale a uno sono indicate nel pannello di controllo del componente.

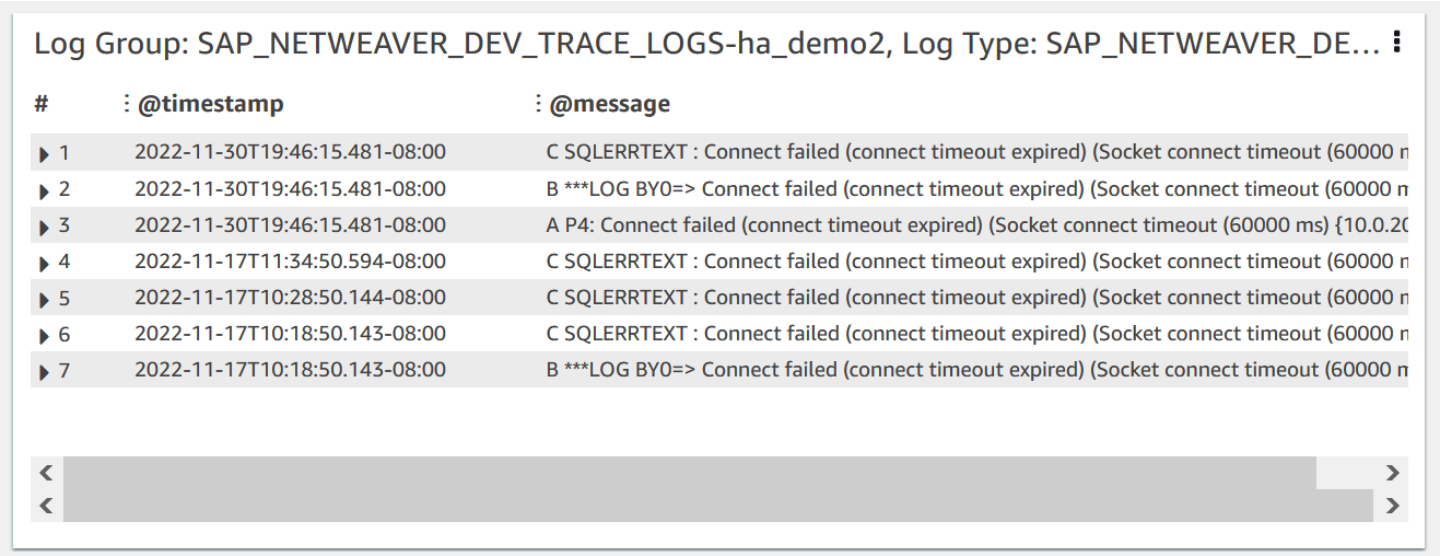


L'esempio seguente mostra il parametro `sap_alerts_Shortdumps`, che indica una riduzione delle prestazioni dell'applicazione SAP quando è stato rilevato il problema.

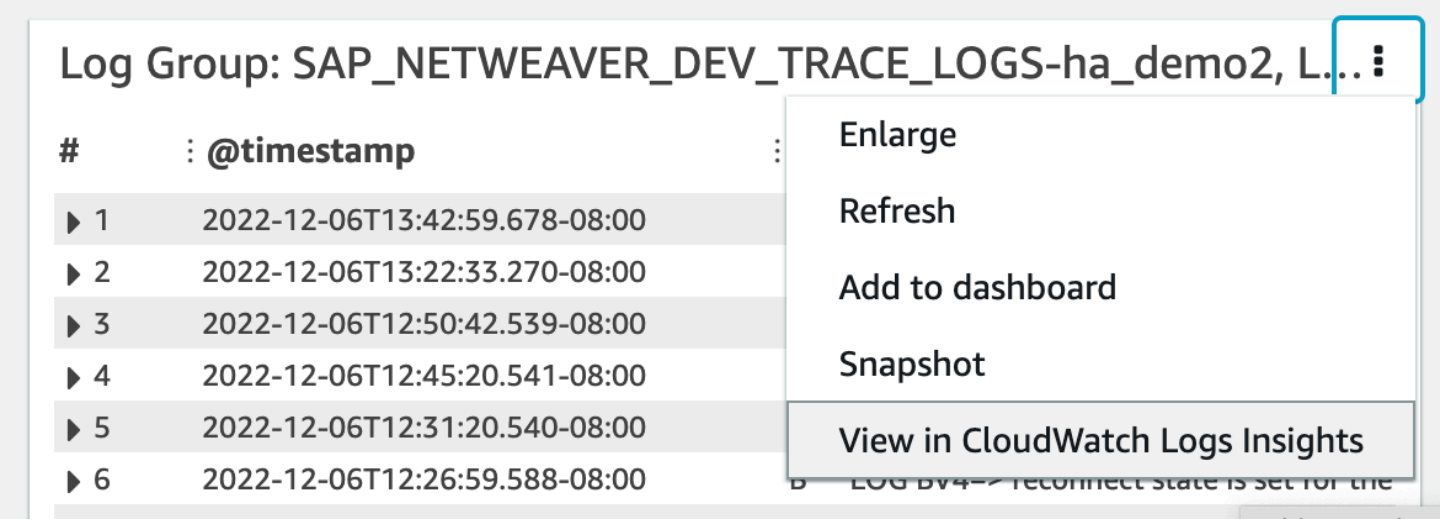


Log

Le voci di registro sono utili per comprendere meglio i problemi che si sono verificati a NetWeaver livello SAP quando il problema è stato rilevato. Il widget del gruppo di log nel pannello di controllo dei problemi mostra l'ora specifica del problema.



Per visualizzare informazioni dettagliate sui log, seleziona i tre punti verticali nell'angolo in alto a destra e seleziona Visualizza in Logs Insights. CloudWatch

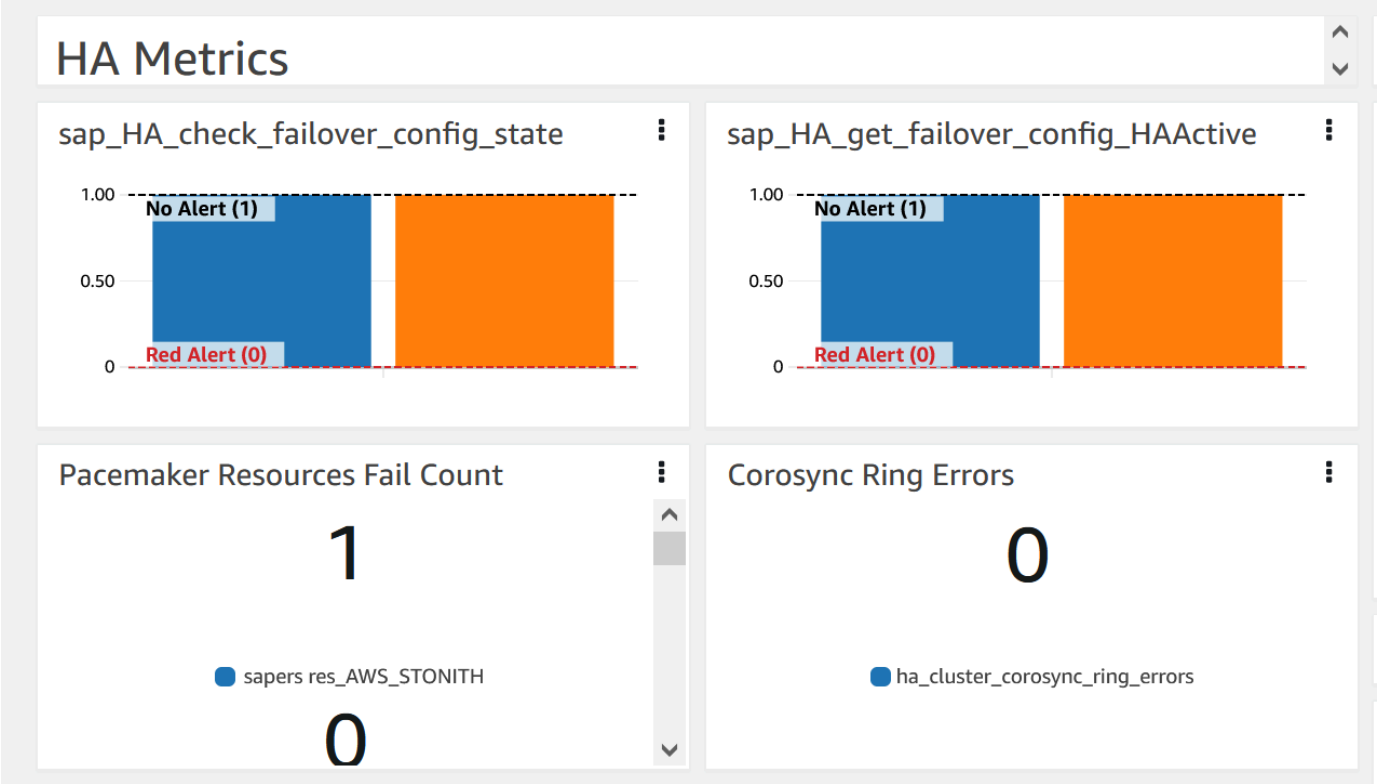


Utilizza la procedura seguente per ottenere maggiori informazioni su metriche e allarmi visualizzati nel pannello di controllo dei problemi.

Per ottenere maggiori informazioni su metriche e allarmi

1. Apri la CloudWatch console.
2. Nel pannello di navigazione a sinistra, in Insights (Approfondimenti) seleziona Application Insights. Scegli quindi la scheda List view (Visualizzazione elenco) e seleziona l'applicazione.
3. Seleziona la scheda Components (Componenti). Quindi, seleziona il NetWeaver componente SAP su cui desideri ottenere ulteriori informazioni.

L'esempio seguente mostra la sezione HA Metrics (Metriche HA) con il parametro `ha_cluster_pacemaker_fail_count` visualizzato nel pannello di controllo dei problemi.

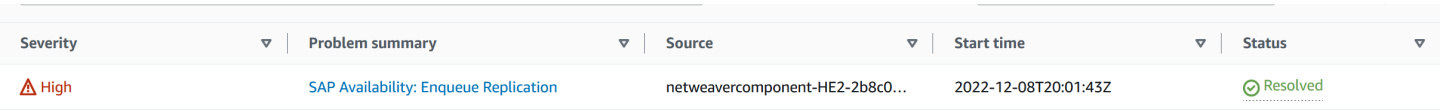


Risoluzione

Approfondimenti sulle applicazioni monitora il problema rilevato ogni ora. Se non ci sono nuove voci di registro correlate nei file di NetWeaver registro SAP, le voci di registro precedenti verranno considerate risolte. È necessario correggere eventuali condizioni di errore relative a questo problema.

Per l'`sap_alerts_Shortdumps`allarme, è necessario risolvere l'avviso nel NetWeaver sistema SAP utilizzando il codice di transazione `RZ20 # R3Abap # Shortdumps` per accedere all'avviso CCMS. Per ulteriori informazioni sugli avvisi CCMS, consulta il sito Web di SAP. Risolvi tutti gli avvisi CCMS nella struttura Shortdumps. Dopo che tutti gli avvisi sono stati risolti nel NetWeaver sistema SAP, CloudWatch non riporta più la metrica in stato di allarme.

Quando tutti gli errori di CloudWatch registro e gli allarmi vengono risolti, Application Insights smette di rilevare gli errori e il problema viene risolto automaticamente entro un'ora. Ti consigliamo di risolvere tutte le condizioni di errore e gli allarmi del log in modo da visualizzare i problemi più recenti nel relativo pannello di controllo. Nell'esempio seguente, il problema High Availability Enqueue Replication (Replica accodamento a disponibilità elevata) di SAP Netweaver è stato risolto.



Risoluzione dei problemi relativi a Application Insights per SAP NetWeaver

Questa sezione fornisce i passaggi per aiutarti a risolvere gli errori comuni restituiti dal pannello di controllo di Application Insights.

Impossibile aggiungere più di 60 parametri del monitor

Errore resituito:`Component cannot have more than 60 monitored metrics.`

Causa principale:`The current metric limit is 60 monitor metrics per component.`

Resolution (Risoluzione): Rimuovere i parametri che non sono necessari per rispettare il limite.

Le metriche SAP non vengono visualizzati nel pannello di controllo dopo il processo di onboarding

Causa principale: il pannello di controllo dei componenti utilizza un periodo di cinque minuti per aggregare i punti dati.

Risoluzione: tutte le metriche dovrebbero essere visualizzate nel pannello di controllo dopo cinque minuti.

Le metriche e gli allarmi SAP non vengono visualizzati nel pannello di controllo

Utilizza la procedura seguente per identificare il motivo per cui le metriche e gli allarmi SAP non vengono visualizzati nel pannello di controllo dopo il processo di onboarding.

Per identificare il problema con metriche e allarmi

1. Apri la CloudWatch console.
2. Nel pannello di navigazione a sinistra, in Insights (Approfondimenti) seleziona Application Insights. Scegli quindi la scheda List view (Visualizzazione elenco) e seleziona l'applicazione.
3. Scegli la scheda Configuration history (Cronologia delle configurazioni).
4. Se alcuni punti dati delle metriche non sono presenti, verifica la presenza di errori relativi a `prometheus-sap_host_exporter`.
5. Se non trovi alcun errore nel passaggio precedente, esegui la connessione all'istanza di Linux. Per le implementazioni ad alta disponibilità, connettiti all'istanza Amazon EC2 del cluster principale.

6. All'interno dell'istanza, verifica che l'esportatore sia in esecuzione utilizzando il seguente comando. La porta predefinita è `9680`. Se stai utilizzando una porta diversa, sostituisci `9680` con la porta in uso.

```
curl localhost:9680/metrics
```

   Se non viene restituito alcun dato, l'esportatore non è stato avviato.

7. Per trovare la convenzione di denominazione corretta da utilizzare `WORKLOAD_SHORT_NAME` nei due passaggi successivi, esegui il comando seguente.

> **Note**
>
> Application Insights aggiunge un suffisso,`WORKLOAD_SHORT_NAME`, al nome del servizio in base al carico di lavoro in esecuzione. I nomi brevi per le implementazioni NetWeaver Distributed, Standard e High Availability sono`SAP_NWD`, e. `SAP_NWS` `SAP_NWH`

```
sudo systemctl | grep exporter
```

8. Per verificare la presenza di errori nei registri del servizio di esportazione, esegui il comando seguente:

```
sudo journalctl -e --unit=prometheus-sap_host_exporter_WORKLOAD_SHORT_NAME.service
```

9. Per verificare la presenza di errori nei registri del servizio di Exporter Manager, esegui il comando seguente:

```
sudo journalctl -e --unit=prometheus-
sap_host_exporter_manager_WORKLOAD_SHORT_NAME.service
```

> **Note**
>
> Questo servizio deve essere sempre attivo e funzionante.

   Se questo comando non restituisce alcun errore, continua con il passaggio successivo.

10. Per avviare manualmente l'esportatore, esegui il comando seguente. Quindi, controlla l'output dell'esportatore.

    ```
    sudo /opt/aws/sap_host_exporter/sap_host_exporter
    ```

    Puoi uscire dal processo di esportazione dopo aver verificato la presenza di errori.

Causa principale: le cause per questo problema sono diverse. Una causa comune è che l'esportatore non è in grado di connettersi a una delle istanze del server di applicazioni.

Resolution (Risoluzione)

Utilizza la procedura seguente per connettere l'esportatore alle istanze del server di applicazioni. Verificherai che l'istanza dell'applicazione SAP sia in esecuzione e utilizzerai SAPControl per connetterti all'istanza.

Per connettere l'esportatore alle istanze del server di applicazioni

1. Nella tua istanza Amazon EC2, esegui il comando seguente per verificare che l'applicazione SAP sia in esecuzione.

   ```
   sapcontrol -nr <App_InstNo> -function GetProcessList
   ```

2. È necessario stabilire una connessione SAPControl funzionante. Se la connessione SAPControl non funziona, individua la causa principale del problema nell'istanza dell'applicazione SAP pertinente.
3. Per avviare manualmente l'esportatore dopo aver risolto il problema di connessione SAPControl, esegui il comando seguente:

   ```
   sudo systemctl start prometheus-sap_host_exporter.service
   ```

4. Se non è possibile risolvere il problema di connessione SAPControl, utilizza la procedura seguente come correzione temporanea.

   a. Apri la AWS Systems Manager console.

   b. Nel pannello di navigazione a sinistra, scegli State Manager.

   c. In Associazioni cerca l'associazione del NetWeaver sistema SAP.

```
Association Name: Equal: AWS-ApplicationInsights-
SSMSAPHostExporterAssociationForCUSTOMSAPNW<SID>-1
```

d. Seleziona l'Association id (ID associazione).

e. Scegli la scheda Parameters (Parametri) e rimuovi il numero del server dell'applicazione da additionalArguments.

f. Scegli Apply Association Now (Applica l'associazione ora).

> **Note**
>
> Si tratta di una correzione temporanea. Se vengono apportate modifiche alle configurazioni di monitoraggio del componente, l'istanza verrà aggiunta nuovamente.

Visualizza e risolvi i problemi rilevati da Amazon Application Insights CloudWatch

Gli argomenti di questa sezione forniscono informazioni dettagliate sui problemi rilevati e le informazioni dettagliate visualizzate da Application Insights. Fornisce inoltre risoluzioni suggerite per i problemi rilevati con il tuo account o la tua configurazione.

Argomenti sulla risoluzione dei problemi

- CloudWatch panoramica della console
- Pagina di riepilogo dei problemi di Application Insights
- CloudWatch errori di unione dei conflitti tra agenti
- Gli allarmi non vengono creati
- Feedback
- Errori di configurazione

CloudWatch panoramica della console

Una panoramica dei problemi che influiscono sulle applicazioni monitorate è disponibile nel riquadro CloudWatch Application Insights nella pagina di panoramica della CloudWatch console. Per ulteriori informazioni, consulta Inizia a usare Amazon CloudWatch Application Insights.

Il riquadro di panoramica di CloudWatch Application Insights mostra quanto segue:

- La gravità dei problemi rilevati: alta/media/bassa
- Un breve riepilogo del problema
- La fonte del problema
- L'ora in cui il problema si è verificato
- La stato di risoluzione del problema
- Il Gruppo di risorse interessato

Per espandere i dettagli di un problema specifico, in Problem Summary (Riepilogo problema), seleziona la descrizione del problema. Un pannello di controllo dettagliato visualizza approfondimenti del problema e anomalie parametri correlati e snippet di errori di log. Da qui, è possibile fornire un feedback sulla pertinenza dell'informazione dettagliata se è utile.

Se viene rilevata una nuova risorsa non configurata, la descrizione del riepilogo del problema consente di accedere alla procedura guidata Edit configuration (Modifica configurazione) per configurare la nuova risorsa. Se necessario, puoi visualizzare o modificare la configurazione del gruppo di risorse scegliendo View/edit configuration (Visualizza/modifica configurazione) nell'angolo in alto a destra del pannello di controllo dettagliato.

Per tornare alla panoramica, scegli Torna alla panoramica, che si trova accanto all'intestazione dettagliata del dashboard di CloudWatch Application Insights.

Pagina di riepilogo dei problemi di Application Insights

Pagina di riepilogo dei problemi di Application Insights

CloudWatch Application Insights fornisce le seguenti informazioni sui problemi rilevati nella pagina di riepilogo dei problemi:

- Un breve riepilogo del problema
- L'ora e la data di inizio del problema
- La gravità del problema: forte/media/bassa
- Lo stato del problema rilevato: in corso/risolto
- Approfondimenti: approfondimenti generati automaticamente sul problema rilevato e la possibile causa principale

- Feedback sugli approfondimenti: feedback che hai fornito sull'utilità degli approfondimenti generati da CloudWatch Application Insights
- Osservazioni correlate: una vista dettagliata delle anomalie parametro e frammenti di errore di log pertinenti correlati al problema su vari componenti dell'applicazione

CloudWatch errori di unione dei conflitti tra agenti

CloudWatch Application Insights installa e configura l' CloudWatch agente sulle istanze del cliente. Ciò include la creazione di un file di configurazione CloudWatch dell'agente con configurazioni per metriche o log. Un conflitto di fusione può verificarsi se l'istanza di un cliente ha già un file di configurazione CloudWatch dell'agente con configurazioni diverse definite per gli stessi parametri o log. Per risolvere il conflitto di unione, segui la procedura riportata di seguito:

1. Identifica i file di configurazione CloudWatch dell'agente sul tuo sistema. Per ulteriori informazioni sulle posizioni dei file, consulta CloudWatch file e posizioni degli agenti.
2. Rimuovi le configurazioni delle risorse che desideri utilizzare in Application Insights dal file di configurazione dell' CloudWatch agente esistente. Se desideri utilizzare solo le configurazioni di Application Insights, elimina i file di configurazione dell' CloudWatch agente esistenti.

Gli allarmi non vengono creati

Per alcuni parametri, Application Insights prevede la soglia di allarme in base ai dati precedenti relativi al parametro. Per abilitare questa previsione, devono essere soddisfatti i seguenti criteri.

- Punti dati recenti: devono essere presenti almeno 100 punti dati delle ultime 24 ore. I punti dati non devono essere continui e possono essere sparsi nell'arco di 24 ore.
- Dati storici: devono essere presenti almeno 100 punti dati che coprono l'intervallo di tempo compreso tra 15 giorni prima della data corrente e 1 giorno prima della data corrente. I punti dati non devono essere continui e possono essere sparsi nell'arco di 15 giorni.

> **Note**
>
> Per alcuni parametri, Application Insights ritarda la creazione di allarmi fino al soddisfacimento delle condizioni precedenti. In questo caso, si verifica un evento della

cronologia di configurazione che indica che il parametro non dispone di punti dati sufficienti per stabilire la soglia di allarme.

Feedback

Feedback

Puoi fornire feedback sugli approfondimenti generati automaticamente sui problemi rilevati designandoli come utili o non utili. Il feedback sugli approfondimenti, insieme alla diagnostica dell'applicazione (anomalie parametri ed eccezioni di log), viene utilizzato per migliorare il rilevamento futuro di problemi simili.

Errori di configurazione

CloudWatch Application Insights utilizza la configurazione dell'utente per creare telemetrie di monitoraggio per i componenti. Quando Application Insights rileva un problema con l'account o la configurazione, nel campo Remarks (Osservazioni) del ripelogo dell'applicazione vengono fornite informazioni su come risolvere il problema di configurazione per l'applicazione.

La tabella seguente mostra le soluzioni suggerite per osservazioni specifiche.

Remarks	Risoluzione suggerita	Note aggiuntive
La quota per CloudFormation è già stata raggiunta.	Application Insights crea uno CloudFormation stack per ogni applicazione per gestire l'installazione e la configura zione degli CloudWatch agenti per tutti i componenti dell'appl icazione. Per impostazione predefinita, ogni AWS account può avere 2000 pile. Vedi Limiti di AWS CloudForm ation. Per risolvere questo problema, aumenta il limite per gli CloudFormation stack.	N/A

Remarks	Risoluzione suggerita	Note aggiuntive
Nessun ruolo dell'istanza SSM sulle seguenti istanze.	Affinché Application Insights sia in grado di installare e configurare l' CloudWatch agente sulle istanze dell'applicazione, è necessario associare AmazonSSMManagedInstanceCore e CloudWatchAgentServerPolicy le policy al ruolo dell'istanza.	Application Insights chiama l'DescribeInstanceInformation API SSM per ottenere l'elenco delle istanze con autorizzazione SSM. Dopo aver associato il ruolo all'istanza, SSM impiega del tempo per includere l'istanza nel risultato. DescribeInstanceInformation Finché SSM non include l'istanza nel risultato, l'errore NO_SSM_INSTANCE_ROLE rimane presente per l'applicazione.
Nuovi componenti possono richiedere la configurazione.	Application Insights rileva che sono presenti nuovi componenti nel Gruppo di risorse dell'applicazione. Per risolvere questo problema, configura i nuovi componenti di conseguenza.	N/A

Log e parametri supportati da Amazon Application Insights CloudWatch

I seguenti elenchi mostrano i log e i parametri supportati per Amazon CloudWatch Application Insights.

CloudWatch Application Insights supporta i seguenti log:

- Log di Microsoft Internet Information Services (IIS)
- Log di errore per SQL Server su EC2
- Log di applicazioni .NET personalizzate, ad esempio Log4Net

- Log di eventi Windows, inclusi i log di Windows (sistema, applicazione e sicurezza) e il log delle applicazioni e dei servizi
- Amazon CloudWatch Logs per AWS Lambda
- Log di errore e log lento per RDS MySQL, Aurora MySQL e MySQL su EC2
- Log Postgresql per PostgreSQL RDS e PostgreSQL su EC2
- Amazon CloudWatch Logs per AWS Step Functions
- Log di esecuzione e log di accesso (JSON, CSV e XML, ma non CLF) per le fasi REST API in API Gateway
- Log di Prometheus JMX exporter (EMF)
- Log di avvisi e log listener per Oracle su Amazon RDS e Oracle su Amazon EC2
- Instradamento dei log dei container dai contenitori Amazon ECS all' CloudWatch utilizzo `awslogs` del driver di registro.
- Instradamento dei log dei container dai container Amazon ECS all' CloudWatch utilizzo del FireLens Container Log Router.
- Instradamento dei log dei container da Amazon EKS o Kubernetes in esecuzione su Amazon EC2 all' CloudWatch utilizzo del processore di log Fluent Bit o Fluentd con Container Insights.
- Registri di traccia e errori di SAP HANA
- Registri pacemaker HA
- Log del server SAP ASE
- Log del server di backup SAP ASE
- Log del server di replica SAP ASE
- Log degli agenti SAP ASE RMA
- Log di SAP ASE Fault Manager
- NetWeaver Registri di traccia degli sviluppatori SAP
- Metriche di processo per i processi Windows che utilizzano il plug-in proctstat per agente CloudWatch
- Log delle query DNS pubbliche per la zona ospitata
- Amazon Route 53 Resolver Registri delle interrogazioni DNS

CloudWatch Application Insights supporta le seguenti classi di log:

- Standard: Amazon CloudWatch Application Insights richiede che i gruppi di log siano configurati con la classe di log CloudWatch Logs Standard per consentire il monitoraggio.

CloudWatch Application Insights supporta i parametri per i seguenti componenti dell'applicazione:

- Amazon Elastic Compute Cloud (EC2)
  - CloudWatch metriche integrate
  - CloudWatch metriche degli agenti (server Windows)
  - CloudWatch metriche dei processi degli agenti (server Windows)
  - CloudWatch metriche degli agenti (server Linux)
- Elastic Block Store (EBS)
- Amazon Elastic File System (Amazon EFS)
- Elastic Load Balancer (ELB)
- ELB applicazione
- Gruppi Amazon EC2 Auto Scaling
- Amazon Simple Queue Server (SQS)
- Amazon Relational Database Service (RDS)
  - Istanze database RDS
  - Cluster di database RDS
- AWS Lambda funzione
- Tabella Amazon DynamoDB
- Bucket Amazon S3
- AWS Step Functions
  - Execution-level
  - Attività
  - Funzione Lambda
  - Integrazione dei servizi
  - API Step Functions
- Fasi REST API di API Gateway
- SAP HANA
- SAP ASE
- SAP ASE High Availability su Amazon EC2

- SAP NetWeaver
- Cluster HA
- Java
- Amazon Elastic Container Service (Amazon ECS)
  - CloudWatch metriche integrate
  - Parametri di Container Insights
  - Parametri di Container Insights Prometheus
- Kubernetes attivo AWS
  - Parametri di Container Insights
  - Parametri di Container Insights Prometheus
- Amazon FSx
- Amazon VPC
- Gateway NAT di Amazon VPC
- Controllo dell'integrità di Amazon Route 53
- Zona ospitata di Amazon Route 53
- Amazon Route 53 Resolver endpoint
- AWS Network Firewall gruppo di regole
- AWS Network Firewall associazione di gruppi di regole
- Parametri con requisiti di punti dati
  - AWS/ApplicationELB
  - AWS/ AutoScaling
  - <shared id="xxx"/>/EC2
  - Elastic Block Store (EBS)
  - AWS/ELB
  - AWS/RDS
  - AWS/Lambda
  - AWS/SQS
  - AWS/CWAgent
  - AWS/DynamoDB
  - AWS/S3

  - AWS/States
  - AWS/ ApiGateway
  - AWS/SNS
- Parametri consigliati
- Parametri del contatore di prestazioni

Amazon Elastic Compute Cloud (EC2)

CloudWatch Application Insights supporta le seguenti metriche:

Metriche

- CloudWatch metriche integrate
- CloudWatch metriche degli agenti (server Windows)
- CloudWatch metriche dei processi degli agenti (server Windows)
- CloudWatch metriche degli agenti (server Linux)

CloudWatch metriche integrate

CPU CreditBalance

CPU CreditUsage

CPU SurplusCreditBalance

CPU SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBS% ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Istanza

StatusCheckFailed_Sistema

CloudWatch metriche degli agenti (server Windows)

Numero di eccezioni .NET CLR generate

Numero di eccezioni .NET CLR Exceps Thrown/sec

Numero di eccezioni .NET CLR Filters/sec

Numero di eccezioni .NET CLR Finallys/sec

Eccezioni .NET CLR Throw to Catch Depth/sec

Numero di Interop di .NET CLR di CCW

Numero di Interop .NET CLR di stub

Numero di Interop .NET CLR delle esportazioni TLB/sec

Numero di Interop .NET CLR delle importazioni TLB/sec

Numero di Interop .NET CLR di marshalling

Percentuale di tempo in Jit .NET CLR in Jit

Errori Jit standard .NET CLR Jit

Percentuale tempo di caricamento .NET CLR

Velocità di caricamento .NET CLR di errori di caricamento

Frequenza di contenzio.NET CLR/sec LocksAndThreads

Lunghezza della coda.NET CLR/sec LocksAndThreads

Numero di byte totali impegnati memoria CLR .NET

% di tempo in memoria .NET CLR in GC

.NET CLR Networking 4.0.0.0 Tempo di coda medio HttpWebRequest

.NET CLR Networking 4.0.0.0 interrotto/sec HttpWebRequests

.NET CLR Networking 4.0.0.0 non HttpWebRequests riuscita/sec

.NET CLR Networking 4.0.0.0 in HttpWebRequests coda/sec

Errori di ping del processo di lavoro totali APP_POOL_WAS Total

Riavvii applicazione ASP.NET

Tempo del processore gestito in percentuale delle applicazioni (stimato) ASP.NET

Errori totali/sec delle applicazioni ASP.NET

Errori delle applicazioni ASP.NET non gestiti durante l'esecuzione/sec

Richieste delle applicazioni ASP.NET nella coda delle applicazioni

Richieste delle applicazioni ASP.NET/sec

Tempo di attesa delle richieste ASP.NET

Richieste ASP.NET in coda

Code di richieste di assistenza HTTP CurrentQueueSize

LogicalDisk % di spazio libero

Memory: byte impegnati della percentuale di memoria in uso

Mbyte di memoria disponibili

Pagine di memoria/sec

Byte totali al secondo dell'interfaccia di rete

Uso in percentuale del file di paginazione

PhysicalDisk % Tempo su disco

PhysicalDisk Media. Lunghezza coda disco

PhysicalDisk Avg. Sec disco/Lettura

PhysicalDisk Avg. Sec disco/Scrittura

PhysicalDisk Byte di lettura su disco/sec

PhysicalDisk Letture su disco/sec

PhysicalDisk Byte di scrittura su disco/sec

PhysicalDisk Scritture su disco/sec

Tempo di inattività in percentuale del processore

Tempo di interruzione in percentuale del processore

Tempo processore in percentuale del processore

Tempo utente in percentuale del processore

SQLServer:Access Methods Forwarded Records/sec

SQLServer:Access Methods Full Scans/sec

SQLServer:Access Methods Page Splits/sec

SQLServer:Buffer Manager Buffer cache hit ratio

SQLServer:Buffer Manager Page life expectancy

SQLServer:General Statistics Processes blocked

SQLServer:General Statistics User Connections

SQLServer:Latches Average Latch Wait Time (ms)

SQLServer:Locks Average Wait Time (ms)

SQLServer:Locks Lock Timeouts/sec

SQLServer:Locks Lock Waits/sec

SQLServer:Locks Number of Deadlocks/sec

SQLServer:Memory Manager Memory Grants Pending

SQLServer:SQL Statistics Batch Requests/sec

SQLServer:SQL Statistics SQL Compilations/sec

SQLServer:SQL Statistics SQL Re-Compilations/sec

Lunghezza coda processore di sistema

Connessioni TCPv4 stabilite

Connessioni TCPv6 stabilite

Svuotamenti della cache dei file W3SVC_W3WP

Mancati riscontri della cache dei file W3SVC_W3WP

Richieste W3SVC_W3WP/sec

Svuotamenti della cache URI W3SVC_W3WP

Mancati riscontri della cache URI W3SVC_W3WP

Byte del servizio Web ricevuti/sec

Byte del servizio Web inviati/sec

Tentativi di connessione/sec al servizio Web

Connessioni correnti del servizio Web

Richieste Get al secondo del servizio Web

Richieste Post al secondo del servizio Web

Byte ricevuti/sec

Lunghezza coda messaggi normali/sec

Lunghezza coda messaggi urgenti/sec

Conteggio riconnessione

Lunghezza coda messaggi non confermati/sec

Messaggi in sospeso

Messaggi inviati/sec

Messaggi di aggiornamento del database/sec

Aggiorna messaggi/sec

Svuotamenti/sec

Checkpoint crittografia salvati/sec

Checkpoint crittografia ripristinati/sec

Checkpoint del registro ripristinati/sec

Checkpoint del registro salvati/sec

Chiamate API cluster/sec

Chiamate API risorsa/sec

Gestioni cluster /sec

Gestioni risorse/sec

CloudWatch metriche dei processi degli agenti (server Windows)

Le metriche di processo vengono raccolte utilizzando il plugin CloudWatch agent procstat. Solo le istanze Amazon EC2 che eseguono carichi di lavoro Windows supportano i parametri di processo.

procstat cpu_time_system

procstat cpu_time_user

procstat cpu_usage

procstat memory_rss

procstat memory_vms

procstat read_bytes

procstat write_bytes

.procstat read_count

procstat write_count

CloudWatch metriche degli agenti (server Linux)

cpu_time_active

cpu_time_guest

cpu_time_guest_nice

cpu_time_idle

cpu_time_iowait

cpu_time_irq

cpu_time_nice

cpu_time_softirq

cpu_time_steal

cpu_time_system

cpu_time_user

cpu_usage_active

cpu_usage_guest

cpu_usage_guest_nice

cpu_usage_idle

cpu_usage_iowait

cpu_usage_irq

cpu_usage_nice

cpu_usage_softirq

cpu_usage_steal

cpu_usage_system

cpu_usage_user

disk_free

disk_inodes_free

disk_inodes_used

disk_used

disk_used_percent

diskio_io_time

diskio_iops_in_progress

diskio_read_bytes

diskio_read_time

diskio_reads

diskio_write_bytes

diskio_write_time

diskio_writes

mem_active

mem_available

mem_available_percent

mem_buffered

mem_cached

mem_free

mem_inactive

mem_used

mem_used_percent

net_bytes_recv

net_bytes_sent

net_drop_in

net_drop_out

net_err_in

net_err_out

net_packets_recv

net_packets_sent

netstat_tcp_close

netstat_tcp_close_wait

netstat_tcp_closing

netstat_tcp_established

netstat_tcp_fin_wait1

netstat_tcp_fin_wait2

netstat_tcp_last_ack

netstat_tcp_listen

netstat_tcp_none

netstat_tcp_syn_recv

netstat_tcp_syn_sent

netstat_tcp_time_wait

netstat_udp_socket

processes_blocked

processes_dead

processes_idle

processes_paging

processes_running

processes_sleeping

processes_stopped

processes_total

processes_total_threads

processes_wait

processes_zombies

swap_free

swap_used

swap_used_percent

Elastic Block Store (EBS)

CloudWatch Application Insights supporta le seguenti metriche:

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueueLength

VolumeThroughputPercentage

VolumeConsumedReadWriteOps

BurstBalance

Amazon Elastic File System (Amazon EFS)

CloudWatch Application Insights supporta le seguenti metriche:

BurstCreditBalance

PercentIOLimit

PermittedThroughput

MeteredIOBytes

Totale degli obyte

DataWriteIObyte

DataReadIObyte

Metadati Iobyte

ClientConnections

TimeSinceLastSync

StorageBytes

Prestazioni

PercentageOfPermittedThroughputUtilization

ThroughputIOPS

PercentThroughputDataReadIObyte

PercentThroughputDataWriteIObyte

PercentageOfIObyte IOPS DataRead

PercentageOfDataWriteIObyte IOPS

AverageDataReadIO BytesSize

AverageDataWriteIO BytesSize

Elastic Load Balancer (ELB)

CloudWatch Application Insights supporta le seguenti metriche:

Stimato ALB ActiveConnectionCount

EstimatedALBConsumedLCUs

Alb stimato NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

RequestCount

UnHealthyHostCount

ELB applicazione

CloudWatch Application Insights supporta le seguenti metriche:

Stimato ALB ActiveConnectionCount

EstimatedALBConsumedLCUs

Alb stimato NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latenza

RequestCount

SurgeQueueLength

UnHealthyHostCount

Gruppi Amazon EC2 Auto Scaling

CloudWatch Application Insights supporta le seguenti metriche:

CPU CreditBalance

CPU CreditUsage

CPU SurplusCreditBalance

CPU SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBS% ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Istanza

StatusCheckFailed_Sistema

Amazon Simple Queue Server (SQS)

CloudWatch Application Insights supporta le seguenti metriche:

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

Amazon Relational Database Service (RDS)

CloudWatch Application Insights supporta le seguenti metriche:

Metriche

- Istanze database RDS
- Cluster di database RDS

Istanze database RDS

BurstBalance

CPU CreditBalance

CPUUtilization

DatabaseConnections

DiskQueueDepth

SQL fallito ServerAgentJobsCount

FreeStorageSpace

FreeableMemory

NetworkReceiveThroughput

NetworkTransmitThroughput

ReadIOPS

ReadLatency

ReadThroughput

WriteIOPS

WriteLatency

WriteThroughput

Cluster di database RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

BufferCacheHitRatio

CPUUtilization

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Deadlock

DeleteLatency

DeleteThroughput

EngineUptime

FreeLocalStorage

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Query

ResultSetCacheHitRatio

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPS

VolumeWriteIOPS

AWS Lambda funzione

CloudWatch Application Insights supporta le seguenti metriche:

Errori

DeadLetterErrors

Durata

Throttles

IteratorAge

ProvisionedConcurrencySpilloverInvocations

Tabella Amazon DynamoDB

CloudWatch Application Insights supporta le seguenti metriche:

SystemErrors

UserErrors

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

ConditionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

Bucket Amazon S3

CloudWatch Application Insights supporta le seguenti metriche:

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS Step Functions

CloudWatch Application Insights supporta le seguenti metriche:

Metriche

- Execution-level
- Attività
- Funzione Lambda
- Integrazione dei servizi
- API Step Functions

Execution-level

ExecutionTime

ExecutionThrottled

ExecutionsFailed

ExecutionsTimedOut

ExecutionsAborted

ExecutionsSucceeded

ExecutionsStarted

Attività

ActivityRunTime

ActivityScheduleTime

ActivityTime

ActivitiesFailed

ActivitiesHeartbeatTimedOut

ActivitiesTimedOut

ActivitiesScheduled

ActivitiesSucceeded

ActivitiesStarted

Funzione Lambda

LambdaFunctionRunTime

LambdaFunctionScheduleTime

LambdaFunctionTime

LambdaFunctionsFailed

LambdaFunctionsTimedOut

LambdaFunctionsScheduled

LambdaFunctionsSucceeded

LambdaFunctionsStarted

Integrazione dei servizi

ServiceIntegrationRunTime

ServiceIntegrationScheduleTime

ServiceIntegrationTime

ServiceIntegrationsFailed

ServiceIntegrationsTimedOut

ServiceIntegrationsScheduled

ServiceIntegrationsSucceeded

ServiceIntegrationsStarted

API Step Functions

ThrottledEvents

ProvisionedBucketSize

ProvisionedRefillRate

ConsumedCapacity

Fasi REST API di API Gateway

CloudWatch Application Insights supporta le seguenti metriche:

4XXError

5XXError

IntegrationLatency

Latenza

CacheHitCount

CacheMissCount

SAP HANA

> **Note**
>
> CloudWatch Application Insights supporta solo ambienti SID HANA singoli. Se sono collegati più SID HANA, il monitoraggio verrà configurato solo per il primo SID rilevato.

CloudWatch Application Insights supporta le seguenti metriche:

hanadb_every_service_started_status

hanadb_daemon_service_started_status

hanadb_preprocessor_service_started_status

hanadb_webdispatcher_service_started_status

hanadb_compileserver_service_started_status

hanadb_nameserver_service_started_status

hanadb_server_startup_time_variations_seconds

hanadb_level_5_alerts_count

hanadb_level_4_alerts_count

hanadb_out_of_memory_events_count

hanadb_max_trigger_read_ratio_percent

hanadb_max_trigger_write_ratio_percent

hanadb_log_switch_wait_ratio_percent

hanadb_log_switch_race_ratio_percent

hanadb_time_since_last_savepoint_seconds

hanadb_disk_usage_highlevel_percent

hanadb_max_converter_page_number_count

hanadb_long_running_savepoints_count

hanadb_failed_io_reads_count

hanadb_failed_io_writes_count

hanadb_disk_data_unused_percent

hanadb_current_allocation_limit_used_percent

hanadb_table_allocation_limit_used_percent

hanadb_host_total_physical_memory_mb

hanadb_host_physical_memory_used_mb

hanadb_host_physical_memory_free_mb

hanadb_swap_memory_free_mb

hanadb_swap_memory_used_mb

hanadb_host_allocation_limit_mb

hanadb_host_total_memory_used_mb

hanadb_host_total_peak_memory_used_mb

hanadb_host_total_allocation_limit_mb

hanadb_host_code_size_mb

hanadb_host_shared_memory_allocation_mb

hanadb_cpu_usage_percent

hanadb_cpu_user_percent

hanadb_cpu_system_percent

hanadb_cpu_waitio_percent

hanadb_cpu_busy_percent

hanadb_cpu_idle_percent

hanadb_long_delta_merge_count

hanadb_unsuccessful_delta_merge_count

hanadb_successful_delta_merge_count

hanadb_row_store_allocated_size_mb

hanadb_row_store_free_size_mb

hanadb_row_store_used_size_mb

hanadb_temporary_tables_count

hanadb_large_non_compressed_tables_count

hanadb_total_non_compressed_tables_count

hanadb_longest_running_job_seconds

hanadb_average_commit_time_milliseconds

hanadb_suspended_sql_statements_count

hanadb_plan_cache_hit_ratio_percent

hanadb_plan_cache_lookup_count

hanadb_plan_cache_hit_count

hanadb_plan_cache_total_execution_microseconds

hanadb_plan_cache_cursor_duration_microseconds

hanadb_plan_cache_preparation_microseconds

hanadb_plan_cache_evicted_count

hanadb_plan_cache_evicted_microseconds

hanadb_plan_cache_evicted_preparation_count

hanadb_plan_cache_evicted_execution_count

hanadb_plan_cache_evicted_preparation_microseconds

hanadb_plan_cache_evicted_cursor_duration_microseconds

hanadb_plan_cache_evicted_total_execution_microseconds

hanadb_plan_cache_evicted_plan_size_mb

hanadb_plan_cache_count

hanadb_plan_cache_preparation_count

hanadb_plan_cache_execution_count

hanadb_network_collision_rate

hanadb_network_receive_rate

hanadb_network_transmit_rate

hanadb_network_packet_receive_rate

hanadb_network_packet_transmit_rate

hanadb_network_transmit_error_rate

hanadb_network_receive_error_rate

hanadb_time_until_license_expires_days

hanadb_is_license_valid_status

hanadb_local_running_connections_count

hanadb_local_idle_connections_count

hanadb_remote_running_connections_count

hanadb_remote_idle_connections_count

hanadb_last_full_data_backup_age_days

hanadb_last_data_backup_age_days

hanadb_last_log_backup_age_hours

hanadb_failed_data_backup_past_7_days_count

hanadb_failed_log_backup_past_7_days_count

hanadb_oldest_backup_in_catalog_age_days

hanadb_backup_catalog_size_mb

hanadb_hsr_replication_status

hanadb_hsr_log_shipping_delay_seconds

hanadb_hsr_secondary_failover_count

hanadb_hsr_secondary_reconnect_count

hanadb_hsr_async_buffer_used_mb

hanadb_hsr_secondary_active_status

hanadb_handle_count

hanadb_ping_time_milliseconds

hanadb_connection_count

hanadb_internal_connection_count

hanadb_external_connection_count

hanadb_idle_connection_count

hanadb_transaction_count

hanadb_internal_transaction_count

hanadb_external_transaction_count

hanadb_user_transaction_count

hanadb_blocked_transaction_count

hanadb_statement_count

hanadb_active_commit_id_range_count

hanadb_mvcc_version_count

hanadb_pending_session_count

hanadb_record_lock_count

hanadb_read_count

hanadb_write_count

hanadb_merge_count

hanadb_unload_count

hanadb_active_thread_count

hanadb_waiting_thread_count

hanadb_total_thread_count

hanadb_active_sql_executor_count

hanadb_waiting_sql_executor_count

hanadb_total_sql_executor_count

hanadb_data_write_size_mb

hanadb_data_write_time_milliseconds

hanadb_log_write_size_mb

hanadb_log_write_time_milliseconds

hanadb_data_read_size_mb

hanadb_data_read_time_milliseconds

hanadb_log_read_size_mb

hanadb_log_read_time_milliseconds

hanadb_data_backup_write_size_mb

hanadb_data_backup_write_time_milliseconds

hanadb_log_backup_write_size_mb

hanadb_log_backup_write_time_milliseconds

hanadb_mutex_collision_count

hanadb_read_write_lock_collision_count

hanadb_admission_control_admit_count

hanadb_admission_control_reject_count

hanadb_admission_control_queue_size_mb

hanadb_admission_control_wait_time_milliseconds

SAP ASE

CloudWatch Application Insights supporta le seguenti metriche:

asedb_database_availability

asedb_trunc_log_on_chkpt_enabled

asedb_last_db_backup_age_in_days

asedb_last_transaction_log_backup_age_in_hours

asedb_suspected_database

asedb_db_space_usage_percent

asedb_db_log_space_usage_percent

asedb_locked_login

asedb_has_mixed_log_and_data

asedb_runtime_for_open_transactions

asedb_data_cache_hit_ratio

asedb_data_cache_usage

asedb_sql_cache_hit_ratio

asedb_cache_usage

asedb_run_queue_length

asedb_number_of_rollbacks

asedb_number_of_commits

asedb_number_of_transactions

asedb_outstanding_disk_io

asedb_percent_io_busy

asedb_percent_system_busy

asedb_percent_locks_active

asedb_scheduled_jobs_failed_percent

asedb_user_connections_percent

asedb_query_logical_reads

asedb_query_physical_reads

asedb_query_cpu_time

asedb_query_memory_usage

SAP ASE High Availability su Amazon EC2

CloudWatch Application Insights supporta le seguenti metriche:

asedb_ha_replication_state

asedb_ha_replication_mode

asedb_ha_replication_latency_in_minutes

SAP NetWeaver

CloudWatch Application Insights supporta le seguenti metriche:

Parametro	Descrizione
sap_alerts_ ResponseTime	L'avviso sul tempo di risposta SAP di CCMS (RZ20) >R3Services>Dialog>. ResponseTime
sap_alerts_ ResponseTimeDialog	L'avviso di dialogo sul tempo di risposta SAP di CCMS (RZ20) >R3Services>Dialog>. ResponseTimeDialog
ResponseTimeDialogsap_alerts_ RFC	L'avviso sul tempo di risposta SAP di CCMS (RZ20) >R3Services> Dialog> RFC. ResponseTimeDialog
SAP_Alerts_db RequestTime	L'avviso sul tempo di risposta SAP di CCMS (RZ20) >R3Services>Dialog>DB. RequestTime
sap_alerts_ FrontendResponseTime	L'avviso sul tempo di risposta SAP di CCMS (RZ20) >R3Services > Dialog>. FrontEndR esponseTime
sap_alerts_Database	Il sistema SAP ha registrato errori relativi al database. Avviso da SM21 o CCMS (RZ20)>R3 Syslog>Database.
sap_alerts_ QueueTime	L'avviso SAP Queue Time di CCMS (RZ20) >R3Services>Dialog>. QueueTime
sap_alerts_ AbortedJobs	Processi in background non riusciti nel sistema SAP. Avviso da (RZ20) >R3Services > Background>. AbortedJobs

Parametro	Descrizione
sap_alerts_ BasisSystem	Errori a livello di sistema registrati dal sistema SAP. Avviso da SM21 o CCMS (RZ20) >R3Syslog>. BasisSystem
sap_alerts_Security	Messaggi relativi alla sicurezza registrati dal sistema SAP. Avviso da SM21 o CCMS (RZ20)>R3Syslog>Security.
sap_alerts_System	Il sistema SAP ha registrato messaggi relativi alla sicurezza o all'audit. Avviso da SM21 o CCMS (RZ20)>Security>System.
sap_alerts_ LongRunners	Nel sistema SAP sono presenti programmi a esecuzione prolungata. Avviso da CCMS (RZ20) >R3Services > Dialog>. LongRunners
sap_alerts_ SqlError	Sono presenti log degli errori a livello client del database SAP. Avviso da CCMS (RZ20) > > >. DatabaseClient AbapSql SqlError
sap_alerts_State	Avviso di stato da CCMS (RZ20)>OS Collector >State.
sap_alerts_Shortdumps	Avviso di shortdumps da ST22 e CCMS (RZ20)>R3Abap>Shortdumps.
sap_alerts_Availability	Avviso di disponibilità per l'istanza dell'appl ication server SAP da SM21, SM50, SM51, SM66 e CCMS (RZ20) > >Disponibilità. InstanceAsTask
sap_dispatcher_queue_high	La funzione `GetQueueStatistic` del servizio Web SAPControl mostra il conteggio elevato della coda del dispatcher.

Parametro	Descrizione
sap_dispatcher_queue_max	La funzione `GetQueueStatistic` del servizio Web SAPControl mostra il conteggio massimo della coda del dispatcher.
sap_dispatcher_queue_now	La funzione `GetQueueStatistic` del servizio Web SAPControl mostra il conteggio attuale della coda del dispatcher.
sap_dispatcher_queue_reads	La funzione `GetQueueStatistic` del servizio Web SAPControl mostra il conteggio delle letture della coda del dispatcher.
sap_dispatcher_queue_writes	La funzione `GetQueueStatistic` del servizio Web SAPControl mostra il conteggio delle scritture della coda del dispatcher.
sap_enqueue_server_arguments_high	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce gli argomenti di accodamento elevati.
sap_enqueue_server_arguments_max	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce gli argomenti di accodamento massimi.
sap_enqueue_server_arguments_now	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce gli argomenti di accodamento attuali.
sap_enqueue_server_arguments_state	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce lo stato degli argomenti di accodamento.
sap_enqueue_server_backup_requests	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce le richieste di backup di accodamento.

Parametro	Descrizione
sap_enqueue_server_cleanup_requests	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce le richieste di pulizia di accodamento.
sap_enqueue_server_dequeue_all_requests	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce la rimozione di tutte le richieste dalla coda.
sap_enqueue_server_dequeue_errors	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce la rimozione di tutte le richieste di accodamento.
sap_enqueue_server_dequeue_requests	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce le richieste di rimozione accodamento.
sap_enqueue_server_enqueue_errors	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce gli errori di accodamento.
sap_enqueue_server_enqueue_rejects	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce i rifiuti di accodamento.
sap_enqueue_server_enqueue_requests	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce le richieste di accodamento.
sap_enqueue_server_lock_time	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce il tempo di blocco di accodamento.
sap_enqueue_server_lock_wait_time	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce il tempo di attesa del blocco dell'accodamento.

Parametro	Descrizione
sap_enqueue_server_locks_high	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce un numero elevato di blocchi di accodamento.
sap_enqueue_server_locks_max	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce il numero massimo di blocchi di accodamento.
sap_enqueue_server_locks_now	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce i blocchi di accodamento attuali.
sap_enqueue_server_locks_state	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce lo stato dei blocchi di accodamento.
sap_enqueue_server_owner_high	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce il numero più elevato di proprietari di accodamento.
sap_enqueue_server_owner_max	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce il numero massimo di proprietari di accodamento.
sap_enqueue_server_owner_now	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce il proprietario attuale di accodamento.
sap_enqueue_server_owner_state	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce lo stato del proprieta rio di accodamento.
sap_enqueue_server_replication_state	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce lo stato di replica di accodamento.

Parametro	Descrizione
sap_enqueue_server_reporting_requests	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce lo stato delle richieste di segnalazione.
sap_enqueue_server_server_time	La funzione `EnqGetStatistic` del servizio Web SAPControl fornisce l'ora del server di accodamento.
sap_HA_check_failover_config_state	La funzione `HACheckFailoverConfig` del servizio Web SAPControl fornisce lo stato SAP a disponibilità elevata.
sap_HA_get_failover_config_HAActive	La funzione `HAGetFailoverConfig` del servizio Web SAPControl fornisce la configura zione e lo stato del cluster SAP a disponibilità elevata.
sap_start_service_processes	La funzione `GetProcessList` del servizio Web SAPControl fornisce lo stato dei processi disp+work, IGS, gwrd, icman, message server ed enqueue server.

Cluster HA

CloudWatch Application Insights supporta le seguenti metriche:

ha_cluster_pacemaker_stonith_enabled

ha_cluster_corosync_quorate

hanadb_webdispatcher_service_started_status

ha_cluster_pacemaker_nodes

ha_cluster_corosync_ring_errors

ha_cluster_pacemaker_fail_count

Java

CloudWatch Application Insights supporta le seguenti metriche:

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_threading_daemonthreadcount

java_lang_classloading_loadedclasscount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Amazon Elastic Container Service (Amazon ECS)

CloudWatch Application Insights supporta le seguenti metriche:

Metriche

- CloudWatch metriche integrate
- Parametri di Container Insights
- Parametri di Container Insights Prometheus

CloudWatch metriche integrate

CPUReservation

CPUUtilization

MemoryReservation

MemoryUtilization

GPUReservation

Parametri di Container Insights

ContainerInstanceCount

CpuUtilized

CpuReserved

DeploymentCount

DesiredTaskCount

MemoryUtilized

MemoryReserved

NetworkRxBytes

NetworkTxBytes

PendingTaskCount

RunningTaskCount

ServiceCount

StorageReadBytes

StorageWriteBytes

TaskCount

TaskSetCount

instance_cpu_limit

instance_cpu_reserved_capacity

instance_cpu_usage_total

instance_cpu_utilization

instance_filesystem_utilization

instance_memory_limit

instance_memory_reserved_capacity

instance_memory_use_

instance_memory_working_set

instance_network_total_bytes

instance_number_of_running_tasks

Parametri di Container Insights Prometheus

Parametri Java JMX

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Kubernetes attivo AWS

CloudWatch Application Insights supporta le seguenti metriche:

Metriche

- Parametri di Container Insights
- Parametri di Container Insights Prometheus

Parametri di Container Insights

cluster_failed_node_count

cluster_node_count

namespace_number_of_running_pods

node_cpu_limit

node_cpu_reserved_capacity

node_cpu_usage_total

node_cpu_utilization

node_filesystem_utilization

node_memory_limit

node_memory_reserved_capacity

node_memory_use

node_memory_working_set

node_network_total_bytes

node_number_of_running_containers

node_number_of_running_pod

pod_cpu_reserved_capacity

pod_cpu_usage

pod_cpu_utilization_over_pod_limit

pod_memory_reserved_capacity

pod_memory_use

pod_memory_utilization_over_pod_limit

pod_network_rx_bytes

pod_network_tx_bytes

service_number_of_running_pods

Parametri di Container Insights Prometheus

Parametri Java JMX

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Amazon FSx

CloudWatch Application Insights supporta le seguenti metriche:

DataReadBytes

DataWriteBytes

DataReadOperations

DataWriteOperations

MetadataOperations

FreeStorageCapacity

FreeDataStorageCapacity

LogicalDiskUsage

PhysicalDiskUsage

Amazon VPC

CloudWatch Application Insights supporta le seguenti metriche:

NetworkAddressUsage

NetworkAddressUsagePeered

VPC FirewallQueryVolume

Gateway NAT di Amazon VPC

CloudWatch Application Insights supporta le seguenti metriche:

ErrorPortAllocation

IdleTimeoutCount

Controllo dell'integrità di Amazon Route 53

CloudWatch Application Insights supporta le seguenti metriche:

ChildHealthCheckHealthyCount

ConnectionTime

HealthCheckPercentageHealthy

HealthCheckStatus

SSL HandshakeTime

TimeToFirstByte

Zona ospitata di Amazon Route 53

CloudWatch Application Insights supporta le seguenti metriche:

DNSQueries

DNSSEC InternalFailure

DNSSEC KeySigningKeysNeedingAction

DNSSEC KeySigningKeyMaxNeedingActionAge

DNSSEC KeySigningKeyAge

Amazon Route 53 Resolver endpoint

CloudWatch Application Insights supporta le seguenti metriche:

EndpointHealthyEniCount

EndpointUnHealthyConteggio Eni

InboundQueryVolume

OutboundQueryVolume

OutboundQueryAggregateVolume

AWS Network Firewall gruppo di regole

CloudWatch Application Insights supporta le seguenti metriche:

FirewallRuleGroupQueryVolume

AWS Network Firewall associazione di gruppi di regole

CloudWatch Application Insights supporta le seguenti metriche:

FirewallRuleGroupVpcQueryVolume

Parametri con requisiti di punti dati

Per i parametri senza una soglia predefinita evidente di attivazione dell'allarme, Application Insights attende finché il parametro non dispone di un numero sufficiente di punti dati per prevedere una

soglia ragionevole di attivazione dell'allarme. I requisiti relativi ai punti di dati metrici che CloudWatch Application Insights verifica prima della creazione di un allarme sono:

- Il parametro dispone di almeno 100 punti dati dagli ultimi 15 giorni fino agli ultimi 2 giorni.
- Il parametro dispone di almeno 100 punti dati dall'ultimo giorno.

I parametri sottostanti seguono questi requisiti di punti dati. Tieni presente che le metriche degli CloudWatch agenti richiedono fino a un'ora per creare allarmi.

Metriche

- AWS/ApplicationELB
- AWS/ AutoScaling
- <shared id="xxx"/>/EC2
- Elastic Block Store (EBS)
- AWS/ELB
- AWS/RDS
- AWS/Lambda
- AWS/SQS
- AWS/CWAgent
- AWS/DynamoDB
- AWS/S3
- AWS/States
- AWS/ ApiGateway
- AWS/SNS

AWS/ApplicationELB

ActiveConnectionCount

ConsumedLCUs

HTTPCode_ELB_4XX_Count

HTTPCode_Target_2XX_Count

HTTPCode_Target_3XX_Count

HTTPCode_Target_4XX_Count

HTTPCode_Target_5XX_Count

NewConnectionCount

ProcessedBytes

TargetResponseTime

UnHealthyHostCount

AWS/ AutoScaling

GroupDesiredCapacity

GroupInServiceInstances

GroupMaxSize

GroupMinSize

GroupPendingInstances

GroupStandbyInstances

GroupTerminatingInstances

GroupTotalInstances

<shared id="xxx"/>/EC2

CPU CreditBalance

CPU CreditUsage

CPU SurplusCreditBalance

CPU SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

EBS% ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

Elastic Block Store (EBS)

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueueLength

VolumeThroughputPercentage

VolumeConsumedReadWriteOps

BurstBalance

AWS/ELB

ALB stimato ActiveConnectionCount

EstimatedALBConsumedLCUs

Alb stimato NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latenza

RequestCount

SurgeQueueLength

UnHealthyHostCount

AWS/RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

CPU CreditBalance

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Deadlock

DeleteLatency

DeleteThroughput

DiskQueueDepth

EngineUptime

FreeLocalStorage

FreeStorageSpace

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Query

ReadIOPS

ReadThroughput

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPS

VolumeWriteIOPS

WriteIOPS

WriteThroughput

AWS/Lambda

Errori

DeadLetterErrors

Durata

Throttles

IteratorAge

ProvisionedConcurrencySpilloverInvocations

AWS/SQS

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

AWS/CWAgent

LogicalDisk % di spazio libero

Memory: byte impegnati della percentuale di memoria in uso

Mbyte di memoria disponibili

Byte totali al secondo dell'interfaccia di rete

Uso in percentuale del file di paginazione

PhysicalDisk % Tempo su disco

PhysicalDisk Media. Sec disco/Lettura

PhysicalDisk Avg. Sec disco/Scrittura

PhysicalDisk Byte di lettura su disco/sec

PhysicalDisk Letture su disco/sec

PhysicalDisk Byte di scrittura su disco/sec

PhysicalDisk Scritture su disco/sec

Tempo di inattività in percentuale del processore

Tempo di interruzione in percentuale del processore

Tempo processore in percentuale del processore

Tempo utente in percentuale del processore

SQLServer:Access Methods Forwarded Records/sec

SQLServer:Access Methods Page Splits/sec

SQLServer:Buffer Manager Buffer cache hit ratio

SQLServer:Buffer Manager Page life expectancy

SQLServer:Database Replica File Bytes Received/sec

SQLServer:Database Replica Log Bytes Received/sec

SQLServer:Database Replica Log remaining for undo

SQLServer:Database Replica Log Send Queue

SQLServer:Database Replica Mirrored Write Transaction/sec

SQLServer:Database Replica Recovery Queue

SQLServer:Database Replica Redo Bytes Remaining

SQLServer:Database Replica Redone Bytes/sec

SQLServer:Database Replica Total Log requiring undo

SQLServer:Database Replica Transaction Delay

SQLServer:General Statistics Processes blocked

SQLServer:SQL Statistics Batch Requests/sec

SQLServer:SQL Statistics SQL Compilations/sec

SQLServer:SQL Statistics SQL Re-Compilations/sec

Lunghezza coda processore di sistema

Connessioni TCPv4 stabilite

Connessioni TCPv6 stabilite

AWS/DynamoDB

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

ConditionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

AWS/S3

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS/States

ActivitiesScheduled

ActivitiesStarted

ActivitiesSucceeded

ActivityScheduleTime

ActivityRuntime

ActivityTime

LambdaFunctionsScheduled

LambdaFunctionsStarted

LambdaFunctionsSucceeded

LambdaFunctionScheduleTime

LambdaFunctionRuntime

LambdaFunctionTime

ServiceIntegrationsScheduled

ServiceIntegrationsStarted

ServiceIntegrationsSucceeded

ServiceIntegrationScheduleTime

ServiceIntegrationRuntime

ServiceIntegrationTime

ProvisionedRefillRate

ProvisionedBucketSize

ConsumedCapacity

ThrottledEvents

AWS/ ApiGateway

4XXError

IntegrationLatency

Latenza

DataProcessed

CacheHitCount

CacheMissCount

AWS/SNS

NumberOfNotificationsDelivered

NumberOfMessagesPublished

NumberOfNotificationsFailed

NumberOfNotificationsFilteredOut

NumberOfNotificationsFilteredOut-InvalidAttributes

NumberOfNotificationsFilteredOut-NoMessageAttributes

NumberOfNotificationsRedrivenToDlq

NumberOfNotificationsFailedToRedriveToDlq

SMS SuccessRate

Parametri consigliati

Nella tabella riportata di seguito sono elencati i parametri consigliati per ogni tipo di componente.

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Istanza EC2 (server Windows)	Predefinita/personalizzata	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso LogicalDisk % di spazio libero Mbyte di memoria disponibili
	Active Directory	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso Mbyte di memoria disponibili Database ==> Cache del database delle istanze % occorrenza DirectoryServices Operazioni di replica DRA in sospeso DirectoryServices Sincronizzazioni di replica DRA in sospeso Errore di query ricorsivo DNS/sec

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
		LogicalDisk Avg. Lunghezza coda disco
	Applicazione Java	CPUUtilization
		StatusCheckFailed
		Tempo processore in percentuale del processore
		Memory: byte impegnati della percentuale di memoria in uso
		Mbyte di memoria disponibili
		java_lang_threading_threadcount
		java_lang_classloading_loadedclasscount
		java_lang_memory_heapmemoryusage_used
		java_lang_memory_heapmemoryusage_committed
		java_lang_operatingsystem_freephysicalmemorysize
		java_lang_operatingsystem_freeswapspacesize

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Microsoft II/.NET Web front-end	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso Mbyte di memoria disponibili Numero di eccezioni .NET CLR Exceps Thrown/sec Numero di byte totali impegnati memoria CLR .NET % di tempo in memoria .NET CLR in GC Richieste delle applicazioni ASP.NET nella coda delle applicazioni Richieste ASP.NET in coda Riavvii applicazione ASP.NET

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Livello database Microsoft SQL Server	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso Mbyte di memoria disponibili Uso in percentuale del file di paginazione Lunghezza coda processore di sistema Byte totali al secondo dell'interfaccia di rete PhysicalDisk % Tempo su disco SQLServer:Buffer Manager Buffer Cache Hit ratio SQLServer:Buffer Manager Page Life Expectancy SQLServer:General Statistics Processes Blocked SQLServer:General Statistics User Connections SQLServer:Locks Number of Deadlocks/Sec

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
		SQLServer:SQL Statistics Batch Requests/Sec
	MySQL	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso LogicalDisk % di spazio libero Mbyte di memoria disponibili
	.NET workerpool/Mid-Tier	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso Mbyte di memoria disponibili Numero di eccezioni .NET CLR Exceps Thrown/sec Numero di byte totali impegnati memoria CLR .NET % di tempo in memoria .NET CLR in GC

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Livello .NET Core	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso Mbyte di memoria disponibili
	Oracle	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso LogicalDisk % di spazio libero Mbyte di memoria disponibili
	Postgres	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso LogicalDisk % di spazio libero Mbyte di memoria disponibili

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	SharePoint	CPUUtilization StatusCheckFailed Tempo processore in percentuale del processore Memory: byte impegnati della percentuale di memoria in uso Mbyte di memoria disponibili Riduce gli API della cache delle applicazioni ASP.NET Richieste rifiutate ASP.NET Riavvii processo di lavoro ASP.NET Pagine di memoria/sec SharePoint Publishing Cache La cache di pubblicazione viene scaricata al secondo SharePoint Data/ora di esecuzione della richiesta Foundation SharePoint Cache basata su disco Numero totale di compattazioni della cache SharePoint Cache basata su disco, rapporto di successo della cache Blob

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
		SharePoint Cache basata su disco, rapporto di riempimento della cache Blob SharePoint La cache Blob basata su disco si scarica al secondo Richieste ASP.NET in coda Richieste delle applicazioni ASP.NET nella coda delle applicazioni Riavvii applicazione ASP.NET LogicalDisk Avg. Sec disco/Scrittura LogicalDisk Avg. Sec disco/Lettura Tempo di interruzione in percentuale del processore
Istanze EC2 (server Linux)	Predefinita/personalizzata	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Applicazione Java	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize
	Livello database .NET Core Tier o SQL Server	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Oracle	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent
	Postgres	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Gruppo di istanze EC2	Multi-nodo SAP HANA o nodo singolo	• hanadb_server_startup_time_variations_seconds • hanadb_level_5_alerts_count • hanadb_level_4_alerts_count • hanadb_out_of_memory_events_count • hanadb_max_trigger_read_ratio_percent • hanadb_max_trigger_write_ratio_percent • hanadb_log_switch_race_ratio_percent • hanadb_time_since_last_savepoint_seconds • hanadb_disk_usage_highlevel_percent • hanadb_current_allocation_limit_used_percent • hanadb_table_allocation_limit_used_percent • hanadb_cpu_usage_percent • hanadb_plan_cache_hit_ratio_percent • hanadb_last_data_backup_age_days

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Volume EBS	Qualsiasi	VolumeReadBytes VolumeWriteBytes VolumeReadOps VolumeWriteOps VolumeQueueLength VolumeThroughputPercentage VolumeConsumedReadWriteOps BurstBalance
Classic ELB	Qualsiasi	HTTPCode_Backend_4XX HTTPCode_Backend_5XX Latenza SurgeQueueLength UnHealthyHostCount
ELB applicazione	Qualsiasi	HTTPCode_Target_4XX_Count HTTPCode_Target_5XX_Count TargetResponseTime UnHealthyHostCount

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Istanza di database RDS	Qualsiasi	CPUUtilization ReadLatency WriteLatency BurstBalance SQL non riuscito ServerAgentJobsCount
Database e cluster RDS Database	Qualsiasi	CPUUtilization CommitLatency DatabaseConnections Deadlock FreeableMemory NetworkThroughput VolumeBytesUsed
Funzione Lambda	Qualsiasi	Durata Errori IteratorAge ProvisionedConcurrencySpilloverInvocations Throttles

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Coda SQS	Qualsiasi	ApproximateAgeOfOldestMessage ApproximateNumberOfMessagesVisible NumberOfMessagesSent
Tabella Amazon DynamoDB	Qualsiasi	SystemErrors UserErrors ConsumedReadCapacityUnits ConsumedWriteCapacityUnits ReadThrottleEvents WriteThrottleEvents ConditionalCheckFailedRequests TransactionConflict

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Bucket Amazon S3	Qualsiasi	Se la configurazione di replica con Replication Time Control (RTC) è abilitata: ReplicationLatency BytesPendingReplication OperationsPendingReplication Se i parametri di richiesta sono attivati: 5xxErrors 4xxErrors BytesDownloaded BytesUploaded

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
AWS Step Functions	Qualsiasi	Generali • ExecutionThrottled • ExecutionsAborted • ProvisionedBucketSize • ProvisionedRefillRate • ConsumedCapacity Se il tipo di macchina a stati è **`EXPRESS`** o il livello del gruppo di log è **`OFF`** • ExecutionsFailed • ExecutionsTimedOut Se la macchina a stati ha funzioni Lambda • LambdaFunctionsFailed • LambdaFunctionsTimedOut Se la macchina a stati ha attività • ActivitiesFailed • ActivitiesTimedOut • ActivitiesHeartbea tTimedOut Se la macchina a stati dispone di integrazioni di servizio • ServiceIntegrationsFailed

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
		• ServiceIntegration sTimedOut
Fase REST API di API Gateway	Qualsiasi	• 4XXErrors • 5XXErrors • Latenza

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Cluster ECS	Qualsiasi	CpuUtilized
		MemoryUtilized
		NetworkRxBytes
		NetworkTxBytes
		RunningTaskCount
		PendingTaskCount
		StorageReadBytes
		StorageWriteBytes
		CPUReservation (solo tipo di avvio EC2)
		CPUutilization (solo tipo di avvio EC2)
		MemoryReservation (Solo tipo di avvio EC2)
		MemoryUtilization (Solo tipo di lancio EC2)
		GPUReservation (solo tipo di avvio EC2)
		instance_cpu_utilization (solo tipo di avvio EC2)
		instance_filesystem_utilization (solo tipo di avvio EC2)
		instance_memory_utilization (solo tipo di avvio EC2)

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
		instance_network_total_bytes (solo tipo di avvio EC2)

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Applicazione Java	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (solo tipo di avvio EC2) CPUutilization (solo tipo di avvio EC2) MemoryReservation (Solo tipo di lancio EC2) MemoryUtilization (Solo tipo di lancio EC2) GPUReservation (solo tipo di avvio EC2) instance_cpu_utilization (solo tipo di avvio EC2) instance_filesystem_utilization (solo tipo di avvio EC2) instance_memory_utilization (solo tipo di avvio EC2)

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
		instance_network_total_bytes (solo tipo di avvio EC2) java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize
Servizio ECS	Qualsiasi	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Applicazione Java	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Cluster EKS	Qualsiasi	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_use
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_usage
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_use
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Applicazione Java	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserved_capacity node_memory_use node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_usage pod_cpu_utilization_over_pod_limit pod_memory_reserved_capacity pod_memory_use pod_memory_utilization_over_pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
Cluster Kubernetes su EC2	Qualsiasi	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_use
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_usage
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_use
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
	Applicazione Java	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserved_capacity node_memory_use node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_usage pod_cpu_utilization_over_pod_limit pod_memory_reserved_capacity pod_memory_use pod_memory_utilization_over_pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount

Tipo di componente	Tipo di carico di lavoro	Parametro consigliato
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

La tabella seguente elenca i processi e le metriche di processo consigliati per ogni tipo di componente. CloudWatch Application Insights non consiglia il monitoraggio dei processi per i processi che non vengono eseguiti su un'istanza.

Tipo di componente	Tipo di carico di lavoro	Processo consigliato	Parametro consigliato

Istanza EC2 (server Windows)	Microsoft II/.NET Web front-end	`w3wp`	`procstat cpu_usage ,`
			
			`procstat memory_rss ,`
			
			`procstat memory_vms ,`
			
			`procstat read_bytes ,`
			
			`procstat write_bytes`

	Livello database Microsoft SQL Server	`SQLAgent`	`procstat cpu_usage ,`

Tipo di componente	Tipo di carico di lavoro	Processo consigliato	Parametro consigliato

			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

		`sqlservr`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

		`sqlwriter`	`procstat cpu_usage ,` `procstat memory_rss`

Tipo di componente	Tipo di carico di lavoro	Processo consigliato	Parametro consigliato
		`Reporting ServicesS ervice`	`procstat cpu_usage ,` `procstat memory_rss`

		`MsDtsServr`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

		`Msmdsrv`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

Tipo di componente	Tipo di carico di lavoro	Processo consigliato	Parametro consigliato

	.NET workerpool/Mid-Tier	`w3wp`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

	Livello .NET Core	`w3wp`	`procstat cpu_usage ,`
			`procstat memory_rss ,`
			`procstat memory_vms ,`
			`procstat read_bytes ,`
			`procstat write_bytes`

Parametri del contatore di prestazioni

I parametri del contatore delle prestazioni sono consigliati per le istanze solo quando i set di contatori di prestazioni corrispondenti sono installati nelle istanze di Windows.

Nome del parametro del contatore di prestazioni	Nome del set di contatori di prestazioni
Numero di eccezioni .NET CLR generate	Eccezioni .NET CLR
Numero di eccezioni .NET CLR Exceps Thrown/sec	Eccezioni .NET CLR
Numero di eccezioni.NET CLR # di filtri/Sec	Eccezioni .NET CLR
Numero di eccezioni .NET CLR Finallys/sec	Eccezioni .NET CLR
Eccezioni .NET CLR Throw to Catch Depth/sec	Eccezioni .NET CLR
Numero di Interop di .NET CLR di CCW	Interop .NET CLR
Numero di Interop .NET CLR di stub	Interop .NET CLR
Numero di Interop .NET CLR delle esportazioni TLB/sec	Interop .NET CLR
Numero di Interop .NET CLR delle importazioni TLB/sec	Interop .NET CLR
Numero di Interop .NET CLR di marshalling	Interop .NET CLR
Percentuale di tempo in Jit .NET CLR in Jit	.NET CLR Jit
Errori Jit standard .NET CLR Jit	.NET CLR Jit
Percentuale tempo di caricamento .NET CLR	Caricamento .NET CLR
Velocità di caricamento .NET CLR di errori di caricamento	Caricamento .NET CLR
Frequenza di LocksAndThreads contenzione.NET CLR/sec	.NET CLR LocksAndThreads
Lunghezza della coda CLR.NET CLR/sec LocksAndThreads	.NET CLR LocksAndThreads

Nome del parametro del contatore di prestazioni	Nome del set di contatori di prestazioni
Numero di byte totali impegnati memoria CLR .NET	Memoria .NET CLR
% di tempo in memoria .NET CLR in GC	Memoria .NET CLR
.NET CLR Networking 4.0.0.0 Tempo HttpWebRequest di coda medio	.NET CLR Networking 4.0.0.0
.NET CLR Networking 4.0.0.0 interrotto/sec HttpWebRequests	.NET CLR Networking 4.0.0.0
.NET CLR Networking 4.0.0.0 non HttpWebRequests riuscita/sec	.NET CLR Networking 4.0.0.0
.NET CLR Networking 4.0.0.0 in HttpWebRequests coda/sec	.NET CLR Networking 4.0.0.0
Errori di ping del processo di lavoro totali APP_POOL_WAS Total	APP_POOL_WAS
Riavvii applicazione ASP.NET	ASP.NET
Richieste rifiutate ASP.NET	ASP.NET
Riavvii processo di lavoro ASP.NET	ASP.NET
Riduce gli API della cache delle applicazioni ASP.NET	Applicazioni ASP.NET
Tempo del processore gestito in percentuale delle applicazioni (stimato) ASP.NET	Applicazioni ASP.NET
Errori totali/sec delle applicazioni ASP.NET	Applicazioni ASP.NET
Errori delle applicazioni ASP.NET non gestiti durante l'esecuzione/sec	Applicazioni ASP.NET

Nome del parametro del contatore di prestazioni	Nome del set di contatori di prestazioni
Richieste delle applicazioni ASP.NET nella coda delle applicazioni	Applicazioni ASP.NET
Richieste delle applicazioni ASP.NET/sec	Applicazioni ASP.NET
Tempo di attesa delle richieste ASP.NET	ASP.NET
Richieste ASP.NET in coda	ASP.NET
Database ==> Cache del database delle istanze % occorrenza	Database ==> Istanze
Database ==> Il database I/O delle istanze legge la latenza media	Database ==> Istanze
Database ==> Letture database I/O istanze/sec	Database ==> Istanze
Database ==> Il log I/O delle istanze scrive la latenza media	Database ==> Istanze
DirectoryServices Operazioni di replica DRA in sospeso	DirectoryServices
DirectoryServices Sincronizzazioni di replica DRA in sospeso	DirectoryServices
DirectoryServices Tempo di associazione LDAP	DirectoryServices
Query ricorsive DNS/sec	DNS
Errore di query ricorsivo DNS/sec	DNS
Query TCP DNS ricevute/sec	DNS
Query totale DNS ricevute/sec	DNS
Risposta totale DNS inviata/sec	DNS

Nome del parametro del contatore di prestazioni	Nome del set di contatori di prestazioni
Query UDP DNS ricevute/sec	DNS
Code di richieste di servizio HTTP CurrentQueueSize	Code di richiesta di servizio HTTP
LogicalDisk % di spazio libero	LogicalDisk
LogicalDisk Media. Sec disco/Scrittura	LogicalDisk
LogicalDisk Avg. Sec disco/Lettura	LogicalDisk
LogicalDisk Avg. Lunghezza coda disco	LogicalDisk
Memory: byte impegnati della percentuale di memoria in uso	Memoria
Mbyte di memoria disponibili	Memoria
Pagine di memoria/sec	Memoria
Durata media della cache in standby a lungo termine della memoria	Memoria
Byte totali al secondo dell'interfaccia di rete	Interfaccia di rete
Byte dell'interfaccia di rete ricevuti/sec	Interfaccia di rete
Byte dell'interfaccia di rete inviati/sec	Interfaccia di rete
Larghezza di banda corrente dell'interfaccia di rete	Interfaccia di rete
Uso in percentuale del file di paginazione	File di paginazione
PhysicalDisk % Tempo su disco	PhysicalDisk
PhysicalDisk Media. Lunghezza coda disco	PhysicalDisk

Nome del parametro del contatore di prestazioni	Nome del set di contatori di prestazioni
PhysicalDisk Avg. Lettura disco/sec	PhysicalDisk
PhysicalDisk Avg. Scrittura disco/sec	PhysicalDisk
PhysicalDisk Byte/sec di lettura su disco	PhysicalDisk
PhysicalDisk Letture su disco/sec	PhysicalDisk
PhysicalDisk Byte di scrittura su disco/sec	PhysicalDisk
PhysicalDisk Scritture su disco/sec	PhysicalDisk
Tempo di inattività in percentuale del processore	Processore
Tempo di interruzione in percentuale del processore	Processore
Tempo processore in percentuale del processore	Processore
Tempo utente in percentuale del processore	Processore
SharePoint Rapporto di riempimento della cache Blob Cache basata su disco	SharePoint Cache basata su disco
SharePoint La cache Blob basata su disco viene scaricata al secondo	SharePoint Cache basata su disco
SharePoint Cache basata su disco, rapporto di successo della cache Blob	SharePoint Cache basata su disco
SharePoint Cache basata su disco Numero totale di compattazioni della cache	SharePoint Cache basata su disco
SharePoint Richiesta di data/pagina di esecuzione Foundation	SharePoint Fondazione

Nome del parametro del contatore di prestazioni	Nome del set di contatori di prestazioni
SharePoint Publishing Cache La cache di pubblicazione svuota /secondo	SharePoint Cache di pubblicazione
Autenticazioni Kerberos delle statistiche di sicurezza a livello di sistema	Statistiche di sicurezza a livello di sistema
Autenticazioni NTLM delle statistiche di sicurezza a livello di sistema	Statistiche di sicurezza a livello di sistema
SQLServer:Access Methods Forwarded Records/sec	SQLServer: metodi di accesso
SQLServer:Access Methods Full Scans/Sec	SQLServer: metodi di accesso
SQLServer:Access Methods Page Splits/Sec	SQLServer: metodi di accesso
SQLServer:Buffer Manager Buffer cache hit Ratio	SQLServer: gestore buffer
SQLServer:Buffer Manager Page life Expectancy	SQLServer: gestore buffer
SQLServer:Database Replica File Bytes Received/sec	SQLServer:Database Replica
SQLServer:Database Replica Log Bytes Received/sec	SQLServer:Database Replica
SQLServer:Database Replica Log remaining for undo	SQLServer:Database Replica
SQLServer:Database Replica Log Send Queue	SQLServer:Database Replica
SQLServer:Database Replica Mirrored Write Transaction/sec	SQLServer:Database Replica
SQLServer:Database Replica Recovery Queue	SQLServer:Database Replica

Nome del parametro del contatore di prestazioni	Nome del set di contatori di prestazioni
SQLServer:Database Replica Redo Bytes Remaining	SQLServer:Database Replica
SQLServer:Database Replica Redone Bytes/sec	SQLServer:Database Replica
SQLServer:Database Replica Total Log requiring undo	SQLServer:Database Replica
SQLServer:Database Replica Transaction Delay	SQLServer:Database Replica
SQLServer:General Statistics Processes Blocked	SQLServer:General Statistics
SQLServer:General Statistics User Connections	SQLServer:General Statistics
SQLServer:Latches Average Latch Wait Time (ms)	SQLServer:Latches
SQLServer:Locks Average Wait Time (ms)	SQLServer:Locks
SQLServer:Locks Lock Timeouts/Sec	SQLServer:Locks
SQLServer:Locks Lock Waits/Sec	SQLServer:Locks
SQLServer:Locks Number of Deadlocks/Sec	SQLServer:Locks
SQLServer:Memory Manager Memory Grants Pending	SQLServer:Memory Manager
SQLServer:SQL Statistics Batch Requests/Sec	SQLServer: statistiche SQL
SQLServer:SQL Statistics SQL Compilations/Sec	SQLServer: statistiche SQL

Nome del parametro del contatore di prestazioni	Nome del set di contatori di prestazioni
SQLServer:SQL Statistics SQL Re-Compilations/Sec	SQLServer: statistiche SQL
Lunghezza coda processore di sistema	System (Sistema)
Connessioni TCPv4 stabilite	TCPv4
Connessioni TCPv6 stabilite	TCPv6
Svuotamenti della cache dei file W3SVC_W3WP	W3SVC_W3WP
Mancati riscontri della cache dei file W3SVC_W3WP	W3SVC_W3WP
Richieste W3SVC_W3WP/sec	W3SVC_W3WP
Svuotamenti della cache URI W3SVC_W3WP	W3SVC_W3WP
Mancati riscontri della cache URI W3SVC_W3WP	W3SVC_W3WP
Byte del servizio Web ricevuti/sec	Servizio Web
Byte del servizio Web inviati/sec	Servizio Web
Tentativi di connessione/sec al servizio Web	Servizio Web
Connessioni correnti del servizio Web	Servizio Web
Richieste Get al secondo del servizio Web	Servizio Web
Richieste Post al secondo del servizio Web	Servizio Web

Utilizzo della visualizzazione dello stato delle risorse nella CloudWatch console

Puoi utilizzare la visualizzazione dell'integrità delle risorse per individuare, gestire e visualizzare automaticamente l'integrità e le prestazioni degli host nelle relative applicazioni in un'unica visualizzazione. Puoi visualizzare lo stato dei loro host in base a una dimensione delle prestazioni, ad esempio CPU o memoria, e suddividere centinaia di host in un'unica vista utilizzando i filtri. Puoi filtrare gli elementi in base ai tag o ai casi d'uso, ad esempio gli host dello stesso gruppo con scalabilità automatica o gli host che utilizzano lo stesso sistema di bilanciamento del carico,

Prerequisiti

Per accertati di ottenere i vantaggi della visualizzazione dell'integrità delle risorse, accertati che siano soddisfatti i seguenti prerequisiti.

- Per visualizzare l'utilizzo della memoria degli host e utilizzarla come filtro, è necessario installare l' CloudWatch agente sugli host e configurarlo per l'invio di una metrica di memoria CloudWatch nello spazio dei nomi predefinito`CWAgent`. Nelle istanze Linux e macOS, l' CloudWatch agente deve inviare la metrica. `mem_used_percent` Nelle istanze di Windows, l'agente deve inviare il parametro `Memory % Committed Bytes In Use`. Queste metriche sono incluse se si utilizza la procedura guidata per creare il file di configurazione dell' CloudWatch agente e selezionare uno dei set di metriche predefiniti. Le metriche raccolte dall' CloudWatch agente vengono fatturate come metriche personalizzate. Per ulteriori informazioni, consulta Installazione dell'agente CloudWatch .

  Quando si utilizza l' CloudWatch agente per raccogliere queste metriche di memoria da utilizzare con la visualizzazione dello stato delle risorse, è necessario includere la sezione seguente nel file di configurazione dell' CloudWatch agente. Questa sezione contiene le impostazioni delle dimensioni predefinite e viene creata per impostazione predefinita, quindi non occorre modificare alcuna parte di questa sezione con elementi diversi da quelli mostrati nell'esempio seguente.

  ```
  "append_dimensions": {
    "ImageId": "${aws:ImageId}",
    "InstanceId": "${aws:InstanceId}",
    "InstanceType": "${aws:InstanceType}",
    "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
  },
  ```

- Per visualizzare tutte le informazioni disponibili nella visualizzazione dell'integrità delle risorse, devi aver effettuato l'accesso a un account che dispone delle seguenti autorizzazioni. Se hai effettuato

la connessione con meno autorizzazioni, puoi comunque utilizzare la visualizzazione dell'integrità delle risorse, ma alcuni dati sulle prestazioni non saranno visibili.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:Describe*",
                "cloudwatch:Describe*",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:Describe*",
                "sns:Get*",
                "sns:List*",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeRegions"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Per visualizzare lo stato delle risorse nel tuo account

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Monitoraggio dellinfrastruttura, Integrità delle risorse.

   Viene visualizzata la pagina relativa all'integrità delle risorse, che mostra un riquadro per ogni host dell'account. Ogni riquadro è colorato in base allo stato corrente di quell'host, in base all'impostazione Color by (Colora per). I riquadri dell'host con un simbolo di allarme hanno uno o più allarmi attualmente in stato ALARM.

   È possibile visualizzare fino a 500 host in un'unica vista. Se hai più host nell'account, utilizza le impostazioni del filtro nel passaggio 6 di questa procedura.

3. Per modificare i criteri utilizzati per mostrare l'integrità di ciascun host, scegli un'impostazione per Color by (Colora per). Puoi scegliere CPU Utilization (Utilizzo CPU), Memory Utilization (Utilizzo della memoria), oppure Status check (Controllo dello stato). Le metriche di utilizzo della memoria sono disponibili solo per gli host che eseguono l' CloudWatch agente e lo hanno configurato per raccogliere i parametri della memoria e inviarli allo spazio dei nomi predefinito. `CWAgent` Per ulteriori informazioni, consulta Raccogli metriche, log e tracce con l'agente CloudWatch .
4. Per modificare le soglie e i colori utilizzati per gli indicatori di integrità nella griglia, scegli l'icona a forma di ingranaggio sopra la griglia.
5. Per attivare o disattivare la visualizzazione degli allarmi nella griglia degli host, seleziona o deseleziona Show alarms across all metrics (Mostra allarmi in tutti i parametri).
6. Per suddividere gli host nella mappa in gruppi, scegli un criterio di raggruppamento per Group by (Raggruppa per).
7. Per restringere la vista a un numero inferiore di host, scegli un criterio di filtro per Filter by (Filtra per). Puoi filtrare gli elementi in base ai tag e ai raggruppamenti di risorse, ad esempio gruppo Auto Scaling, tipo di istanza, gruppo di sicurezza e altro ancora.
8. Per ordinare gli host, scegli un criterio di ordinamento per Sort by (Ordina per). Puoi ordinare gli elementi in base ai risultati del controllo dello stato, allo stato dell'istanza, all'utilizzo della CPU o della memoria e al numero di allarmi in stato ALARM.
9. Per visualizzare ulteriori informazioni su un host, scegli il riquadro che rappresenta tale host. Viene visualizzato un riquadro popup. Per approfondire le informazioni sull'host, scegli View dashboard (Visualizza pannello di controllo) o View on list (Visualizza nell'elenco).

CloudWatch osservabilità tra più account

Con l'osservabilità CloudWatch tra più account di Amazon, puoi monitorare e risolvere i problemi delle applicazioni che si estendono su più account all'interno di una regione. Cerca, visualizza e analizza senza problemi metriche, log, tracce, applicazioni Application Insights e monitor Internet Monitor in qualsiasi account collegato senza limiti di account.

Configura uno o più AWS account come account di monitoraggio e collegali a più account di origine. Un account di monitoraggio è un account AWS principale in grado di visualizzare e interagire con i dati di osservabilità generati dagli account di origine. Un account di origine è un AWS account individuale che genera dati di osservabilità per le risorse che vi risiedono. Gli account di origine condividono i dati di osservabilità con l'account di monitoraggio. I dati di osservabilità condivisi possono includere i seguenti tipi di dati di telemetria:

- Metriche in Amazon CloudWatch. Puoi scegliere di condividere le metriche di tutti i namespace con l'account di monitoraggio o filtrare in base a un sottoinsieme di namespace.
- Gruppi di log in Amazon CloudWatch Logs. Puoi scegliere di condividere tutti i gruppi di log con l'account di monitoraggio o filtrare in base a un sottoinsieme di gruppi di log.
- Tracce in AWS X-Ray
- Applicazioni in Amazon CloudWatch Application Insights
- Monitor in CloudWatch Internet Monitor

Per creare collegamenti tra gli account di monitoraggio e gli account di origine, puoi utilizzare la CloudWatch console. In alternativa, utilizza i comandi di Observability Access Manager nell'API AWS CLI and. Per ulteriori informazioni, consulta Documentazione di riferimento delle API di Observability Access Manager.

Un sink è una risorsa che rappresenta un punto di attacco in un account di monitoraggio. Gli account di origine possono collegarsi al sink per condividere i dati di osservabilità. Ogni account può avere un sink per regione. Ogni sink è gestito dall'account di monitoraggio in cui si trova. Un link di osservabilità è una risorsa che rappresenta il collegamento stabilito tra un account di origine e un account di monitoraggio. I collegamenti sono gestiti dall'account di origine.

Per una dimostrazione video sulla configurazione dell'osservabilità CloudWatch tra più account, guarda il video seguente.

L'argomento successivo spiega come impostare l'osservabilità tra account sia CloudWatch negli account di monitoraggio che negli account di origine. Per informazioni sulla dashboard interregionale CloudWatch tra account, consulta. Console per più account e più regioni CloudWatch

Utilizzo di Organizations per gli account di origine

Sono disponibili due opzioni per collegare gli account di origine all'account di monitoraggio. Puoi utilizzare una o entrambe le opzioni.

- Utilizzato AWS Organizations per collegare gli account di un'organizzazione o unità organizzativa all'account di monitoraggio.
- Connect AWS i singoli account all'account di monitoraggio.

Ti consigliamo di utilizzare Organizations in modo che AWS i nuovi account creati in un secondo momento nell'organizzazione vengano automaticamente inseriti come account di origine per consentire l'osservabilità tra più account.

Dettagli sul collegamento di account di monitoraggio e account di origine

- Ogni account di monitoraggio può essere collegato a un massimo di 100.000 account di origine.
- Ogni account di origine può condividere dati con un massimo di cinque account di monitoraggio.
- Puoi configurare un singolo account sia come account di monitoraggio che come account di origine. In questo caso, tale account invia all'account di monitoraggio collegato soltanto i dati di osservabilità provenienti da se stesso, non dai relativi account di origine.
- Un account di monitoraggio specifica i dati di telemetria che possono essere condivisi con esso. Un account di origine specifica i dati di telemetria che desidera condividere.
  - Se nell'account di monitoraggio sono selezionati più tipi di telemetria rispetto all'account di origine, gli account sono collegati. Vengono condivisi solo i tipi di dati selezionati in entrambi gli account.
  - Se nell'account di origine sono selezionati più tipi di dati di telemetria rispetto all'account di monitoraggio, la creazione del collegamento ha esito negativo e non viene condiviso nulla.
  - Il nome di una metrica non viene visualizzato nella console dell'account di monitoraggio finché tale metrica non emette nuovi punti dati dopo la creazione del collegamento.
- Per rimuovere un collegamento tra account, esegui questa operazione dall'account di origine.
- Per eliminare un sink in un account di monitoraggio, devi prima rimuovere tutti i link che rimandano a quel sink dell'account di monitoraggio.

Prezzi

L'osservabilità tra più account non CloudWatch comporta costi aggiuntivi per log e metriche e la prima copia di traccia è gratuita. Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatchdei prezzi di Amazon.

Indice

Collegamento degli account di monitoraggio agli account di origine

Gli argomenti di questa sezione illustrano come configurare i collegamenti tra account di monitoraggio e account di origine.

Ti consigliamo di creare un nuovo AWS account che funga da account di monitoraggio per la tua organizzazione.

Indice

- Passaggio 2: (Facoltativo) Scarica un modello o un URL AWS CloudFormation
- Passaggio 3: collegamento degli account di origine
  - Utilizzo di un modello AWS CloudFormation per configurare tutti gli account di un'organizzazione o di un'unità organizzativa come account di origine
  - Utilizzo di un modello AWS CloudFormation per configurare singoli account di origine
  - Utilizzo di un URL per configurare singoli account di origine

Autorizzazioni necessarie

Per creare un collegamento tra un account di monitoraggio e un account di origine, devi accedere con determinate autorizzazioni.

- Per configurare un account di monitoraggio. È necessario disporre dell'accesso completo come amministratore nell'account di monitoraggio oppure accedere a tale account con le seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSinkModification",
            "Effect": "Allow",
            "Action": [
                "oam:CreateSink",
                "oam:DeleteSink",
                "oam:PutSinkPolicy",
                "oam:TagResource"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowReadOnly",
            "Effect": "Allow",
            "Action": ["oam:Get*", "oam:List*"],
            "Resource": "*"
        }
    ]
}
```

- Account di origine associato a un account di monitoraggio specifico. Per creare, aggiornare e gestire i collegamenti per un solo account di monitoraggio specificato, è necessario accedere all'account almeno con le autorizzazioni seguenti. In questo esempio, l'account di monitoraggio è 999999999999.

  Se il link non intende condividere tutti e cinque i tipi di risorse (metriche, log, tracce, applicazioni Application Insights e monitor Internet Monitor), puoi omettere`cloudwatch:Link`, `logs:Link` `xray:Link`applicationinsights:Link, o se necessario. `internetmonitor:Link`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink",
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:link/*"
        },
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:sink/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": [
                        "999999999999"
                    ]
                }
            }
        },
        {
            "Action": "oam:ListLinks",
            "Effect": "Allow",
            "Resource": "*"
```

```
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
         },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

- Account di origine, con autorizzazioni per il collegamento a qualsiasi account di monitoraggio: per creare un collegamento a qualsiasi account di monitoraggio esistente, analizzare e condividere metriche, gruppi di log, tracce, applicazioni Application Insights e monitor Internet Monitor, è necessario accedere all'account di origine con le autorizzazioni complete di amministratore o accedervi con le seguenti autorizzazioni

  Se il link non intende condividere tutti e cinque i tipi di risorse (metriche, log, tracce, applicazioni Application Insights e monitor Internet Monitor), puoi omettere `cloudwatch:Linklogs:Link`,`xray:Link`,`applicationinsights:Link`, o `internetmonitor:Link` se necessario.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:List*",
                "oam:Get*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
```

```
            "Action": "applicationinsights:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "internetmonitor:Link",
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Panoramica della configurazione

I seguenti passaggi di alto livello mostrano come configurare l'osservabilità tra account. CloudWatch

> **Note**
>
> Ti consigliamo di creare un nuovo AWS account da utilizzare come account di monitoraggio della tua organizzazione.

1. Configura un account di monitoraggio dedicato.
2. (Facoltativo) Scarica un AWS CloudFormation modello o copia un URL per collegare gli account di origine.
3. Collega gli account di origine all'account di monitoraggio.

Dopo aver completato questi passaggi, puoi utilizzare l'account di monitoraggio per visualizzare i dati di osservabilità degli account di origine.

Passaggio 1: configurazione di un account di monitoraggio

Segui i passaggi in questa sezione per configurare un AWS account come account di monitoraggio per l'osservabilità CloudWatch tra account.

Prerequisiti

- Se stai configurando gli account di un' AWS Organizations organizzazione come account di origine, ottieni il percorso dell'organizzazione o l'ID dell'organizzazione.

- Se non utilizzi Organizations per gli account di origine, ottieni gli ID degli account di origine.

Per configurare un account come account di monitoraggio, devi disporre di determinate autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni necessarie.

Per configurare un account di monitoraggio

1. Accedi all'account da utilizzare come account di monitoraggio.
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel riquadro di navigazione a sinistra scegliere Impostazioni.
4. In Monitoring account configuration (Configurazione dell'account di monitoraggio), scegli Configure (Configura).
5. Per Select data, scegli se questo account di monitoraggio sarà in grado di visualizzare i log, le metriche, le tracce, Application Insights - Applications e Internet Monitor - Monitors i dati degli account di origine a cui è collegato.
6. In List source accounts (Elenca account di origine), inserisci gli account di origine che verranno visualizzati da questo account di monitoraggio. Per identificare gli account di origine, inserisci gli ID dei singoli account, i percorsi dell'organizzazione o gli ID dell'organizzazione. Se inserisci un percorso o un ID dell'organizzazione, l'account di monitoraggio può visualizzare i dati di osservabilità da tutti gli account collegati in tale organizzazione.

   Separa le voci in elenco con virgole.

   > **Important**
   >
   > Quando inserisci un percorso organizzativo, segui il formato esatto. L'ou-id deve terminare con un / (un carattere barra). Ad esempio: `o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-def0-awsbbbb/`

7. In Define a label to identify your source account (Definisci un'etichetta per identificare l'account di origine), specifica se identificare gli account di origine tramite i nomi account o gli indirizzi e-mail quando usi l'account di monitoraggio per visualizzarli.
8. Scegli Configura.

> **Important**
>
> Il collegamento tra gli account di monitoraggio e di origine non è completo finché non si configurano gli account di origine. Per ulteriori informazioni, consultare le sezioni indicate di seguito.

Passaggio 2: (Facoltativo) Scarica un modello o un URL AWS CloudFormation

Per collegare gli account di origine a un account di monitoraggio, ti consigliamo di utilizzare un modello AWS CloudFormation o un URL.

- Se stai collegando un'intera organizzazione, CloudWatch fornisce un AWS CloudFormation modello.
- Se stai collegando singoli account, utilizza un AWS CloudFormation modello o un URL che CloudWatch lo fornisca.

Per utilizzare un AWS CloudFormation modello, devi scaricarlo durante questi passaggi. Dopo aver collegato l'account di monitoraggio ad almeno un account di origine, il AWS CloudFormation modello non è più disponibile per il download.

Per scaricare un AWS CloudFormation modello o copiare un URL per collegare gli account di origine all'account di monitoraggio

1. Accedi all'account da utilizzare come account di monitoraggio.
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel riquadro di navigazione a sinistra scegliere Impostazioni.
4. In Monitoring account configuration (Configurazione dell'account di monitoraggio), scegli Resources to link accounts (Risorse per collegare gli account).
5. Esegui una di queste operazioni:

   - Scegli Organizzazione AWS per scaricare un modello da utilizzare per collegare gli account di un'organizzazione a questo account di monitoraggio.
   - Scegli Any account (Qualsiasi account) per ottenere un modello o un URL al fine di configurare i singoli account come account di origine.

6. Esegui una di queste operazioni:
   - Se hai scelto AWS l'organizzazione, scegli Scarica CloudFormation modello.
   - Se hai scelto Qualsiasi account, scegli Scarica CloudFormation modello o Copia URL.
7. (Facoltativo) Ripeti i passaggi 5-6 per scaricare sia il AWS CloudFormation modello che l'URL.

Passaggio 3: collegamento degli account di origine

Utilizza la procedura riportata in queste sezioni per collegare gli account di origine a un account di monitoraggio.

Per collegare gli account di monitoraggio agli account di origine, devi disporre di determinate autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni necessarie.

Utilizzo di un modello AWS CloudFormation per configurare tutti gli account di un'organizzazione o di un'unità organizzativa come account di origine

Questi passaggi presuppongono che tu abbia già scaricato il AWS CloudFormation modello necessario eseguendo i passaggi indicati inPassaggio 2: (Facoltativo) Scarica un modello o un URL AWS CloudFormation.

Per utilizzare un AWS CloudFormation modello per collegare gli account di un'organizzazione o di un'unità organizzativa all'account di monitoraggio

1. Accedi all'account di gestione dell'organizzazione.
2. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
3. Nella barra di navigazione a sinistra, scegli StackSets.
4. Verifica di aver effettuato l'accesso alla regione desiderata, quindi scegli Crea StackSet.
5. Seleziona Successivo.
6. Scegli Template is ready (Il modello è pronto) e infine Upload a template file (Carica un file di modello).
7. Seleziona Choose file (Scegli file), scegli il modello che hai scaricato dall'account di monitoraggio e infine scegli Open (Apri).
8. Seleziona Successivo.
9. Per Specificare StackSet i dettagli, inserisci un nome per la StackSet e scegli Avanti.

10. In Add stacks to stack set (Aggiungi stack a un set di stack), scegli Deploy new stacks (Implementa nuovi stack).
11. In Deployment targets (Destinazioni di implementazione), scegli se distribuirlo all'intera organizzazione o a unità organizzative specifiche.
12. Per Specificare le regioni, scegli in quali regioni implementare l' CloudWatch osservabilità tra account.
13. Seleziona Successivo.
14. Nella pagina Review (Revisione), conferma le opzioni e selezionate scegli Submit (Invia).
15. Nella scheda Stack instances (Istanze stack), aggiorna la schermata finché non viene visualizzato lo stato CREATE_COMPLETE per le istanze dello stack.

Utilizzo di un modello AWS CloudFormation per configurare singoli account di origine

Questi passaggi presuppongono che tu abbia già scaricato il AWS CloudFormation modello necessario eseguendo i passaggi indicati in. Passaggio 2: (Facoltativo) Scarica un modello o un URL AWS CloudFormation

Utilizzare un AWS CloudFormation modello per configurare account di origine individuali per l'osservabilità CloudWatch tra account

1. Accedi all'account di origine.
2. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
3. Nella barra di navigazione a sinistra, seleziona Stacks (Stack).
4. Verifica di aver effettuato l'accesso alla regione desiderata, quindi scegli Create StackSet (Crea StackSet) e infine With new resources (standard) (Con nuove risorse [standard]).
5. Seleziona Successivo.
6. Scegliere Upload a template file (Carica un file di modello).
7. Seleziona Choose file (Scegli file), scegli il modello che hai scaricato dall'account di monitoraggio e infine scegli Open (Apri).
8. Seleziona Successivo.
9. In Specify stack details (Specifica i dettagli dello stack), inserisci un nome per lo stack e seleziona Next (Successivo).
10. Nella pagina Configure stack options (Configura opzioni pila), scegliere Next (Successivo).

11. Nella pagina Review (Revisione), scegli Submit (Invia).
12. Nella pagina relativa allo stato dello stack, aggiorna la schermata finché non viene visualizzato lo stato CREATE_COMPLETE.
13. Per collegare più account di origine a questo account di monitoraggio utilizzando lo stesso modello, esci da questo account e accedi all'account di origine successivo. Quindi ripeti i passaggi 2-12.

Utilizzo di un URL per configurare singoli account di origine

Questi passaggi presuppongono che tu abbia già copiato l'URL necessario seguendo le fasi descritte in Passaggio 2: (Facoltativo) Scarica un modello o un URL AWS CloudFormation.

Per utilizzare un URL al fine di collegare i singoli account di origine all'account di monitoraggio

1. Accedi all'account da utilizzare come account di origine.
2. Inserisci l'URL copiato dall'account di monitoraggio.

   Viene visualizzata la pagina CloudWatch delle impostazioni, con alcune informazioni inserite.
3. Per Seleziona dati, scegli se questo account di origine condividerà i dati di log, metriche, tracce, Application Insights - Applications e Internet Monitor - Monitor - Monitor con questo account di monitoraggio.

   Sia per Logs che per Metrics, puoi scegliere se condividere tutte le risorse o un sottoinsieme con l'account di monitoraggio.

   a. (Facoltativo) Per condividere un sottoinsieme dei gruppi di log di questo account con l'account di monitoraggio, seleziona Registri e scegli Filtra registri. Quindi usa la casella Filtra i registri per creare una query per trovare i gruppi di log che desideri condividere. La query utilizzerà il termine `LogGroupName` e uno o più dei seguenti operandi.

      - `=` e `!=`
      - `AND`
      - `OR`
      - `^`indica LIKE e `!^` indica NOT LIKE. Questi possono essere usati solo come ricerche con prefissi. Includi un `%` alla fine della stringa che desideri cercare e includere.
      - `IN`e`NOT IN`, usando le parentesi () `(  )`

L'interrogazione completa non deve superare i 2000 caratteri ed è limitata a cinque operandi condizionali. Gli operandi condizionali sono e. `AND` `OR` Non esiste un limite al numero di altri operandi.

> **Tip**
>
> Scegliete Visualizza query di esempio per visualizzare la sintassi corretta per i formati di query più comuni.

b. (Facoltativo) Per condividere un sottoinsieme dei namespace delle metriche di questo account con l'account di monitoraggio, seleziona Metriche e scegli Filtra metriche. Quindi utilizza la casella Filtra metriche per creare una query per trovare i namespace delle metriche che desideri condividere. Utilizzate il termine `Namespace` e uno o più dei seguenti operandi.

- `=` e `!=`
- `AND`
- `OR`
- `LIKE` e `NOT LIKE`. Questi possono essere utilizzati solo come ricerche per prefissi. Includi un `%` alla fine della stringa che desideri cercare e includere.
- `IN`e`NOT IN`, usando le parentesi () `(  )`

L'interrogazione completa non deve superare i 2000 caratteri ed è limitata a cinque operandi condizionali. Gli operandi condizionali sono e. `AND` `OR` Non esiste un limite al numero di altri operandi.

> **Tip**
>
> Scegliete Visualizza query di esempio per visualizzare la sintassi corretta per i formati di query più comuni.

4. Non modificare l'ARN in Enter monitoring account configuration ARN (Inserisci l'ARN di configurazione dell'account di monitoraggio).

5. La sezione Define a label to identify your source account (Definisci un'etichetta per identificare l'account di origine) è precompilata con l'etichetta scelta dall'account di monitoraggio. Se lo desideri, puoi modificarla selezionando Edit (Modifica).
6. Scegliere Link (Collegamento).
7. Nella casella, inserisci **`Confirm`** e scegli Confirm (Conferma).
8. Per collegare più account di origine a questo account di monitoraggio utilizzando lo stesso URL, esci da questo account e accedi all'account di origine successivo. Quindi ripeti i passaggi 2-7.

Gestione degli account di monitoraggio e di origine

Dopo aver impostato gli account di monitoraggio e di origine, puoi utilizzare i passaggi descritti in queste sezioni per gestirli.

Indice

Collegamento di molteplici account di origine a un account di monitoraggio esistente

Segui la procedura riportata in questa sezione per aggiungere collegamenti da account di origine aggiuntivi a un account di monitoraggio esistente.

Ogni account di origine può essere collegato a un massimo di cinque account di monitoraggio. Ogni account di monitoraggio può essere collegato a un massimo di 100.000 account di origine.

Per gestire un account di origine, devi disporre di determinate autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni necessarie.

Per aggiungere altri account di origine a un account di monitoraggio

1. Accedi all'account di monitoraggio.
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel riquadro di navigazione a sinistra scegliere Impostazioni.

4. In Monitoring account configuration (Configurazione dell'account di monitoraggio), scegli Manage source accounts (Gestisci account di origine).
5. Scegli la scheda Configuration policy (Policy di configurazione).
6. Nella casella Configuration policy (Policy di configurazione), aggiungi il nuovo ID dell'account di origine nella riga Principal (Principale).

   Ad esempio, supponiamo che la riga Principal (Principale) sia attualmente la seguente:

   ```
   "Principal": {"AWS": ["111111111111", "222222222222"]}
   ```

   Per aggiungere 999999999999 come terzo account di origine, modifica la riga come segue:

   ```
   "Principal": {"AWS": ["111111111111", "222222222222", "999999999999"]}
   ```
7. Scegli Aggiorna.
8. Scegli la scheda Configuration details (Dettagli della configurazione).
9. Scegli l'icona di copia accanto all'ARN del sink dell'account di monitoraggio.
10. Accedi all'account da utilizzare come nuovo account di origine.
11. Incolla l'ARN del sink dell'account di monitoraggio che hai copiato nel passaggio 9.

    Viene visualizzata la pagina CloudWatch delle impostazioni, con alcune informazioni inserite.
12. Per Seleziona dati, scegli se questo account di origine invierà i dati di Log, Parametri, Tracce e Approfondimenti sulle applicazioni - Applicazioni agli account di monitoraggio a cui è collegato.
13. Non modificare l'ARN in Enter monitoring account configuration ARN (Inserisci l'ARN di configurazione dell'account di monitoraggio).
14. La sezione Define a label to identify your source account (Definisci un'etichetta per identificare l'account di origine) è precompilata con l'etichetta scelta dall'account di monitoraggio. Se lo desideri, puoi modificarla selezionando Edit (Modifica).
15. Scegliere Link (Collegamento).
16. Nella casella, inserisci **`Confirm`** e scegli Confirm (Conferma).

Rimozione del collegamento tra un account di monitoraggio e un account di origine

Segui la procedura riportata in questa sezione per interrompere l'invio dei dati da un account di origine a un account di monitoraggio.

Per completare questa attività, devi disporre delle autorizzazioni necessarie per gestire un account di origine. Per ulteriori informazioni, consulta Autorizzazioni necessarie.

Per rimuovere il collegamento tra un account di origine e un account di monitoraggio

1. Accedi all'account di origine.
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel riquadro di navigazione a sinistra scegliere Impostazioni.
4. In Source account information (Informazioni sull'account di origine), scegli View monitoring accounts (Visualizza account di monitoraggio).
5. Seleziona la casella di controllo accanto all'account di monitoraggio con cui si desidera interrompere la condivisione dei dati.
6. Scegli Stop sharing data (Interrompi la condivisione dei dati), quindi Confirm (Conferma).
7. Accedi all'account di monitoraggio.
8. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
9. Seleziona Impostazioni.
10. In Monitoring account information (Informazioni sull'account di monitoraggio), scegli View configuration (Visualizza la configurazione).
11. Nella casella Policy, elimina l'ID dell'account di origine dalla riga Principal (Principale) e scegli Update (Aggiorna).

Visualizzazione delle informazioni relative a un account di monitoraggio

Segui la procedura riportata in questa sezione per visualizzare le impostazioni multi-account di un account di monitoraggio.

Per gestire un account di monitoraggio, devi disporre di determinate autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni necessarie.

Per gestire un account di monitoraggio

1. Accedi all'account di monitoraggio.
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel riquadro di navigazione a sinistra scegliere Impostazioni.
4. In Monitoring account configuration (Configurazione dell'account di monitoraggio), scegli Manage source accounts (Gestisci account di origine).
5. Per visualizzare la policy di Observability Access Manager che consente di utilizzare questo account come un account di monitoraggio, scegli la scheda Configuration policy (Policy di configurazione).
6. Per visualizzare gli account di origine collegati a questo account di monitoraggio, scegli la scheda Linked source accounts (Account di origine collegati).
7. Per visualizzare l'ARN sink dell'account di monitoraggio e i tipi di dati che l'account di monitoraggio può visualizzare negli account di origine collegati, scegli la scheda Linked source accounts (Account di origine collegati).

Recupero dei parametri da altre origini dati

Puoi utilizzarlo CloudWatch per interrogare, visualizzare e creare allarmi per metriche provenienti da altre fonti di dati. Per farlo, ti connetti CloudWatch alle altre fonti di dati. Questo ti offre un'esperienza di monitoraggio unica e consolidata all'interno della CloudWatch console. Puoi avere una visualizzazione unificata dei parametri dell'infrastruttura e delle applicazioni a prescindere da dove sono archiviati i dati, per identificare e risolvere i problemi più rapidamente.

Dopo esserti connesso a una fonte di dati tramite una CloudWatch procedura guidata, CloudWatch crea uno AWS CloudFormation stack che distribuisce e configura una funzione. AWS Lambda Questa funzione Lambda viene eseguita su richiesta ogni volta che si esegue una query sull'origine dati. Il generatore di CloudWatch query mostra in tempo reale un elenco di elementi su cui è possibile eseguire query, come metriche, tabelle, campi o etichette. Man mano che effettui delle scelte, il generatore di query precompila una query nella lingua nativa dell'origine selezionata.

CloudWatch fornisce procedure guidate per la connessione alle seguenti fonti di dati. Per queste origini dati, fornisci informazioni di base per identificare l'origine dati e le credenziali. Puoi anche creare manualmente connettori ad altre origini dati creando le tue funzioni Lambda.

- Amazon OpenSearch Service: ricava i parametri dai log e dalle tracce OpenSearch del tuo servizio.
- Amazon Managed Service per Prometheus: consulta questi parametri utilizzando PromQL.
- Amazon RDS per MySQL: usa SQL per convertire i dati archiviati nelle tabelle Amazon RDS in parametri.
- Amazon RDS per PostgreSQL: usa SQL per convertire i dati archiviati nelle tabelle Amazon RDS in parametri.
- File CSV di Amazon S3: visualizza i dati dei parametri da un file CSV archiviato in un bucket Amazon S3.
- Microsoft Azure Monitor: esegui query sui parametri del tuo account Microsoft Azure Monitor.
- Prometheus: consulta questi parametri utilizzando PromQL.

Dopo aver creato i connettori alle origini dati, consulta Creazione di un grafico dei parametri da un'altra origine dati per informazioni sulla rappresentazione grafica di un parametro da un'origine dati. Per informazioni sull'impostazione di un allarme su un parametro proveniente da un'origine dati, consulta Creazione di un allarme basato su un'origine dati connessa.

Argomenti

Gestione dell'accesso alle origini dati

CloudWatch utilizza AWS CloudFormation per creare le risorse richieste nel tuo account. Ti consigliamo di utilizzare la `cloudformation:TemplateUrl` condizione per controllare l'accesso ai AWS CloudFormation modelli quando concedi `CreateStack` le autorizzazioni agli utenti IAM.

> **Warning**
>
> Qualsiasi utente a cui concedi l'autorizzazione per richiamare l'origine dati può interrogare i parametri di tale origine di dati anche se non dispone delle autorizzazioni IAM dirette per quest'ultima. Ad esempio, se concedi le autorizzazioni `lambda:InvokeFunction` per una funzione Lambda dell'origine dati Amazon Managed Service per Prometheus a un utente, quell'utente sarà in grado di eseguire query sui parametri dall'area di lavoro Amazon Managed Service per Prometheus corrispondente anche se non gli hai concesso l'accesso IAM diretto a quell'area di lavoro.

Puoi trovare gli URL dei modelli per le fonti di dati nella pagina Crea stack nella Console delle CloudWatch impostazioni.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "cloudformation:TemplateUrl" : [ data-source-template-url ]
        }
      }
```

```
    }
  ]
}
```

Per ulteriori informazioni sul controllo degli AWS CloudFormation accessi, vedere Controllo dell'accesso con AWS Identity and Access Management

Connessione a un'origine dati predefinita con una procedura guidata

Questo argomento fornisce istruzioni per l'utilizzo della procedura guidata per la connessione CloudWatch alle seguenti fonti di dati.

- OpenSearch Servizio Amazon
- Amazon Managed Service per Prometheus
- Amazon RDS per MySQL
- Amazon RDS per PostgreSQL
- File CSV di Amazon S3
- Microsoft Azure Monitor
- Prometheus

Più avanti in questa sezione sono riportate delle sottosezioni con note sulla gestione e l'esecuzione di query con ciascuna di queste origini dati.

Per creare un connettore a un'origine dati

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Impostazioni.
3. Seleziona la scheda Origini dati dei parametri.
4. Seleziona Create data source (Crea origine dati).
5. Seleziona l'origine che desideri, quindi scegli Successivo.
6. Inserisci un nome per l'origine dati.
7. Inserisci le altre informazioni richieste, a seconda dell'origine dati che hai scelto. Queste possono includere credenziali per accedere all'origine dati e alle informazioni di identificazione dell'origine

dati, come il nome dell'area di lavoro Prometheus, del database o del bucket Amazon S3. Per AWS i servizi, la procedura guidata rileva le risorse e le inserisce nel menu a discesa di selezione.

Per ulteriori note sull'origine dati che stai utilizzando, consulta le sezioni successive a questa procedura.

8. Per CloudWatch connetterti alla sorgente dati in un VPC, scegli Usa un VPC e seleziona il VPC da usare. Quindi seleziona la sottorete e il gruppo di sicurezza.
9. Scegli Riconosco che AWS CloudFormation creerà risorse IAM. Questa risorsa è il ruolo di esecuzione della funzione Lambda.
10. Seleziona Create data source (Crea origine dati).

    La nuova fonte che hai appena aggiunto non viene visualizzata finché lo AWS CloudFormation stack non ha finito di crearla. Per verificare i progressi, puoi scegliere Visualizza lo stato del mio CloudFormation stack. In alternativa, puoi selezionare l'icona di aggiornamento per aggiornare questo elenco.

    Quando la nuova origine dati viene visualizzata nell'elenco, è pronta per essere utilizzata. Puoi scegliere Query tra le CloudWatch metriche per iniziare a eseguire query con essa. Per ulteriori informazioni, consulta Creazione di un grafico dei parametri da un'altra origine dati.

Amazon Managed Service per Prometheus

Aggiornamento della configurazione dell'origine dati

- Puoi aggiornare l'origine dati manualmente con le seguenti operazioni:
  - Per aggiornare l'ID dell'area di lavoro Amazon Managed Service per Prometheus, aggiorna la variabile di ambiente `AMAZON_PROMETHEUS_WORKSPACE_ID` per la funzione Lambda del connettore di origine dati.
  - Per aggiornare la configurazione VPC, consulta Configurazione dell'accesso VPC (console) per ulteriori informazioni.

Esecuzione di query sull'origine dati

- Quando esegui una query su Amazon Managed Service per Prometheus, dopo aver selezionato l'origine dati nella scheda Query da più origini e selezionato un connettore Amazon Managed Service per Prometheus, puoi utilizzare la Guida alle query per scoprire parametri ed etichette e

fornire semplici query PromQL. Puoi anche utilizzare l'editor di query PromQL per creare una query PromQL.

- Le interrogazioni multilinea non sono supportate dai connettori delle sorgenti dati. CloudWatch Ogni avanzamento riga viene sostituito da uno spazio quando la query viene eseguita o quando si crea un allarme o un widget del pannello di controllo con la query. In alcuni casi, ciò potrebbe rendere la query non valida. Ad esempio, se la query contiene un commento di una singola riga, non sarà valida. Se provi a creare una dashboard o un allarme con una query su più righe dalla riga di comando o Infrastructure as Code, l'API rifiuta l'azione generando un errore di analisi.

OpenSearch Servizio Amazon

Creazione di un'origine dati

Se il OpenSearch dominio è abilitato per FGAC, è necessario mappare il ruolo di esecuzione della funzione Lambda del connettore a un utente in Service. OpenSearch Per ulteriori informazioni, consulta la sezione Mappatura degli utenti ai ruoli nella documentazione sulla gestione delle autorizzazioni nella documentazione del servizio. OpenSearch

Se il tuo OpenSearch dominio è accessibile solo all'interno di un Virtual Private Cloud (VPC), devi includere manualmente una nuova variabile di ambiente nella funzione Lambda chiamata. `AMAZON_OPENSEARCH_ENDPOINT` Il valore di questa variabile deve essere il dominio principale dell' OpenSearch endpoint. È possibile ottenere questo dominio radice rimuovendo `https://` e `<region>.es.amazonaws.com` dall'endpoint del dominio elencato nella console di OpenSearch servizio. Ad esempio, se l'endpoint del dominio è`https://sample-domain.us-east-1.es.amazonaws.com`, il dominio principale sarebbe. `sample-domain`

Aggiornamento di un'origine dati

- Puoi aggiornare l'origine dati manualmente con le seguenti operazioni:
  - Per aggiornare il dominio OpenSearch Service, aggiorna la variabile di `AMAZON_OPENSEARCH_DOMAIN_NAME` ambiente per la funzione Lambda del connettore di origine dati.
  - Per aggiornare la configurazione VPC, consulta Configurazione dell'accesso VPC (console) per ulteriori informazioni.

Esecuzione di query sull'origine dati

- Quando esegui una query su OpenSearch Service, dopo aver selezionato l'origine dati nella scheda Query da più fonti, procedi come segue:
  - Seleziona l'indice per eseguire query.
  - Seleziona il nome del parametro (qualsiasi campo numerico nel documento) e le statistiche.
  - Seleziona l'asse del tempo (qualsiasi campo di data nel documento).
  - Seleziona i filtri da applicare (qualsiasi campo di stringa nel documento).
  - Scegli Query a grafo.

Amazon RDS per PostgreSQL e Amazon RDS per MySQL

Creazione di un'origine dati

- Se l'origine dati è accessibile solo in un VPC, è necessario includere la configurazione VPC per il connettore, come descritto in Connessione a un'origine dati predefinita con una procedura guidata. Se l'origine dati deve connettersi al VPC per le credenziali, l'endpoint deve essere configurato nel VPC. Per ulteriori informazioni, consulta Utilizzo di un AWS Secrets Manager endpoint VPC.

  Inoltre, è necessario creare un endpoint VPC per il servizio Amazon RDS. Per ulteriori informazioni, consulta l'API Amazon RDS e gli endpoint AWS PrivateLink VPC di interfaccia ().

Aggiornamento di un'origine dati

- Puoi aggiornare l'origine dati manualmente con le seguenti operazioni:
  - Per aggiornare l'istanza di database, aggiorna la variabile di ambiente `RDS_INSTANCE` per la funzione Lambda del connettore di origine dati.
  - Per aggiornare il nome utente e la password utilizzati per connettersi ad Amazon RDS, usa AWS Secrets Manager. Puoi trovare l'ARN del segreto usato per l'origine dati nella variabile di ambiente `RDS_SECRET` della funzione Lambda dell'origine dati. Per ulteriori informazioni sull'aggiornamento del segreto in AWS Secrets Manager, consulta Modificare un segreto AWS Secrets Manager.
  - Per aggiornare la configurazione VPC, consulta Configurazione dell'accesso VPC (console) per ulteriori informazioni.

Esecuzione di query sull'origine dati

- Quando esegui una query su Amazon RDS, dopo aver selezionato l'origine dati nella scheda Query da più origini e selezionato un connettore Amazon RDS, puoi utilizzare il rilevatore di database per visualizzare database, tabelle e colonne disponibili. Puoi inoltre utilizzare l'editor SQL per creare una query SQL.

  Puoi utilizzare le seguenti variabili nella query:

  - `$start.iso` – L'ora di inizio nel formato data ISO
  - `$end.iso` – L'ora di fine nel formato data ISO
  - `$period` – Il periodo selezionato in secondi

  Ad esempio, è possibile eseguire una query `SELECT value, timestamp FROM table WHERE timestamp BETWEEN $start.iso and $end.iso`
- Le query multilinea non sono supportate dai connettori delle sorgenti dati. CloudWatch Ogni avanzamento riga viene sostituito da uno spazio quando la query viene eseguita o quando si crea un allarme o un widget del pannello di controllo con la query. In alcuni casi, ciò potrebbe rendere la query non valida. Ad esempio, se la query contiene un commento di una singola riga, non sarà valida. Se provi a creare una dashboard o un allarme con una query su più righe dalla riga di comando o Infrastructure as Code, l'API rifiuta l'azione generando un errore di analisi.

> **Note**
>
> Se nei risultati non viene trovato alcun campo relativo alla data, i valori di ogni campo numerico vengono sommati in valori singoli e tracciati nell'intervallo di tempo fornito. Se i timestamp non sono allineati con il periodo selezionato in CloudWatch, i dati vengono aggregati automaticamente utilizzando `SUM` e allineati con il periodo in. CloudWatch

File CSV di Amazon S3

Esecuzione di query sull'origine dati

- Quando esegui una query sui file CSV di Amazon S3, dopo aver selezionato l'origine dati nella scheda Query da più origini e selezionato un connettore Amazon S3, devi selezionare il bucket e la chiave Amazon S3.

  Il file CSV deve essere formattato nei seguenti modi:

  - Il timestamp deve essere la prima colonna.

- La tabella deve avere una riga di intestazione. Le intestazioni vengono utilizzate per denominare le metriche. Il titolo della colonna del timestamp verrà ignorato, verranno utilizzati solo i titoli delle colonne delle metriche.
- I timestamp devono essere in formato data ISO.
- Le metriche devono essere campi numerici.

```
Timestamp, Metric-1, Metric-2, ...
```

Di seguito è riportato un esempio:

timestamp	CPU (%)	Memoria (%)	Archiviazione (%)
2023-11-23T17:09:41+00:00	1	2	3
2023-11-23T17:04:41+00:00	4	5	6
2023-11-23T16:59:41+00:00	7	8	9
2023-11-23T16:54:41+00:00	10	11	12

> **Note**
>
> Se non viene fornito alcun timestamp, i valori di ogni parametro vengono sommati in valori singoli e tracciati nell'intervallo di tempo fornito. Se i timestamp non sono allineati con il periodo selezionato in CloudWatch, i dati vengono aggregati automaticamente utilizzando `SUM` e allineati con il periodo in. CloudWatch

Microsoft Azure Monitor

Creazione di un'origine dati

- È necessario fornire l'ID tenant, l'ID client e il segreto client per connettersi a Microsoft Azure Monitor. Le credenziali verranno archiviate in. AWS Secrets Manager Per maggiori informazioni, consulta Creazione di un'applicazione Microsoft Entra e di un principale del servizio in grado di accedere alle risorse nella documentazione di Microsoft.

Aggiornamento di un'origine dati

- Puoi aggiornare l'origine dati manualmente con le seguenti operazioni:
  - Per aggiornare l'ID tenant, l'ID client e il segreto client usati per connettersi ad Azure Monitor, puoi trovare l'ARN del segreto usato per l'origine dati come variabile di ambiente `AZURE_CLIENT_SECRET` della funzione Lambda dell'origine dati. Per ulteriori informazioni sull'aggiornamento del segreto in AWS Secrets Manager, consulta Modificare un AWS Secrets Manager segreto.

Esecuzione di query sull'origine dati

- Quando si esegue una query di Azure Monitor, dopo aver selezionato l'origine dati nella scheda Query da più origini e aver selezionato un connettore Azure Monitor, si specifica la sottoscrizione di Azure, il gruppo di risorse e la risorsa. È quindi possibile selezionare lo spazio dei nomi del parametro, il parametro e l'aggregazione dei parametri e filtrare per dimensioni.

Prometheus

Creazione di un'origine dati

- È necessario fornire l'endpoint Prometheus e l'utente e la password necessari per interrogare Prometheus. Le credenziali verranno archiviate in AWS Secrets Manager.
- Se l'origine dati è accessibile solo in un VPC, è necessario includere la configurazione VPC per il connettore, come descritto in Connessione a un'origine dati predefinita con una procedura guidata. Se l'origine dati deve essere collegata per le credenziali, l'endpoint deve essere configurato nel VPC. Per ulteriori informazioni, consulta Utilizzo di un AWS Secrets Manager endpoint VPC.

Aggiornamento della configurazione dell'origine dati

- Puoi aggiornare l'origine dati manualmente con le seguenti operazioni:

- Per aggiornare l'endpoint Prometheus, specifica il nuovo endpoint come variabile di ambiente `PROMETHEUS_API_ENDPOINT` nella funzione Lambda dell'origine dati.
- Per aggiornare il nome utente e la password usati per connettersi a Prometheus, puoi trovare l'ARN del segreto usato per l'origine dati come variabile di ambiente `PROMETHEUS_API_SECRET` della funzione Lambda dell'origine dati. Per ulteriori informazioni sull'aggiornamento del segreto in AWS Secrets Manager, consulta Modificare un AWS Secrets Manager segreto.
- Per aggiornare la configurazione VPC, consulta Configurazione dell'accesso VPC (console) per ulteriori informazioni.

Esecuzione di query sull'origine dati

> **Important**
>
> I tipi di metriche di Prometheus sono CloudWatch diversi dalle metriche e molte metriche disponibili tramite Prometheus sono cumulative per definizione. Quando si esegue una query sulle metriche di Prometheus CloudWatch , non applica alcuna trasformazione aggiuntiva ai dati: se si specifica solo il nome o l'etichetta della metrica, il valore visualizzato sarà cumulativo. Per ulteriori informazioni, consulta Tipi di parametri nella documentazione di Prometheus.
>
> Per vedere i dati delle metriche di Prometheus come valori discreti, ad CloudWatch esempio metriche, devi modificare la query prima di eseguirla. Ad esempio, potresti dover aggiungere una chiamata alla funzione rate al nome del parametro Prometheus. Per la documentazione sulla funzione rate e altre funzioni di Prometheus, consulta rate() nella documentazione di Prometheus.

Le interrogazioni multilinea non sono supportate dai connettori delle sorgenti dati. CloudWatch Ogni avanzamento riga viene sostituito da uno spazio quando la query viene eseguita o quando si crea un allarme o un widget del pannello di controllo con la query. In alcuni casi, ciò potrebbe rendere la query non valida. Ad esempio, se la query contiene un commento di una singola riga, non sarà valida. Se provi a creare una dashboard o un allarme con una query su più righe dalla riga di comando o Infrastructure as Code, l'API rifiuta l'azione generando un errore di analisi.

Notifica degli aggiornamenti disponibili

Periodicamente, Amazon potrebbe inviarti una notifica per consigliarti di aggiornare i connettori con una versione disponibile più recente e ti fornirà istruzioni su come farlo.

Creare un connettore personalizzato a un'origine dati

Per connettere un'origine dati personalizzata a CloudWatch, hai due opzioni:

- Inizia utilizzando un modello di esempio che CloudWatch fornisce. Puoi usare entrambi JavaScript o Python con questo modello. Questi modelli includono codice Lambda di esempio che ti sarà utile durante la creazione della tua funzione Lambda. Puoi quindi modificare la funzione Lambda del modello per connetterti alla tua origine dati personalizzata.
- Crea una AWS Lambda funzione da zero che implementi il connettore di origine dati, la query dei dati e la preparazione delle serie temporali da utilizzare da. CloudWatch Questa funzione deve preaggregare o unire i punti dati, se necessario, e inoltre allineare il periodo e i timestamp con cui è compatibile. CloudWatch

Indice

Utilizzo dei modelli

L'utilizzo di un modello crea una funzione Lambda di esempio e può aiutarti a creare più velocemente il tuo connettore personalizzato. Queste funzioni di esempio forniscono codice di esempio per molti scenari comuni relativi alla creazione di un connettore personalizzato. Puoi esaminare il codice Lambda dopo aver creato un connettore con un modello, quindi modificarlo per utilizzarlo per connetterti alla tua origine dati.

Inoltre, se utilizzi il modello, CloudWatch si occupa di creare la politica di autorizzazioni Lambda e di allegare i tag delle risorse alla funzione Lambda.

Per usare il modello per creare un connettore a un'origine dati personalizzata

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nel pannello di navigazione scegli Impostazioni.
3. Seleziona la scheda Origini dati dei parametri.
4. Seleziona Create data source (Crea origine dati).
5. Seleziona il pulsante di opzione Personalizzato - modello di base, quindi scegli Successivo.
6. Inserisci un nome per l'origine dati.
7. Seleziona uno dei modelli elencati.
8. Seleziona Node.js o Python.
9. Seleziona Create data source (Crea origine dati).

   La nuova fonte personalizzata che hai appena aggiunto non viene visualizzata finché lo AWS CloudFormation stack non finisce di crearla. Per verificare i progressi, puoi scegliere Visualizza lo stato del mio CloudFormation stack. In alternativa, puoi selezionare l'icona di aggiornamento per aggiornare questo elenco.

   Quando la nuova origine dati viene visualizzata in questo elenco, è pronta per essere testata nella console e modificata.
10. (Facoltativo) Per eseguire query sui dati di test provenienti da questa origine nella console, consulta le istruzioni fornite in Creazione di un grafico dei parametri da un'altra origine dati.
11. Modifica la funzione Lambda in base alle necessità.

    a. Nel pannello di navigazione scegli Impostazioni.

    b. Seleziona la scheda Origini dati dei parametri.

    c. Seleziona Visualizza nella console Lambda per l'origine che desideri modificare.

    Ora puoi modificare la funzione per accedere alla tua origine dati. Per ulteriori informazioni, consulta Fase 1: Creare la funzione.

    > **Note**
    >
    > Utilizzando il modello, quando scrivi la funzione Lambda non è necessario seguire le istruzioni in Fase 2: creazione di una policy di autorizzazioni Lambda o. Fase 3:

collegamento di un tag delle risorse alla funzione Lambda Questi passaggi sono stati eseguiti CloudWatch perché hai utilizzato il modello.

Creazione di un'origine dati personalizzata partendo da zero

Segui i passaggi in questa sezione per creare una funzione Lambda che si connette CloudWatch a un'origine dati.

Fase 1: Creare la funzione

Un connettore di origine dati personalizzato deve supportare `GetMetricData` eventi provenienti da CloudWatch. Facoltativamente, puoi anche implementare un `DescribeGetMetricData` evento per fornire la documentazione agli utenti della CloudWatch console su come utilizzare il connettore. La `DescribeGetMetricData` risposta può essere utilizzata anche per impostare i valori predefiniti utilizzati nel generatore di query CloudWatch personalizzato.

CloudWatch fornisce frammenti di codice come esempi per aiutarti a iniziare. Per ulteriori informazioni, consultate l'archivio degli esempi all'indirizzo https://github.com/aws-samples/.cloudwatch-data-source-samples

Vincoli

- La risposta di Lambda deve avere una dimensione inferiore a 6 Mb. Se la risposta supera i 6 Mb, la risposta `GetMetricData` contrassegna la funzione Lambda come `InternalError` e non viene restituito alcun dato.
- La funzione Lambda deve completare la sua esecuzione entro 10 secondi per scopi di visualizzazione e dashboard o entro 4,5 secondi per l'utilizzo degli allarmi. Se la il tempo di esecuzione è più lungo, la risposta `GetMetricData` contrassegna la funzione Lambda come `InternalError` e non viene restituito alcun dato.
- La funzione Lambda deve inviare il suo output utilizzando timestamp epoch in secondi.
- Se la funzione Lambda non ricampiona i dati e restituisce invece dati che non corrispondono all'ora di inizio e alla durata del periodo richiesti dall' CloudWatch utente, tali dati vengono ignorati da. CloudWatch I dati aggiuntivi vengono eliminati da qualsiasi visualizzazione o allarme. Vengono eliminati anche tutti i dati che non si trovano tra l'ora di inizio e l'ora di fine.

  Ad esempio, se un utente richiede dati dalle 10:00 alle 11:00 con una durata di 5 minuti, gli intervalli di tempo validi per la restituzione dei dati sono "dalle 10:00:00 alle 10:04:59" e "dalle

10:05:00 alle 10:09:59". È necessario restituire una serie temporale che includa `10:00 value1`, `10:05 value2` e così via. Se la funzione restituisce `10:03 valueX`, ad esempio, viene eliminata perché 10:03 non corrisponde all'ora e al periodo di inizio richiesti.

- Le interrogazioni multilinea non sono supportate dai connettori delle sorgenti dati. CloudWatch Ogni avanzamento riga viene sostituito da uno spazio quando la query viene eseguita o quando si crea un allarme o un widget del pannello di controllo con la query. In alcuni casi, ciò potrebbe rendere la query non valida.

GetMetricData evento

Payload della richiesta

Di seguito è riportato un esempio di payload della richiesta `GetMetricData` inviato come input alla funzione Lambda.

```
{
  "EventType": "GetMetricData",
  "GetMetricDataRequest": {
    "StartTime": 1697060700,
    "EndTime": 1697061600,
    "Period": 300,
    "Arguments": ["serviceregistry_external_http_requests{host_cluster!=\"prod\"}"]
  }
}
```

- StartTime— Il timestamp che specifica i primi dati da restituire. Il Tipo è timestamp secondi epoch.
- EndTime— Il timestamp che specifica i dati più recenti da restituire. Tipo è timestamp secondi epoch.
- Periodo:: il numero di secondi rappresentato da ciascuna aggregazione dei dati dei parametri. Il valore minimo è 60 secondi. Il Type è Secondi.
- Argomenti: una matrice di argomenti da passare all'espressione matematica del parametro Lambda. Per informazioni sugli argomenti da passare, consulta Come passare argomenti alla funzione Lambda.

Payload della risposta

Di seguito è riportato un esempio di payload di risposta `GetMetricData` per la funzione Lambda.

```
{
   "MetricDataResults": [
      {
         "StatusCode": "Complete",
         "Label": "CPUUtilization",
         "Timestamps": [ 1697060700, 1697061000, 1697061300 ],
         "Values": [ 15000, 14000, 16000 ]
      }
   ]
}
```

Il payload di risposta conterrà un campo `MetricDataResults` o un campo `Error`, ma non entrambi.

Un campo `MetricDataResults` è un elenco di campi di serie temporali di tipo `MetricDataResult`. Ciascuno di questi campi di serie temporali può includere i seguenti campi.

- StatusCode— (Facoltativo) `Complete` indica che sono stati restituiti tutti i punti dati nell'intervallo di tempo richiesto. `PartialData`significa che è stato restituito un set incompleto di punti dati. Se viene omesso, il valore predefinito è `Complete`.

  Valori validi: `Complete` | `InternalError` | `PartialData` | `Forbidden`
- Messaggi: elenco facoltativo di messaggi con informazioni aggiuntive sui dati restituiti.

  Tipo: matrice di MessageDataoggetti con `Code` e `Value` stringhe.
- Etichetta: l'etichetta leggibile dall'uomo associata ai dati.

  ➖Tipo: stringa
- Timestamp: i timestamp per i punti dati, formattati in base all'ora epoch. Il numero di timestamp corrisponde sempre al numero di valori e il valore per `Timestamps[x]` è `Values[x]`.

  Tipo: matrice di timestamp
- Valori: i valori dei punti dati per il parametro, corrispondenti a `Timestamps`. Il numero di valori corrisponde sempre al numero di timestamp e il valore per `Timestamps[x]` è `Values[x]`.

  Tipo: matrice di doppi

Per ulteriori informazioni sull'utilizzo degli oggetti `Error`, consulta le sezioni successive.

Formati di risposta di errore

Puoi utilizzare la risposta di errore per fornire ulteriori informazioni sugli errori. Ti consigliamo di restituire un errore con Code Validation quando si verifica un errore di convalida, ad esempio quando un parametro manca o è del tipo sbagliato.

Di seguito è riportato un esempio di risposta per un caso in cui la funzione Lambda vuole generare un'eccezione di convalida `GetMetricData`.

```
{
   "Error": {
      "Code": "Validation",
      "Value": "Invalid Prometheus cluster"
   }
}
```

Di seguito è riportato un esempio di risposta per un caso in cui la funzione Lambda indica che non è in grado di restituire dati a causa di un problema di accesso. La risposta viene tradotta in una singola serie temporale con un codice di stato di `Forbidden`.

```
{
   "Error": {
      "Code": "Forbidden",
      "Value": "Unable to access ..."
   }
}
```

Di seguito è riportato l'esempio di un caso in cui la funzione Lambda genera un'eccezione `InternalError` generale, che viene tradotta in una singola serie temporale con un codice di stato di `InternalError` e un messaggio. Ogni volta che un codice di errore ha un valore diverso da `Validation` o`Forbidden`, CloudWatch si presume che si tratti di un errore interno generico.

```
{
   "Error": {
      "Code": "PrometheusClusterUnreachable",
      "Value": "Unable to communicate with the cluster"
   }
}
```

DescribeGetMetricData evento

Payload della richiesta

Di seguito è riportato un esempio di payload della richiesta `DescribeGetMetricData`.

```
{
  "EventType": "DescribeGetMetricData"
}
```

Payload della risposta

Di seguito è riportato un esempio di payload della risposta `DescribeGetMetricData`.

```
{
    "Description": "Data source connector",
    "ArgumentDefaults": [{
        Value: "default value"
     }]
}
```

- Descrizione: una descrizione di come utilizzare il connettore di origine dati. Questa descrizione verrà visualizzata nella CloudWatch console. Markdown è supportato.

  ▪Tipo: stringa
- ArgumentDefaults— La matrice opzionale di valori predefiniti degli argomenti utilizzati precompila il generatore di sorgenti dati personalizzato.

  Se `[{ Value: "default value 1"}, { Value: 10}]` viene restituito, il generatore di query nella CloudWatch console visualizza due input, il primo con «valore predefinito 1" e il secondo con 10.

  Se `ArgumentDefaults` non viene fornito, viene visualizzato un singolo input con il tipo predefonito impostato su. `String`

  Tipo: matrice di oggetti contenente Vaolre e Tipo.
- Errore: (facoltativo) è possibile includere un campo di errore in qualsiasi risposta. Puoi consultare degli esempi in GetMetricData evento.

Considerazioni importanti sugli allarmi CloudWatch

Se intendi utilizzare la fonte dati per impostare gli CloudWatch allarmi, dovresti configurarla in modo da riportare dati con timestamp ogni minuto. CloudWatch Per ulteriori informazioni e altre considerazioni sulla creazione di allarmi sui parametri provenienti da origini dati connesse, consulta Creazione di un allarme basato su un'origine dati connessa.

(Facoltativo) Da utilizzare AWS Secrets Manager per memorizzare le credenziali

Se la tua funzione Lambda deve utilizzare credenziali per accedere all'origine dati, ti consigliamo di utilizzare AWS Secrets Manager per archiviare queste credenziali invece di codificarle nella tua funzione Lambda. Per ulteriori informazioni sull'utilizzo AWS Secrets Manager con Lambda, consulta Usare AWS Secrets Manager i segreti nelle AWS Lambda funzioni.

(Facoltativo) Connessione a un'origine dati in un VPC

Se la tua origine dati si trova in un VPC gestito da Amazon Virtual Private Cloud, devi configurare la funzione Lambda per accedervi. Per ulteriori informazioni, consulta Connessione della rete in uscita alle risorse in un VPC.

Potrebbe anche essere necessario configurare gli endpoint del servizio VPC per accedere a servizi come AWS Secrets Manager. Per ulteriori informazioni, consulta Accedere a un AWS servizio utilizzando un endpoint VPC di interfaccia.

Fase 2: creazione di una policy di autorizzazioni Lambda

È necessario utilizzare la funzione di creazione di una dichiarazione politica che conceda l' CloudWatch autorizzazione all'uso della funzione Lambda creata. È possibile utilizzare la console AWS CLI o la console Lambda per creare l'informativa sulla politica.

Da utilizzare AWS CLI per creare l'informativa sulla politica

- Inserire il seguente comando. Sostituisci *123456789012* con l'ID del tuo account, sostituisci *my-data-source-function*con il nome della tua funzione Lambda e sostituisci *MyDataSource-DataSourcePermission 1234* con un valore univoco arbitrario.

```
aws lambda add-permission --function-name my-data-source-function --statement-
id MyDataSource-DataSourcePermission1234 --action lambda:InvokeFunction --principal
 lambda.datasource.cloudwatch.amazonaws.com --source-account 123456789012
```

Fase 3: collegamento di un tag delle risorse alla funzione Lambda

La CloudWatch console determina quali delle funzioni Lambda sono connettori di sorgenti dati utilizzando un tag. Quando crei un'origine dati utilizzando una delle procedure guidate, il tag viene applicato automaticamente dallo AWS CloudFormation stack che lo configura. Quando crei personalmente un'origine dati, puoi usare il tag seguente per la tua funzione Lambda. In questo modo il connettore viene visualizzato nel menu a discesa Origine dati della CloudWatch console quando esegui una query sulle metriche.

- Aggiungi un tag con `cloudwatch:datasource` come chiave e `custom` come valore.

Uso dell'origine dati personalizzata

Dopo aver creato un'origine dati, puoi usarla per eseguire query sui dati da tale origine per visualizzarli e impostare allarmi. Se hai utilizzato il modello per creare il connettore di origine dati personalizzato o hai aggiunto il tag indicato in Fase 3: collegamento di un tag delle risorse alla funzione Lambda, puoi seguire i passaggi indicati in Creazione di un grafico dei parametri da un'altra origine dati per eseguire query su di esso.

Puoi anche usare la funzione matematica dei parametri `LAMBDA` per eseguire query, come spiegato nella sezione seguente.

Per informazioni sulla creazione di allarmi su un parametro proveniente da un'origine dati, consulta Creazione di un allarme basato su un'origine dati connessa.

Come passare argomenti alla funzione Lambda

Il modo consigliato per passare argomenti alla tua fonte di dati personalizzata consiste nell'utilizzare il generatore di query nella CloudWatch console quando esegui una query sull'origine dati.

È inoltre possibile utilizzare la funzione Lambda per recuperare dati dalla fonte dati utilizzando la nuova `LAMBDA` espressione in CloudWatch matematica metrica.

```
LAMBDA("LambdaFunctionName" [, optional-arg]*)
```

`optional-arg` contiene fino a 20 stringhe, numeri o valori booleani. Ad esempio, `param`, `3.14` o `true`.

> **Note**
>
> Le stringhe multilinea non sono supportate dai connettori delle sorgenti dati. CloudWatch Ogni avanzamento riga viene sostituito da uno spazio quando la query viene eseguita o quando si crea un allarme o un widget del pannello di controllo con la query. In alcuni casi, ciò potrebbe rendere la query non valida.

Quando usi la funzione di matematica dei parametri `LAMBDA`, puoi fornire il nome della funzione (`"MyFunction"`). Se la policy delle risorse lo consente, puoi anche utilizzare una versione specifica della funzione (`"MyFunction:22"`) o un alias di funzione Lambda (`"MyFunction:MyAlias"`). Non è possibile utilizzare *

Di seguito sono riportati alcuni esempi di chiamata della funzione `LAMBDA`.

```
LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query")
```

```
LAMBDA("MyCustomDataSource", true, "fuzzy", 99.9)
```

La funzione matematica dei parametri `LAMBDA` restituisce un elenco di serie temporali che possono essere restituite al richiedente o combinate con altre funzioni matematiche dei parametri. Di seguito è riportato un esempio di combinazione di `LAMBDA` con altre funzioni matematiche dei parametri.

```
FILL(LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query"), 0)
```

Eliminazione di un connettore a un'origine dati

Per eliminare un connettore a un'origine dati, segui le istruzioni in questa sezione.

Per eliminare un connettore a un'origine dati

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Impostazioni.
3. Seleziona la scheda Origini dati dei parametri.
4. Scegli Gestisci CloudFormation nella riga dell'origine dati che desideri eliminare.

   Verrai indirizzato alla AWS CloudFormation console.

5. Nella sezione con il nome dell'origine dati, seleziona Elimina.
6. Nella finestra pop-up di conferma, scegli Elimina.

Raccogli metriche, log e tracce con l'agente CloudWatch

L' CloudWatch agente unificato consente di effettuare le seguenti operazioni:

- Raccogliere parametri interni a livello di sistema dalle istanze Amazon EC2 tra sistemi operativi. Oltre a quelli delle istanze EC2, i parametri possono includere quelli in-guest. Gli ulteriori parametri che puoi raccogliere sono elencati in Metriche raccolte dall'agente CloudWatch.
- Raccogliere parametri a livello di sistema dai server locali. Questi possono includere server in un ambiente ibrido e server non gestiti da AWS.
- Recuperare i parametri personalizzati dalle applicazioni o dai servizi con i protocolli `StatsD` e `collectd`. `StatsD` è supportato dai server Linux e da quelli in cui è in esecuzione Windows Server. `collectd` è supportato solo dai server Linux.
- Raccogliere log da istanze Amazon EC2 e server locali che eseguono Linux o Windows Server.

  > **Note**
  >
  > L' CloudWatch agente non supporta la raccolta di log dalle pipe FIFO.

- È possibile utilizzare la versione 1.300031.0 e successive per abilitare Application Signals. CloudWatch Per ulteriori informazioni, consulta Application Signals.
- La versione 1.300025.0 e successive possono raccogliere tracce dagli SDK dei nostri client OpenTelemetryX-Ray e inviarle a X-Ray.

  L'utilizzo dell' CloudWatch agente consente di raccogliere tracce senza dover eseguire un demone di raccolta delle tracce separato, contribuendo a ridurre il numero di agenti che vengono eseguiti e gestiti.

Puoi archiviare e visualizzare le metriche raccolte con l' CloudWatch agente CloudWatch proprio come con qualsiasi altra metrica. CloudWatch Lo spazio dei nomi predefinito per le metriche raccolte dall' CloudWatch agente è`CWAgent`, sebbene sia possibile specificare uno spazio dei nomi diverso quando si configura l'agente.

I log raccolti dall' CloudWatch agente unificato vengono elaborati e archiviati in Amazon CloudWatch Logs, proprio come i log raccolti dal precedente agente Logs. CloudWatch Per informazioni sui prezzi di CloudWatch Logs, consulta la pagina dei CloudWatch prezzi di Amazon.

Le metriche raccolte dall' CloudWatch agente vengono fatturate come metriche personalizzate. Per ulteriori informazioni sui prezzi delle CloudWatch metriche, consulta la pagina dei CloudWatchprezzi di Amazon.

L' CloudWatch agente è open source con licenza MIT ed è ospitato su. GitHub Se desideri creare, personalizzare o contribuire all' CloudWatch agente, consulta il GitHub repository per le istruzioni più recenti. Se ritieni di aver trovato un potenziale problema di sicurezza, non pubblicarlo su GitHub alcun forum pubblico. Segui invece le istruzioni in Segnalazione delle vulnerabilità o segui direttamente la AWS sicurezza delle e-mail.

I passaggi di questa sezione spiegano come installare l' CloudWatch agente unificato su istanze Amazon EC2 e server locali. Per ulteriori informazioni sui parametri che l' CloudWatch agente può raccogliere, consulta. Metriche raccolte dall'agente CloudWatch

Sistemi operativi supportati

L' CloudWatch agente è supportato sull'architettura x86-64 sui seguenti sistemi operativi. È inoltre supportato su tutti gli aggiornamenti delle versioni minori per ciascuna delle versioni principali qui riportate.

- Amazon Linux 2023
- Amazon Linux 2
- Versioni di Ubuntu Server 23.10, 22.04, 20.04, 18.04, 16.04 e 14.04
- CentOS versioni 9, 8 e 7
- Red Hat Enterprise Linux (RHEL) versioni 9, 8 e 7
- Versioni Debian 12, 11 e 10
- SUSE Linux Enterprise Server (SLES) versioni 15 e 12
- Versioni Oracle Linux 9, 8 e 7
- AlmaLinux versioni 9 e 8
- Rocky Linux versioni 9 e 8
- I seguenti computer macOS: istanze EC2 M1 Mac1 e computer che eseguono macOS 14 (Sonoma), macOS 13 (Ventura) e macOS 12 (Monterey)
- Versioni a 64 bit di Windows Server 2022, Windows Server 2019 e Windows Server 2016
- Windows 10 a 64 bit

L'agente è supportato sull'architettura ARM64 nei seguenti sistemi operativi. È inoltre supportato su tutti gli aggiornamenti delle versioni minori per ciascuna delle versioni principali qui riportate.

- Amazon Linux 2023
- Amazon Linux 2
- Versioni di Ubuntu Server 23.10, 22.04, 20.04, 18.04 e 16.04
- CentOS versioni 9 e 8
- Red Hat Enterprise Linux (RHEL) versioni 9, 8 e 7
- Versioni Debian 12, 11 e 10
- SUSE Linux Enterprise Server 15
- I seguenti computer macOS: macOS 14 (Sonoma), macOS 13 (Ventura) e macOS 12 (Monterey)

Panoramica del processo di installazione

Puoi scaricare e installare l' CloudWatch agente manualmente utilizzando la riga di comando oppure puoi integrarlo con SSM. Il flusso generale di installazione dell' CloudWatch agente utilizzando entrambi i metodi è il seguente:

1. Crea ruoli o utenti IAM che consentano all'agente di raccogliere metriche dal server e, facoltativamente, di integrarsi con. AWS Systems Manager
2. Download del pacchetto dell'agente.
3. Modifica il file di configurazione CloudWatch dell'agente e specifica le metriche che desideri raccogliere.
4. Installazione e avvio dell'agente sui server. Durante l'installazione dell'agente in un'istanza EC2, è possibile collegare il ruolo IAM creato nella fase 1. Durante l'installazione dell'agente in un server locale, specificare un profilo contenente le credenziali dell'utente IAM creato nella fase 1.

Indice

- Scenari comuni CloudWatch con l'agente
- Risoluzione dei problemi relativi all'agente CloudWatch

Installazione dell'agente CloudWatch

L' CloudWatch agente è disponibile come pacchetto in Amazon Linux 2023 e Amazon Linux 2. Se utilizzi uno di questi sistemi operativi, puoi installare il pacchetto inserendo il seguente comando. È inoltre necessario assicurarsi che il ruolo IAM associato all'istanza abbia l'CloudWatchAgentServerPolicyallegato. Per ulteriori informazioni, consulta la pagina Crea ruoli IAM da utilizzare con l' CloudWatch agente sulle istanze Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Su tutti i sistemi operativi supportati, inclusi Linux e Windows Server, puoi scaricare e installare l' CloudWatch agente utilizzando la riga di comando con un link per il download di Amazon S3, utilizzando Amazon EC2 Systems Manager o utilizzando un modello. AWS CloudFormation Per ulteriori informazioni, consulta le sezioni seguenti:

Indice

Installazione dell' CloudWatch agente tramite la riga di comando

Utilizza i seguenti argomenti per scaricare, configurare e installare il pacchetto dell'agente. CloudWatch

Argomenti

Scarica e configura l' CloudWatchagente utilizzando la riga di comando

Utilizza i seguenti passaggi per scaricare il pacchetto dell' CloudWatch agente, creare ruoli o utenti IAM e, facoltativamente, modificare il file di configurazione comune.

Scarica il pacchetto dell' CloudWatch agente

> **Note**
>
> Per scaricare l' CloudWatch agente, la connessione deve utilizzare TLS 1.2 o versione successiva.

L' CloudWatch agente è disponibile come pacchetto in Amazon Linux 2023 e Amazon Linux 2. Se utilizzi questo sistema operativo, è possibile installare il pacchetto immettendo il seguente comando. È inoltre necessario assicurarsi che il ruolo IAM associato all'istanza abbia il ruolo CloudWatchAgentServerPolicyallegato. Per ulteriori informazioni, consulta la pagina Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente.

```
sudo yum install amazon-cloudwatch-agent
```

Su tutti i sistemi operativi supportati, puoi scaricare e installare l' CloudWatch agente utilizzando la riga di comando.

Per ogni collegamento per il download è disponibile un collegamento generale e i collegamenti per ogni regione. Ad esempio, per Amazon Linux 2023 e Amazon Linux 2 e l'architettura x86-64, tre dei link di download validi sono:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Puoi anche scaricare un file README relativo alle modifiche più recenti all'agente e un file che indica il numero di versione disponibile per il download. Questi file si trovano nelle seguenti posizioni:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` o `https://amazoncloudwatch-agent-region.s3.region.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` o `https://amazoncloudwatch-agent-region.s3.region.amazonaws.com/info/latest/CWAGENT_VERSION`

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ .rpm amazon-cloudwatch-agent https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/amazon_linux/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/amazon_linux/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/centos/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/centos/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/redhat/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/redhat/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/suse/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/suse/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/debian/amd64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/debian/amd64/latest/ .deb.sig* amazon-cloudwatch-agent
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/ubuntu/amd64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/ubuntu/amd64/latest/ .deb.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/oracle_linux/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/oracle_linux/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/darwin/amd64/latest/ .pkg* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/darwin/amd64/latest/ .pkg.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/windows/amd64/latest/ .msi* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/windows/amd64/latest/ .msi.sig* amazon-cloudwatch-agent
ARM64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/amazon_linux/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/amazon_linux/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/redhat/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/redhat/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/ubuntu/arm64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/ubuntu/arm64/latest/ .deb.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/suse/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/suse/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent
ARM64	MacOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/darwin/arm64/latest/ .pkg* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/ amazon-cloudwatch-agent .pkg.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/darwin/arm64/latest/ .pkg.sig* amazon-cloudwatch-agent

Per utilizzare la riga di comando per scaricare e installare il pacchetto dell'agente CloudWatch

1. Scarica l' CloudWatch agente.

   Su un server Linux, immetti quanto segue. Per *download-link*, utilizza il collegamento per il download appropriato dalla tabella precedente.

   ```
   wget download-link
   ```

In un server che esegue Windows Server, scaricare il seguente file:

```
https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
```

2. Dopo aver scaricato il pacchetto, puoi eventualmente verificare la firma del pacchetto. Per ulteriori informazioni, consulta la pagina Verifica della firma del pacchetto dell'agente CloudWatch .
3. Installare il pacchetto . Se è stato scaricato un pacchetto RPM su un server Linux, passa alla directory contenente il pacchetto e immetti quanto segue:

```
sudo rpm -U ./amazon-cloudwatch-agent.rpm
```

Se è stato scaricato un pacchetto DEB su un server Linux, passa alla directory contenente il pacchetto e immetti quanto segue:

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Se è stato scaricato un pacchetto MSI su un server che esegue Windows Server, passa alla directory contenente il pacchetto e immetti quanto segue:

```
msiexec /i amazon-cloudwatch-agent.msi
```

Questo comando funziona anche dall'interno PowerShell. Per ulteriori informazioni sulle opzioni di comando MSI, consulta Command-Line Options nella documentazione di Microsoft Windows.

Se è stato scaricato un pacchetto PKG su un server macOS, passa alla directory contenente il pacchetto e immetti quanto segue:

```
sudo installer -pkg ./amazon-cloudwatch-agent.pkg -target /
```

Creazione e modifica del file di configurazione dell'agente

Dopo aver scaricato l' CloudWatch agente, è necessario creare il file di configurazione prima di avviare l'agente su qualsiasi server. Per ulteriori informazioni, consulta Creare il file di configurazione CloudWatch dell'agente.

Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente

L'accesso alle AWS risorse richiede autorizzazioni. Crei un ruolo IAM, un utente IAM o entrambi per concedere le autorizzazioni su cui l' CloudWatch agente deve scrivere le metriche. CloudWatch Se intendi usare l'agente nelle istanze Amazon EC2, dovrai creare un ruolo IAM. Se intendi usare l'agente nelle istanze dei server locali, dovrai creare un utente IAM.

> **Note**
>
> Abbiamo recentemente modificato le seguenti procedure utilizzando le nuove policy `CloudWatchAgentServerPolicy` e `CloudWatchAgentAdminPolicy` di Amazon, anziché richiedere ai clienti di creare tali policy. Per scrivere i file e per eseguire il download dei file da Parameter Store, le policy create da Amazon supportano solo i file con nomi che iniziano con `AmazonCloudWatch-`. Se disponi di un file di configurazione CloudWatch dell'agente con un nome di file che non inizia con`AmazonCloudWatch-`, queste policy non possono essere utilizzate per scrivere il file su Parameter Store o scaricarlo da Parameter Store.

Se intendi eseguire l' CloudWatch agente su istanze Amazon EC2, utilizza i seguenti passaggi per creare il ruolo IAM necessario. Questo ruolo fornisce le autorizzazioni per leggere le informazioni dall'istanza e scriverle su. CloudWatch

Per creare il ruolo IAM necessario per eseguire l' CloudWatch agente sulle istanze EC2

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, scegli Roles (Ruoli) e Create role (Crea ruolo).
3. Assicurati di aver selezionato AWS service (Servizio) in Trusted entity type (Tipo di entità attendibile).
4. Per Use case (Caso d'uso), seleziona EC2 in Common use cases (Casi d'uso comuni),
5. Seleziona Successivo.
6. Nell'elenco delle politiche, seleziona la casella di controllo accanto a CloudWatchAgentServerPolicy. Se necessario, utilizzare la casella di ricerca per trovare la policy.

7. (Facoltativo) Se l'agente invia tracce a X-Ray, è necessario assegnare al ruolo anche la AWSXRayDaemonWriteAccesspolicy. Per farlo, individua la policy nell'elenco e seleziona la relativa casella di controllo.
8. Seleziona Successivo.
9. In Nome ruolo, inserisci un nome per il ruolo, ad esempio *`CloudWatchAgentServerRole`*. Se desiderato, fornire una descrizione. Quindi seleziona Create role (Crea ruolo).

   Il ruolo è ora creato.
10. (Facoltativo) Se l'agente intende inviare log a CloudWatch Logs e desideri che sia in grado di impostare le politiche di conservazione per questi gruppi di log, devi aggiungere l'`logs:PutRetentionPolicy`autorizzazione al ruolo. Per ulteriori informazioni, consulta Consentire all' CloudWatch agente di impostare una politica di conservazione dei log.

Se intendi eseguire l' CloudWatch agente su server locali, utilizza i seguenti passaggi per creare l'utente IAM necessario.

> **Warning**
>
> Questo scenario richiede agli utenti IAM accesso programmatico e credenziali a lungo termine, il che presenta un rischio per la sicurezza. Per contribuire a mitigare questo rischio, ti consigliamo di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari. Le chiavi di accesso possono essere aggiornate se necessario. Per ulteriori informazioni, consulta Updating access keys nella IAM User Guide.

Per creare l'utente IAM necessario per l'esecuzione CloudWatch dell'agente sui server locali

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, scegli Users (Utenti), quindi Add users (Aggiungi utenti).
3. Immetti il nome del nuovo utente.
4. Seleziona Access key - Programmatic access (Chiave di accesso - Accesso programmatico) e scegli Next: Permissions (Successivo: Autorizzazioni).
5. Scegli Attach existing policies directly (Collega direttamente le policy esistenti).

6. Nell'elenco delle politiche, seleziona la casella di controllo accanto a CloudWatchAgentServerPolicy. Se necessario, utilizzare la casella di ricerca per trovare la policy.
7. (Facoltativo) Se l'agente ha intenzione di effettuare un tracciamento su X-Ray, è necessario assegnare al ruolo anche la AWSXRayDaemonWriteAccesspolicy. Per farlo, individua la policy nell'elenco e seleziona la relativa casella di controllo.
8. Scegli Successivo: Tag.
9. Facoltativamente, crea i tag per il nuovo utente IAM, quindi scegli Next: Review (Successivo: Rivedi).
10. Conferma che sia elencata la policy corretta, quindi scegli Create user (Crea utente).
11. Accanto al nome del nuovo utente, seleziona Show (Mostra). Copiare la chiave di accesso e la chiave segreta in un file, in modo da poterle utilizzare durante l'installazione dell'agente. Scegli Chiudi.

Consentire all' CloudWatch agente di impostare una politica di conservazione dei log

È possibile configurare l' CloudWatch agente per impostare la politica di conservazione per i gruppi di log a cui invia gli eventi di registro. Se esegui questa operazione, devi concedere l'autorizzazione `logs:PutRetentionPolicy` al ruolo o all'utente IAM utilizzato dall'agente. L'agente utilizza un ruolo IAM per l'esecuzione su istanze Amazon EC2 e un utente IAM per i server on-premise.

Per concedere al ruolo IAM dell' CloudWatch agente l'autorizzazione a impostare le politiche di conservazione dei log

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra, seleziona Ruoli.
3. Nella casella di ricerca, digita l'inizio del nome del ruolo IAM dell' CloudWatch agente. Hai scelto questo nome al momento della creazione del ruolo. Potrebbe essere denominato `CloudWatchAgentServerRole`.

   Quando viene visualizzato il nome del ruolo in questione, sceglilo.
4. Nella scheda Permissions (Autorizzazioni), scegli Add permissions (Aggiungi autorizzazioni), Create inline policy (Crea policy in linea).
5. Scegli la scheda JSON e copia la seguente policy nella casella, sostituendo il JSON predefinito:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:PutRetentionPolicy",
      "Resource": "*"
    }
  ]
}
```

6. Scegli Verifica policy.
7. Nel campo Name (Nome), inserisci **`CloudWatchAgentPutLogsRetention`** o un nome simile e scegli Create policy (Crea policy).

Per concedere all'utente IAM dell' CloudWatch agente l'autorizzazione a impostare le politiche di conservazione dei log

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Users (Utenti).
3. Nella casella di ricerca, digita l'inizio del nome dell'utente IAM dell' CloudWatch agente. Hai scelto questo nome al momento della creazione dell'utente.

   Quando vedi l'utente, scegli il suo nome.
4. Nella scheda Permissions (Autorizzazioni) scegli Add inline policy (Aggiungi policy inline).
5. Scegli la scheda JSON e copia la seguente policy nella casella, sostituendo il JSON predefinito:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "logs:PutRetentionPolicy",
      "Resource": "*"
    }
  ]
}
```

6. Scegli Verifica policy.
7. Nel campo Name (Nome), inserisci **`CloudWatchAgentPutLogsRetention`** o un nome simile e scegli Create policy (Crea policy).

Installazione ed esecuzione dell' CloudWatch agente sui tuoi server

Dopo avere creato il file di configurazione dell'agente desiderato e avere creato un ruolo IAM o un utente IAM, attieniti alla seguente procedura per installare ed eseguire l'agente sui server, utilizzando tale configurazione. Collega innanzitutto un ruolo IAM o un utente ruolo IAM al server che eseguirà l'agente. Quindi, in tale server, scarica il pacchetto dell'agente e avvialo con l'agente di configurazione creato.

Scarica il pacchetto dell' CloudWatch agente utilizzando un link per il download di S3

> **Note**
>
> Per scaricare l' CloudWatch agente, la connessione deve utilizzare TLS 1.2 o versione successiva.

È necessario installare l'agente su ciascun server in cui verrà eseguito l'agente.

AMI Amazon Linux

L' CloudWatch agente è disponibile come pacchetto in Amazon Linux 2023 e Amazon Linux 2. Se utilizzi questo sistema operativo, è possibile installare il pacchetto immettendo il seguente comando. È inoltre necessario assicurarsi che il ruolo IAM associato all'istanza abbia il ruolo CloudWatchAgentServerPolicyallegato. Per ulteriori informazioni, consulta la pagina Crea ruoli IAM da utilizzare con l' CloudWatch agente sulle istanze Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Tutti i sistemi operativi

Su tutti i sistemi operativi supportati, puoi scaricare e installare l' CloudWatch agente utilizzando la riga di comando con un link per il download di Amazon S3 come descritto nei passaggi seguenti.

Per ogni collegamento per il download è disponibile un collegamento generale e i collegamenti per ogni regione. Ad esempio, per Amazon Linux 2023 e Amazon Linux 2 e l'architettura x86-64, tre dei link di download validi sono:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ .rpm amazon-cloudwatch-agent https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/amazon_linux/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/amazon_linux/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaw*	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3.*

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
		s.com/centos/amd64/ latest/ .rpm amazon-cloudwatch-agent	*regione .amazonaws.com/centos/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/redhat/amd64/ latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/redhat/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/suse/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/suse/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/debian/amd64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/debian/amd64/latest/ .deb.sig* amazon-cloudwatch-agent
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/ubuntu/amd64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/ubuntu/amd64/latest/ .deb.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/oracle_linux/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/oracle_linux/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/darwin/amd64/latest/ .pkg* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/darwin/amd64/latest/ .pkg.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/windows/amd64/latest/ .msi* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/windows/amd64/latest/ .msi.sig* amazon-cloudwatch-agent
ARM64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/amazon_linux/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/amazon_linux/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/redhat/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/redhat/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/ubuntu/arm64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/ubuntu/arm64/latest/ .deb.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/suse/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/suse/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent

Per utilizzare la riga di comando per installare l' CloudWatch agente su un'istanza Amazon EC2

1. Scarica l' CloudWatch agente. In un server Linux, immetti quanto segue. Per *download-link*, utilizza il collegamento per il download appropriato dalla tabella precedente.

   ```
   wget download-link
   ```

   Per un server che esegue Windows Server, scarica il file seguente:

   ```
   https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
   ```

2. Dopo aver scaricato il pacchetto, puoi eventualmente verificare la firma del pacchetto. Per ulteriori informazioni, consulta la pagina Verifica della firma del pacchetto dell'agente CloudWatch .

3. Installare il pacchetto . Se è stato scaricato un pacchetto RPM su un server Linux, passa alla directory contenente il pacchetto e immetti quanto segue:

   ```
   sudo rpm -U ./amazon-cloudwatch-agent.rpm
   ```

   Se è stato scaricato un pacchetto DEB su un server Linux, passa alla directory contenente il pacchetto e immetti quanto segue:

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Se è stato scaricato un pacchetto MSI su un server che esegue Windows Server, passa alla directory contenente il pacchetto e immetti quanto segue:

```
msiexec /i amazon-cloudwatch-agent.msi
```

Questo comando funziona anche dall'interno PowerShell. Per ulteriori informazioni sulle opzioni di comando MSI, consulta Command-Line Options nella documentazione di Microsoft Windows.

(Installazione in un'istanza EC2) Collegamento di un ruolo IAM

Per consentire all' CloudWatch agente di inviare dati dall'istanza, devi assegnare un ruolo IAM all'istanza. Il ruolo da allegare è CloudWatchAgentServerRole. Avresti dovuto creare questo ruolo in precedenza. Per ulteriori informazioni, consulta Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente.

Per ulteriori informazioni sul collegamento di un ruolo IAM a un'istanza, consulta Collegamento di un ruolo IAM a un'istanza nella Guida per l'utente di Amazon EC2 per le istanze Windows.

(Installazione su un server locale) Specificare le credenziali e la regione IAM AWS

Per consentire all' CloudWatch agente di inviare dati da un server locale, devi specificare la chiave di accesso e la chiave segreta dell'utente IAM che hai creato in precedenza. Per ulteriori informazioni sulla creazione di questo utente, consulta Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente.

È inoltre necessario specificare la AWS regione a cui inviare le metriche, utilizzando il `region` campo nella `[AmazonCloudWatchAgent]` sezione del file di AWS configurazione, come nell'esempio seguente.

```
[profile AmazonCloudWatchAgent]
region = us-west-1
```

Di seguito è riportato un esempio di utilizzo del `aws configure` comando per creare un profilo denominato per l' CloudWatch agente. Questo esempio presuppone l'utilizzo del nome del profilo predefinito di `AmazonCloudWatchAgent`.

Per creare il AmazonCloudWatchAgent profilo per l' CloudWatch agente

1. Se non l'hai già fatto, installalo AWS Command Line Interface sul server. Per ulteriori informazioni, consulta l'argomento relativo all'installazione di AWS CLI.
2. Nei server Linux, immetti il seguente comando e segui le istruzioni:

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   In Windows Server, apri PowerShell come amministratore, immetti il comando seguente e segui le istruzioni.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

Verifica dell'accesso a Internet

Le istanze Amazon EC2 devono disporre dell'accesso a Internet in uscita per inviare dati ai nostri log. CloudWatch CloudWatch Per ulteriori informazioni su come configurare l'accesso a Internet, consulta Gateway Internet nella Guida per l'utente di Amazon VPC.

Gli endpoint e le porte per configurare il proxy sono i seguenti:

- Se utilizzi l'agente per raccogliere metriche, devi aggiungere gli CloudWatch endpoint per le regioni appropriate all'elenco delle regioni consentite. Questi endpoint sono elencati negli CloudWatch endpoint e nelle quote di Amazon.
- Se utilizzi l'agente per raccogliere i log, devi aggiungere gli endpoint Logs per le regioni CloudWatch appropriate all'elenco delle regioni consentite. Questi endpoint sono elencati negli endpoint e nelle quote di Amazon CloudWatch Logs.
- Se stai utilizzando System Manager per installare l'agente o Parameter Store per archiviare il file di configurazione, devi aggiungere gli endpoint Systems Manager per le regioni appropriate all'elenco consentiti. Questi endpoint sono riportati in Endpoint e quote di AWS Systems Manager.

(Opzionale) Modifica della configurazione comune delle informazioni relative al proxy o alla regione

L' CloudWatch agente include un file di configurazione chiamato. `common-config.toml` Se lo desideri, puoi utilizzare questo file per specificare le informazioni relative al proxy e alla regione.

Su un server che esegue Linux, questo file si trova nella directory `/opt/aws/amazon-cloudwatch-agent/etc`. Su un server che esegue Windows Server, questo file si trova nella directory `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

> **Note**
>
> Ti consigliamo di utilizzare il `common-config.toml` file per fornire configurazioni e credenziali condivise quando esegui l' CloudWatch agente in modalità locale e può essere utile anche quando lavori su Amazon EC2 e desideri riutilizzare i profili e i file di credenziali condivisi esistenti. L'attivazione tramite `common-config.toml` ha l'ulteriore vantaggio che se il file delle credenziali condivise viene ruotato con credenziali rinnovate dopo la scadenza, le nuove credenziali vengono raccolte automaticamente dall'agente senza richiedere un riavvio.

`common-config.toml` predefinito è il seguente.

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for the on-premises case by
 default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Inizialmente, tutte le righe sono commentate. Per impostare questo profilo di credenziali o le impostazioni del proxy, rimuovi # dalla riga e specifica un valore. È possibile modificare questo file manualmente oppure utilizzando il Run Command `RunShellScript` in Systems Manager:

- `shared_credential_profile`— Per i server locali, questa riga specifica il profilo delle credenziali utente IAM a cui inviare i dati. CloudWatch Se lasci il commento nella riga, verrà utilizzato `AmazonCloudWatchAgent`. Per ulteriori informazioni sulla creazione di questo profilo, consulta (Installazione su un server locale) Specificare le credenziali e la regione IAM AWS.

  Su un'istanza EC2, puoi utilizzare questa riga per fare in modo che l' CloudWatch agente invii i dati da questa istanza CloudWatch a un'altra regione. AWS A tale scopo, specifica un profilo che includa un campo `region` con il nome della regione di destinazione.

  Se specifichi un `shared_credential_profile`, occorre anche rimuovere # dall'inizio della riga `[credentials]`.
- `shared_credential_file`: per fare in modo che l'agente cerchi le credenziali in un file che si trova in un percorso diverso da quello predefinito, specificare qui il percorso completo e il nome del file. Il percorso predefinito è `/root/.aws` su Linux e `C:\\Users\\Administrator\\.aws` su Windows Server.

  Il primo esempio di seguito mostra la sintassi di una riga `shared_credential_file` valida per server Linux e il secondo esempio è valido per Windows Server. Su Windows Server, è necessario eseguire l'escape dei caratteri \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Se specifichi un `shared_credential_file`, occorre anche rimuovere # dall'inizio della riga `[credentials]`.
- Impostazioni proxy: se i server utilizzano proxy HTTP o HTTPS per contattare servizi AWS , specifica questi proxy nei campi `http_proxy` e `https_proxy`. Se sono presenti URL che devono essere escludi dal proxy, specificali nel campo `no_proxy`, separati da virgole.

Avvia l' CloudWatch agente utilizzando la riga di comando

Segui questi passaggi per utilizzare la riga di comando per avviare l' CloudWatch agente su un server.

Per utilizzare la riga di comando per avviare l' CloudWatch agente su un server

1. Copiare il file di configurazione dell'agente da utilizzare nel server in cui verrà eseguito l'agente. Annotare il percorso di destinazione della copia.
2. In questo comando, `-a fetch-config` fa sì che l'agente carichi la versione più recente del CloudWatch file di configurazione dell'agente e lo `-s` avvia.

   Inserisci uno dei comandi seguenti. Sostituisci *`configuration-file-path`*con il percorso del file di configurazione dell'agente. Questo file è chiamato `config.json` se è stato creato con la procedura guidata e potrebbe essere chiamato `amazon-cloudwatch-agent.json` se viene creato manualmente.

   In un'istanza EC2 con Linux in esecuzione, immetti il seguente comando.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:configuration-file-path
   ```

   In un server locale con Linux in esecuzione, immetti quanto segue:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m onPremise -s -c file:configuration-file-path
   ```

   Su un'istanza EC2 che esegue Windows Server, inserisci quanto segue dalla PowerShell console:

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
   a fetch-config -m ec2 -s -c file:configuration-file-path
   ```

   Su un server locale che esegue Windows Server, inserisci quanto segue dalla PowerShell console:

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
   a fetch-config -m onPremise -s -c file:configuration-file-path
   ```

Installazione dell' CloudWatch agente utilizzando AWS Systems Manager

Utilizza i seguenti argomenti per installare ed eseguire l'agente utilizzando. CloudWatch AWS Systems Manager

Argomenti

Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente

L'accesso alle AWS risorse richiede autorizzazioni. Puoi creare ruoli e utenti IAM che includono le autorizzazioni necessarie all' CloudWatch agente per scrivere metriche CloudWatch e all' CloudWatch agente per comunicare con Amazon EC2 e. AWS Systems Manager Puoi utilizzare ruoli IAM in istanze Amazon EC2 e utenti con ruolo IAM con server locali.

Un ruolo o utente consente l'installazione dell' CloudWatch agente su un server e l'invio di metriche a. CloudWatch L'altro ruolo o utente è necessario per archiviare la configurazione CloudWatch dell'agente in Systems Manager Parameter Store. Parameter Store consente a più server di utilizzare la configurazione di un solo CloudWatch agente.

La possibilità di scrivere in Parameter Store è un'autorizzazione estesa e potente. È consigliabile utilizzarla solo quando necessaria. Inoltre, non dovrebbe essere collegata a più istanze della distribuzione. Se memorizzi la configurazione CloudWatch dell'agente in Parameter Store, ti consigliamo quanto segue:

- Impostare un'istanza in cui si esegue questa configurazione.
- Utilizzare il ruolo IAM con le autorizzazioni di scrittura in Parameter Store solo su questa istanza.
- Utilizza il ruolo IAM con le autorizzazioni per scrivere su Parameter Store solo mentre lavori e salvi il file di configurazione dell' CloudWatch agente.

> **Note**
>
> Abbiamo recentemente modificato le seguenti procedure utilizzando le nuove policy `CloudWatchAgentServerPolicy` e `CloudWatchAgentAdminPolicy` di Amazon,

anziché richiedere ai clienti di creare tali policy. Per utilizzare queste policy per scrivere il file di configurazione dell'agente e quindi scaricarlo da Parameter Store, il file di configurazione dell'agente deve avere un nome che inizia con `AmazonCloudWatch-`. Se disponi di un file di configurazione CloudWatch dell'agente con un nome di file che non inizia con`AmazonCloudWatch-`, queste policy non possono essere utilizzate per scrivere il file su Parameter Store o per scaricare il file da Parameter Store.

Crea ruoli IAM da utilizzare con l' CloudWatch agente sulle istanze Amazon EC2

La prima procedura crea il ruolo IAM da collegare a ogni istanza Amazon EC2 che esegue l' CloudWatch agente. Questo ruolo fornisce le autorizzazioni per leggere le informazioni dall'istanza e scriverle su. CloudWatch

La seconda procedura crea il ruolo IAM da collegare all'istanza Amazon EC2 utilizzata per creare il file di configurazione dell' CloudWatch agente. Questa operazione è necessaria se si intende archiviare questo file in Systems Manager Parameter Store in modo che altri server possano utilizzarlo. Questo ruolo fornisce le autorizzazioni per la scrittura su Parameter Store, oltre alle autorizzazioni per leggere le informazioni dall'istanza e scriverle. CloudWatch Questo ruolo include le autorizzazioni sufficienti per eseguire l' CloudWatch agente e per scrivere su Parameter Store.

> **Note**
>
> Parameter Store supporta i parametri nei livelli Standard e Advanced. Questi livelli di parametri non sono correlati ai livelli di dettaglio Basic, Standard e Advanced disponibili con i set di metriche predefiniti dell' CloudWatch agente.

Per creare il ruolo IAM necessario a ciascun server per eseguire l'agente CloudWatch

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Ruoli, quindi Crea ruolo.
3. In Select type of trusted entity (Seleziona tipo di entità attendibile), scegli AWS service (Servizio).
4. Immediatamente sotto a Common use cases (Casi di utilizzo comuni), scegli EC2, quindi Next: Permissions (Successivo: Autorizzazioni).

5. Nell'elenco delle politiche, seleziona la casella di controllo accanto a CloudWatchAgentServerPolicy. Se necessario, utilizzare la casella di ricerca per trovare la policy.
6. Per utilizzare Systems Manager per installare o configurare l' CloudWatch agente, seleziona la casella accanto a ManagedInstanceCoreAmazonSSM. Questa policy AWS gestita consente a un'istanza di utilizzare le funzionalità principali del servizio Systems Manager. Se necessario, utilizzare la casella di ricerca per trovare la policy. Questa policy non è necessaria se avvii e configuri l'agente solo tramite la riga di comando.
7. Scegli Successivo: Tag.
8. (Facoltativo) Aggiungere una o più coppie tag-valore chiave per organizzare, monitorare o controllare l'accesso per questo ruolo, quindi scegli Next: Review (Successivo: Rivedi).
9. Per Role name (Nome ruolo), inserire un nome per il nuovo ruolo, ad esempio **`CloudWatchAgentServerRole`** o un altro nome che preferisci.
10. (Facoltativo) Per Role Description (Descrizione ruolo), immetti una descrizione.
11. Confermalo CloudWatchAgentServerPolicye, facoltativamente, AmazonSSM ManagedInstanceCore verrà visualizzato accanto a Policies.
12. Scegli Create role (Crea ruolo).

    Il ruolo è ora creato.

La procedura seguente crea il ruolo IAM che può anche scrivere in Parameter Store. Puoi usare questo ruolo per archiviare il file di configurazione dell'agente in Parameter Store in modo possa essere recuperato da altri server.

Le autorizzazioni per la scrittura in Parameter Store concedono un ampio accesso. Questo ruolo non deve essere collegato a tutti i server e deve essere utilizzato solo dagli amministratori. Una volta creato il file di configurazione dell'agente e copiato in Parameter Store, devi scollegare questo ruolo dall'istanza e utilizzare invece `CloudWatchAgentServerRole`.

Per creare il ruolo IAM per un amministratore per scrivere in Parameter Store

1. Accedi e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/. AWS Management Console
2. Nel riquadro di navigazione, scegli Ruoli, quindi Crea ruolo.
3. In Select type of trusted entity (Seleziona tipo di entità attendibile), scegli AWS service (Servizio).

4. Sotto l'opzione Choose the service that will use this role (Scegli il servizio che utilizzerà questo ruolo) scegli EC2, quindi Next: Permissions (Successivo: Autorizzazioni).
5. Nell'elenco delle politiche, seleziona la casella di controllo accanto a CloudWatchAgentAdminPolicy. Se necessario, utilizzare la casella di ricerca per trovare la policy.
6. Per utilizzare Systems Manager per installare o configurare l' CloudWatch agente, seleziona la casella accanto a ManagedInstanceCoreAmazonSSM. Questa policy AWS gestita consente a un'istanza di utilizzare le funzionalità principali del servizio Systems Manager. Se necessario, utilizzare la casella di ricerca per trovare la policy. Questa policy non è necessaria se avvii e configuri l'agente solo tramite la riga di comando.
7. Scegli Successivo: Tag.
8. (Facoltativo) Aggiungere una o più coppie tag-valore chiave per organizzare, monitorare o controllare l'accesso per questo ruolo, quindi scegli Next: Review (Successivo: Rivedi).
9. Per Role name (Nome ruolo), inserire un nome per il nuovo ruolo, ad esempio **`CloudWatchAgentAdminRole`** o un altro nome che preferisci.
10. (Facoltativo) Per Role Description (Descrizione ruolo), immetti una descrizione.
11. Confermalo CloudWatchAgentAdminPolicye, facoltativamente, AmazonSSM ManagedInstanceCore verrà visualizzato accanto a Policies.
12. Scegli Create role (Crea ruolo).

    Il ruolo è ora creato.

Crea utenti IAM da utilizzare con l'agente su server locali CloudWatch

La prima procedura crea l'utente IAM necessario per eseguire l' CloudWatch agente. Questo utente fornisce le autorizzazioni per inviare dati a CloudWatch.

La seconda procedura crea l'utente IAM che puoi utilizzare per creare il file di configurazione dell' CloudWatchagente. Utilizzare questa procedura per archiviare il file in Systems Manager Parameter Store in modo che possa essere utilizzato da altri server. Questo utente fornisce le autorizzazioni di scrittura su Parameter Store, oltre alle autorizzazioni per scrivere dati. CloudWatch

> **Note**
>
> Parameter Store supporta i parametri nei livelli Standard e Advanced. Questi livelli di parametri non sono correlati ai livelli di dettaglio Basic, Standard e Advanced disponibili con i set di metriche predefiniti dell' CloudWatch agente.

Per creare l'utente IAM necessario all' CloudWatch agente per scrivere i dati CloudWatch

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Users (Utenti), quindi scegli Add user (Aggiungi utente).
3. Immetti il nome del nuovo utente.
4. Per Access type (Tipo di accesso), seleziona Programmatic access (Accesso programmatico), quindi scegli Next: Permissions (Successivo: autorizzazioni).
5. Per Set permissions (Imposta autorizzazioni), scegli Attach existing policies directly (Collega direttamente le policy esistenti).
6. Nell'elenco delle politiche, seleziona la casella di controllo accanto a CloudWatchAgentServerPolicy. Se necessario, utilizzare la casella di ricerca per trovare la policy.
7. Per utilizzare Systems Manager per installare o configurare l' CloudWatch agente, seleziona la casella accanto a ManagedInstanceCoreAmazonSSM. Questa policy AWS gestita consente a un'istanza di utilizzare le funzionalità principali del servizio Systems Manager. (Se necessario, usa la casella di ricerca per trovare la policy. Questa policy non è necessaria se avvii e configuri l'agente solo tramite la riga di comando.)
8. Scegli Successivo: Tag.
9. (Facoltativo) Aggiungere una o più coppie tag-valore chiave per organizzare, monitorare o controllare l'accesso per questo ruolo, quindi scegli Next: Review (Successivo: Rivedi).
10. Verificare che siano elencate le policy corrette, quindi scegli Create user (Crea utente).
11. Nella riga per il nuovo utente, scegli Show (Mostra). Copiare la chiave di accesso e la chiave segreta in un file, in modo da poterle utilizzare durante l'installazione dell'agente. Scegli Chiudi.

La procedura seguente crea l'utente IAM che può anche scrivere in Parameter Store. Se intendi archiviare il file di configurazione dell'agente in Parameter Store in modo che possa essere utilizzato da altri server, devi usare questo utente IAM. Questo utente IAM fornisce le autorizzazioni per la scrittura in Parameter Store. Questo utente fornisce anche le autorizzazioni per leggere le informazioni dall'istanza e scriverle CloudWatch. Le autorizzazioni per la scrittura in Systems Manager Parameter Store concedono un ampio accesso. Questo utente IAM non deve essere collegato a tutti i server e deve essere utilizzato solo dagli amministratori. Devi utilizzare questo utente IAM solo quando archivi il file di configurazione dell'agente in Parameter Store.

Per creare l'utente IAM è necessario archiviare il file di configurazione in Parameter Store e inviare informazioni a CloudWatch

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Users (Utenti), quindi scegli Add user (Aggiungi utente).
3. Immetti il nome del nuovo utente.
4. Per Access type (Tipo di accesso), seleziona Programmatic access (Accesso programmatico), quindi scegli Next: Permissions (Successivo: autorizzazioni).
5. Per Set permissions (Imposta autorizzazioni), scegli Attach existing policies directly (Collega direttamente le policy esistenti).
6. Nell'elenco delle politiche, seleziona la casella di controllo accanto a CloudWatchAgentAdminPolicy. Se necessario, utilizzare la casella di ricerca per trovare la policy.
7. Per utilizzare Systems Manager per installare o configurare l' CloudWatch agente, seleziona la casella di controllo accanto a ManagedInstanceCoreAmazonSSM. Questa policy AWS gestita consente a un'istanza di utilizzare le funzionalità principali del servizio Systems Manager. (Se necessario, usa la casella di ricerca per trovare la policy. Questa policy non è necessaria se avvii e configuri l'agente solo tramite la riga di comando.)
8. Scegli Successivo: Tag.
9. (Facoltativo) Aggiungere una o più coppie tag-valore chiave per organizzare, monitorare o controllare l'accesso per questo ruolo, quindi scegli Next: Review (Successivo: Rivedi).
10. Verificare che siano elencate le policy corrette, quindi scegli Create user (Crea utente).
11. Nella riga per il nuovo utente, scegli Show (Mostra). Copiare la chiave di accesso e la chiave segreta in un file, in modo da poterle utilizzare durante l'installazione dell'agente. Scegli Chiudi.

Scarica e configura l' CloudWatch agente

Questa sezione spiega come utilizzare Systems Manager per scaricare l'agente e come creare il relativo file di configurazione. Prima di poter utilizzare Systems Manager per scaricare l'agente, devi assicurarti che l'istanza sia configurata correttamente per Systems Manager.

Installazione o aggiornamento di SSM Agent

Su un'istanza Amazon EC2, l' CloudWatch agente richiede che l'istanza esegua la versione 2.2.93.0 o successiva. Prima di installare l' CloudWatch agente, aggiorna o installa SSM Agent sull'istanza, se non l'hai già fatto.

Per informazioni sull'installazione o sull'aggiornamento di SSM Agent in un'istanza che esegue Linux, consulta la pagina relativa a installazione e configurazione di SSM Agent in Linux nella Guida per l'utente AWS Systems Manager .

Per ulteriori informazioni sull'installazione o l'aggiornamento di SSM Agent consulta Utilizzo di SSM Agent nella Guida per l'utente di AWS Systems Manager .

(Facoltativo) Verifica dei prerequisiti di Systems Manager

Verifica dell'accesso a Internet

Le istanze Amazon EC2 devono disporre dell'accesso a Internet in uscita per inviare dati ai nostri log. CloudWatch CloudWatch Per ulteriori informazioni su come configurare l'accesso a Internet, consulta Gateway Internet nella Guida per l'utente di Amazon VPC.

Gli endpoint e le porte per configurare il proxy sono i seguenti:

- Se utilizzi l'agente per raccogliere metriche, devi consentire l'elenco degli CloudWatch endpoint per le regioni appropriate. Questi endpoint sono elencati in Amazon CloudWatch nel Riferimenti generali di Amazon Web Services.
- Se utilizzi l'agente per raccogliere i log, devi consentire l'elenco degli endpoint CloudWatch Logs per le regioni appropriate. Questi endpoint sono elencati in Amazon CloudWatch Logs in. Riferimenti generali di Amazon Web Services
- Se stai utilizzando System Manager per installare l'agente o Parameter Store per archiviare il file di configurazione, è necessario consentire l'elenco degli endpoint Systems Manager per le regioni appropriate. Questi endpoint sono elencati in AWS Systems Manager nella Riferimenti generali di Amazon Web Services.

Utilizzare i passaggi seguenti per scaricare il pacchetto dell' CloudWatch agente utilizzando Systems Manager.

Per scaricare l' CloudWatch agente utilizzando Systems Manager

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione seleziona Run Command.

   oppure

Se la AWS Systems Manager home page si apre, scorri verso il basso e scegli Esplora Run Command.

3. Seleziona Run command (Esegui comando).
4. Nell'elenco del documento Comando, scegli AWS-Configure AWSPackage.
5. Nell'area Target, scegli l'istanza su cui installare l' CloudWatchagente. Se un'istanza specifica non viene visualizzata, è possibile che non sia configurata come istanza gestita per l'uso con Systems Manager. Per ulteriori informazioni, consulta Configurazione AWS Systems Manager per ambienti ibridi nella Guida per l'AWS Systems Manager utente.
6. Nell'elenco Action (Operazione), seleziona Install (Installa).
7. Nel campo Name (Nome), inserire *`AmazonCloudWatchAgent`*.
8. Lasciare Version (Versione) impostato su latest (più recente) per installare la versione più recente dell'agente.
9. Seleziona Esegui.
10. Facoltativamente, nelle aree Targets and outputs (Destinazioni e output), seleziona il pulsante accanto al nome dell'istanza, quindi seleziona View output. (Visualizza output). Systems Manager visualizzerà che l'agente è stato installato correttamente.

Creazione e modifica del file di configurazione dell'agente

Dopo aver scaricato l' CloudWatch agente, è necessario creare il file di configurazione prima di avviarlo su qualsiasi server.

Per salvare i file di configurazione dell'utente in Systems Manager Parameter Store, dovrai utilizzare un'istanza EC2 da salvare in Parameter Store. Inoltre, è necessario innanzitutto collegare a tale istanza il ruolo IAM `CloudWatchAgentAdminRole`. Per ulteriori informazioni sul collegamento di un ruolo, consulta Collegamento di un ruolo IAM a un'istanza nella Guida per l'utente di Amazon EC2 per le istanze Windows.

Per ulteriori informazioni sulla creazione del file di configurazione CloudWatch dell'agente, vedereCreare il file di configurazione CloudWatch dell'agente.

Installazione dell' CloudWatchagente su istanze EC2 utilizzando la configurazione dell'agente

Dopo aver salvato la configurazione CloudWatch dell'agente in Parameter Store, è possibile utilizzarla quando si installa l'agente su altri server.

Argomenti

Collegamento di un ruolo IAM all'istanza

È necessario collegare il ruolo CloudWatchAgentServerRoleIAM all'istanza EC2 per poter eseguire l' CloudWatch agente sull'istanza. Questo ruolo consente all' CloudWatch agente di eseguire azioni sull'istanza. Avresti dovuto creare questo ruolo in precedenza. Per ulteriori informazioni, consulta Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente.

Per ulteriori informazioni, consulta la sezione relativa al collegamento di un ruolo IAM a un'istanza nella Guida per l'utente di Amazon EC2 per le istanze Windows.

Scarica il pacchetto CloudWatch dell'agente su un'istanza Amazon EC2

È necessario installare l'agente su ciascun server in cui verrà eseguito l'agente. L' CloudWatch agente è disponibile come pacchetto in Amazon Linux 2023 e Amazon Linux 2. Se utilizzi questo sistema operativo, è possibile installare il pacchetto immettendo il seguente comando. È inoltre necessario assicurarsi che il ruolo IAM associato all'istanza abbia il ruolo CloudWatchAgentServerPolicyallegato. Per ulteriori informazioni, consulta la pagina Crea ruoli IAM da utilizzare con l' CloudWatch agente sulle istanze Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Su tutti i sistemi operativi supportati, puoi scaricare il pacchetto dell' CloudWatch agente utilizzando Systems Manager Run Command o un link per il download di Amazon S3. Per informazioni sull'uso di un collegamento per il download di Amazon S3, consulta Scarica il pacchetto dell' CloudWatch agente.

> **Note**
>
> Quando installi o aggiorni l' CloudWatch agente, è supportata solo l'opzione di disinstallazione e reinstallazione. Non è possibile utilizzare l'opzione Aggiornamento locale.

Scarica l' CloudWatch agente su un'istanza Amazon EC2 utilizzando Systems Manager

Prima di poter utilizzare Systems Manager per installare l' CloudWatch agente, è necessario assicurarsi che l'istanza sia configurata correttamente per Systems Manager.

Installazione o aggiornamento di SSM Agent

Su un'istanza Amazon EC2, l' CloudWatch agente richiede che l'istanza esegua la versione 2.2.93.0 o successiva. Prima di installare l' CloudWatch agente, aggiorna o installa SSM Agent sull'istanza, se non l'hai già fatto.

Per informazioni sull'installazione o sull'aggiornamento di SSM Agent in un'istanza che esegue Linux, consulta la pagina relativa a installazione e configurazione di SSM Agent in Linux nella Guida per l'utente AWS Systems Manager .

Per informazioni sull'installazione o sull'aggiornamento di SSM Agent in un'istanza che esegue Windows Server, consulta Installazione e configurazione di SSM Agent sulle istanze Windows nella Guida per l'utente AWS Systems Manager .

(Facoltativo) Verifica dei prerequisiti di Systems Manager

Prima di utilizzare Systems Manager Run Command per installare e configurare l' CloudWatch agente, verificate che le istanze soddisfino i requisiti minimi di Systems Manager. Per ulteriori informazioni, consulta Configurazione di AWS Systems Manager nella Guida per l'utente di AWS Systems Manager .

Verifica dell'accesso a Internet

Le istanze Amazon EC2 devono disporre di accesso a Internet in uscita per poter inviare dati ai nostri log. CloudWatch CloudWatch Per ulteriori informazioni su come configurare l'accesso a Internet, consulta Gateway Internet nella Guida per l'utente di Amazon VPC.

Scarica il pacchetto dell'agente CloudWatch

Run Command di Systems Manager ti permette di gestire la configurazione delle istanze. Puoi specificare un documento Systems Manager, specificare i parametri ed eseguire il comando in una o più istanze. SSM Agent sull'istanza elabora il comando e configura l'istanza come specificato.

Per scaricare l' CloudWatch agente utilizzando Run Command

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione seleziona Run Command.

   oppure

   Se la AWS Systems Manager home page si apre, scorri verso il basso e scegli Esplora Run Command.
3. Seleziona Run command (Esegui comando).
4. Nell'elenco del documento Comando, scegli AWS-Configure AWSPackage.
5. Nell'area Target, scegli l'istanza su cui installare l' CloudWatch agente. Se un'istanza specifica non viene visualizzata, potrebbe non essere configurata per Run Command. Per ulteriori informazioni, consulta Configurazione di AWS Systems Manager per ambienti ibridi nella Guida per l'utente AWS Systems Manager .
6. Nell'elenco Action (Operazione), seleziona Install (Installa).
7. Nella casella Name (Nome), inserisci *`AmazonCloudWatchAgent`*.
8. Lasciare Version (Versione) impostato su latest (più recente) per installare la versione più recente dell'agente.
9. Seleziona Esegui.
10. Facoltativamente, nelle aree Targets and outputs (Destinazioni e output), seleziona il pulsante accanto al nome dell'istanza, quindi seleziona View output. (Visualizza output). Systems Manager visualizzerà che l'agente è stato installato correttamente.

(Facoltativo) Modificate la configurazione comune e il profilo denominato per CloudWatch l'agente

L' CloudWatch agente include un file di configurazione chiamato`common-config.toml`. Se lo desideri, puoi utilizzare questo file per specificare le informazioni relative al proxy e alla regione.

Su un server che esegue Linux, questo file si trova nella directory `/opt/aws/amazon-cloudwatch-agent/etc`. Su un server che esegue Windows Server, questo file si trova nella directory `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

`common-config.toml` predefinito è il seguente:

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Inizialmente, tutte le righe sono commentate. Per impostare questo profilo di credenziali o le impostazioni del proxy, rimuovi # dalla riga e specifica un valore. Puoi modificare questo file manualmente oppure usando il Run Command `RunShellScript` in Systems Manager:

- `shared_credential_profile`— Per i server locali, questa riga specifica il profilo di credenziali utente IAM a cui inviare i dati. CloudWatch Se lasci il commento nella riga, verrà utilizzato `AmazonCloudWatchAgent`.

  Su un'istanza EC2, puoi utilizzare questa riga per fare in modo che l' CloudWatch agente invii i dati da questa istanza CloudWatch a un'altra regione. AWS A tale scopo, specifica un profilo che includa un campo `region` con il nome della regione di destinazione.

  Se specifichi un `shared_credential_profile`, occorre anche rimuovere # dall'inizio della riga `[credentials]`.

- `shared_credential_file`: per fare in modo che l'agente cerchi le credenziali in un file che si trova in un percorso diverso da quello predefinito, specificare qui il percorso completo e il nome del file. Il percorso predefinito è `/root/.aws` su Linux e `C:\\Users\\Administrator\\.aws` su Windows Server.

  Il primo esempio di seguito mostra la sintassi di una riga `shared_credential_file` valida per server Linux e il secondo esempio è valido per Windows Server. Su Windows Server, è necessario eseguire l'escape dei caratteri \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Se specifichi un `shared_credential_file`, occorre anche rimuovere # dall'inizio della riga `[credentials]`.
- Impostazioni proxy: se i server utilizzano proxy HTTP o HTTPS per contattare servizi AWS , specifica questi proxy nei campi `http_proxy` e `https_proxy`. Se sono presenti URL che devono essere escludi dal proxy, specificali nel campo `no_proxy`, separati da virgole.

Avvia l'agente CloudWatch

È possibile installare l'agente tramite Run Command in Systems Manager o tramite la riga di comando.

Avviare l' CloudWatch agente utilizzando Systems Manager Run Command

Procedi come segue per avviare l'agente tramite Run Command in Systems Manager.

Per avviare l' CloudWatch agente utilizzando Run Command

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione seleziona Run Command.

   oppure

   Se la AWS Systems Manager home page si apre, scorri verso il basso e scegli Esplora Run Command.

3. Seleziona Run command (Esegui comando).
4. Nell'elenco dei documenti Command, scegli AmazonCloudWatch- ManageAgent.
5. Nell'area Target, scegliete l'istanza in cui avete installato l' CloudWatch agente.
6. Nell'elenco Action (Operazione), seleziona configure (configura).
7. Nell'elenco Optional Configuration Source (Origine configurazione opzionale), seleziona ssm.
8. Nella casella Posizione configurazione facoltativa, immetti il nome del parametro di System Manager del file di configurazione dell'agente creato e salvato in Systems Manager Parameter Store, come descritto in Creare il file di configurazione CloudWatch dell'agente.
9. Nell'elenco Optional Restart (Riavvio opzionale), seleziona yes (sì) per avviare l'agente dopo aver completato questa procedura.
10. Seleziona Esegui.
11. Facoltativamente, nelle aree Targets and outputs (Destinazioni e output), seleziona il pulsante accanto al nome dell'istanza, quindi seleziona View output. (Visualizza output). Systems Manager visualizzerà che l'agente è stato avviato correttamente.

Avvia l' CloudWatch agente su un'istanza Amazon EC2 utilizzando la riga di comando

Segui questi passaggi per utilizzare la riga di comando per installare l' CloudWatch agente su un'istanza Amazon EC2.

Per utilizzare la riga di comando per avviare l' CloudWatch agente su un'istanza Amazon EC2

- In questo comando, `-a fetch-config` fa sì che l'agente carichi la versione più recente del file di configurazione dell' CloudWatch agente e lo `-s` avvia.

  Linux e macOS: se hai salvato il file di configurazione nel Parameter Store di Systems Manager, immetti quanto segue:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Linux e macOS: se hai salvato il file di configurazione nel computer locale, immetti il seguente comando. Sostituisci *configuration-file-path*con il percorso del file di configurazione dell'agente. Questo file è chiamato `config.json` se è stato creato con la procedura guidata e potrebbe essere chiamato `amazon-cloudwatch-agent.json` se viene creato manualmente.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Windows Server: se il file di configurazione dell'agente è stato salvato in Systems Manager Parameter Store, immettere quanto segue dalla PowerShell console:

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m ec2 -s -c ssm:configuration-parameter-store-name
```

Windows Server: se hai salvato il file di configurazione dell'agente sul computer locale, inserisci quanto segue dalla PowerShell console:

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\config.json"
```

Installazione dell' CloudWatch agente su server locali

Se l' CloudWatch agente è stato scaricato su un computer e creato il file di configurazione dell'agente desiderato, è possibile utilizzare tale file di configurazione per installare l'agente su altri server locali.

Scarica l' CloudWatch agente su un server locale

Puoi scaricare il pacchetto dell' CloudWatch agente utilizzando Systems Manager Run Command o un link per il download di Amazon S3. Per informazioni sull'uso di un collegamento per il download di Amazon S3, consulta Scarica il pacchetto dell' CloudWatch agente.

Download tramite Systems Manager

Per utilizzare Run Command in Systems Manager, è necessario registrare il server locale con Amazon EC2 Systems Manager. Per ulteriori informazioni, consulta la pagina relativa alla configurazione di Systems Manager in ambienti ibridi nella Guida per l'utente di AWS Systems Manager .

Se hai già registrato il server, aggiorna SSM Agent alla versione più recente.

Per ulteriori informazioni sull'aggiornamento di SSM Agent su un server che esegue Linux, consulta Installare SSM Agent per un ambiente ibrido (Linux) nella Guida per l'utente AWS Systems Manager .

Per ulteriori informazioni sull'aggiornamento di SSM Agent su un server che esegue Windows Server, consulta Installare SSM Agent per un ambiente ibrido (Windows) nella Guida per l'utente AWS Systems Manager .

Per utilizzare l'agente SSM per scaricare il pacchetto dell' CloudWatch agente su un server locale

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione seleziona Run Command.

   oppure

   Se la AWS Systems Manager home page si apre, scorri verso il basso e scegli Esplora Run Command.
3. Seleziona Run command (Esegui comando).
4. Nell'elenco del documento Comando, seleziona il pulsante accanto a AWSPackageAWS-Configure.
5. Nell'area Target, seleziona il server su cui installare l' CloudWatch agente. Se un server specifico non viene visualizzato, potrebbe non essere configurato per Run Command. Per ulteriori informazioni, consulta Configurazione di AWS Systems Manager per ambienti ibridi nella Guida per l'utente AWS Systems Manager .
6. Nell'elenco Action (Operazione), seleziona Install (Installa).
7. Nella casella Name (Nome), inserisci `AmazonCloudWatchAgent`.
8. Non compilare il campo Version (Versione) per installare la versione più recente dell'agente.
9. Seleziona Esegui.

   Il pacchetto dell'agente viene scaricato. Le fasi successive consistono in configurare e avviare l'agente.

(Installazione su un server locale) Specificare le credenziali e la regione IAM AWS

Per consentire all' CloudWatch agente di inviare dati da un server locale, devi specificare la chiave di accesso e la chiave segreta dell'utente IAM che hai creato in precedenza. Per ulteriori informazioni sulla creazione di questo utente, consulta Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente.

È inoltre necessario specificare la AWS regione a cui inviare le metriche, utilizzando il `region` campo.

Di seguito è riportato un esempio del file.

```
[AmazonCloudWatchAgent]
aws_access_key_id=my_access_key
aws_secret_access_key=my_secret_key
region = us-west-1
```

Per *`my_access_key`* e *`my_secret_key`*, utilizza le chiavi dell'utente IAM che non dispone delle autorizzazioni per la scrittura su Systems Manager Parameter Store. Per ulteriori informazioni sugli utenti IAM necessari per l' CloudWatch agente, consultaCrea utenti IAM da utilizzare con l'agente su server locali CloudWatch .

Se al profilo attribuisci il nome `AmazonCloudWatchAgent`, non dovrai effettuare altre operazioni. Eventualmente, puoi assegnare un nome diverso e specificarlo come valore di `shared_credential_profile` nel file `common-config.toml`, descritto nella sezione successiva.

Di seguito è riportato un esempio di utilizzo del aws configure comando per creare un profilo denominato per l' CloudWatch agente. Questo esempio presuppone l'utilizzo del nome del profilo predefinito di `AmazonCloudWatchAgent`.

Per creare il AmazonCloudWatchAgent profilo per l' CloudWatch agente

1. Se non l'hai già fatto, installalo AWS Command Line Interface sul server. Per ulteriori informazioni, consulta l'argomento relativo all'installazione di AWS CLI.
2. Nei server Linux, immetti il seguente comando e segui le istruzioni:

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   In Windows Server, apri PowerShell come amministratore, immetti il comando seguente e segui le istruzioni.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

(Facoltativo) Modifica della configurazione comune e del profilo denominato per l'agente CloudWatch

L' CloudWatch agente include un file di configurazione chiamato`common-config.toml`. Se lo desideri, puoi utilizzare questo file per specificare le informazioni relative al proxy e alla regione.

Su un server che esegue Linux, questo file si trova nella directory `/opt/aws/amazon-cloudwatch-agent/etc`. Su un server che esegue Windows Server, questo file si trova nella directory `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

`common-config.toml` predefinito è il seguente:

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Inizialmente, tutte le righe sono commentate. Per impostare questo profilo di credenziali o le impostazioni del proxy, rimuovi # dalla riga e specifica un valore. Puoi modificare questo file manualmente oppure usando il Run Command `RunShellScript` in Systems Manager:

- `shared_credential_profile`— Per i server locali, questa riga specifica il profilo di credenziali utente IAM a cui inviare i dati. CloudWatch Se lasci il commento nella riga, verrà utilizzato `AmazonCloudWatchAgent`. Per ulteriori informazioni sulla creazione di questo profilo, consulta (Installazione su un server locale) Specificare le credenziali e la regione IAM AWS.

Su un'istanza EC2, puoi utilizzare questa riga per fare in modo che l' CloudWatch agente invii i dati da questa istanza CloudWatch a un'altra regione. AWS A tale scopo, specifica un profilo che includa un campo `region` con il nome della regione di destinazione.

Se specifichi un `shared_credential_profile`, occorre anche rimuovere # dall'inizio della riga `[credentials]`.

- `shared_credential_file`: per fare in modo che l'agente cerchi le credenziali in un file che si trova in un percorso diverso da quello predefinito, specificare qui il percorso completo e il nome del file. Il percorso predefinito è `/root/.aws` su Linux e `C:\\Users\\Administrator\\.aws` su Windows Server.

  Il primo esempio di seguito mostra la sintassi di una riga `shared_credential_file` valida per server Linux e il secondo esempio è valido per Windows Server. Su Windows Server, è necessario eseguire l'escape dei caratteri \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Se specifichi un `shared_credential_file`, occorre anche rimuovere # dall'inizio della riga `[credentials]`.
- Impostazioni proxy: se i server utilizzano proxy HTTP o HTTPS per contattare servizi AWS , specifica questi proxy nei campi `http_proxy` e `https_proxy`. Se sono presenti URL che devono essere escludi dal proxy, specificali nel campo `no_proxy`, separati da virgole.

Avvio dell'agente di CloudWatch

È possibile avviare l' CloudWatch agente utilizzando Systems Manager Run Command o la riga di comando.

Per utilizzare SSM Agent per avviare l' CloudWatch agente su un server locale

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione seleziona Run Command.

   oppure

Se la AWS Systems Manager home page si apre, scorri verso il basso e scegli Esplora Run Command.

3. Seleziona Run command (Esegui comando).
4. Nell'elenco dei documenti Command, seleziona il pulsante accanto a AmazonCloudWatch-ManageAgent.
5. Nell'area Targets (Destinazioni), seleziona l'istanza su cui hai installato l'agente.
6. Nell'elenco Action (Operazione), seleziona configure (configura).
7. Nell'elenco Mode (Modalità), seleziona onPremise (Locale).
8. Nella casella Optional Configuration Location (Posizione configurazione facoltativa), immetti il nome del file di configurazione dell'agente creato con la procedura guidata e archiviato in Parameter Store.
9. Seleziona Esegui.

   L'agente viene avviato con la configurazione specificata nel file di configurazione.

Per utilizzare la riga di comando per avviare l' CloudWatch agente su un server locale

- In questo comando, `-a fetch-config` fa sì che l'agente carichi la versione più recente del file di configurazione dell' CloudWatch agente e lo `-s` avvia.

  Linux: se hai salvato il file di configurazione nel Parameter Store di Systems Manager, immetti quanto segue:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m onPremise -s -c ssm:configuration-parameter-store-name
  ```

  Linux: se hai salvato il file di configurazione nel computer locale, immetti il seguente comando. Sostituisci *configuration-file-path*con il percorso del file di configurazione dell'agente. Questo file è chiamato `config.json` se è stato creato con la procedura guidata e potrebbe essere chiamato `amazon-cloudwatch-agent.json` se viene creato manualmente.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m onPremise -s -c file:configuration-file-path
  ```

Windows Server: se il file di configurazione dell'agente è stato salvato in Systems Manager Parameter Store, immettere quanto segue dalla PowerShell console:

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c ssm:configuration-parameter-store-name
```

Windows Server: se hai salvato il file di configurazione dell'agente sul computer locale, inserisci quanto segue dalla PowerShell console. Sostituire *configuration-file-path*con il percorso del file di configurazione dell'agente. Questo file è chiamato `config.json` se è stato creato con la procedura guidata e potrebbe essere chiamato `amazon-cloudwatch-agent.json` se viene creato manualmente.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c file:configuration-file-path
```

Installazione dell' CloudWatchagente su nuove istanze utilizzando AWS CloudFormation

Amazon ha caricato diversi AWS CloudFormation modelli per aiutarti GitHub a installare e aggiornare l' CloudWatch agente su nuove istanze Amazon EC2. Per ulteriori informazioni sull'utilizzo AWS CloudFormation, consulta What is? AWS CloudFormation .

La posizione del modello è Deploy the Amazon CloudWatch agent to EC2 using. AWS CloudFormation Questo percorso include le directory `inline` e `ssm`. Ognuna di queste directory contiene modelli per le istanze Linux e Windows.

- I modelli nella `inline` directory hanno la configurazione dell' CloudWatch agente incorporata nel modello. AWS CloudFormation Per impostazione predefinita, i modelli di Linux raccolgono i parametri `mem_used_percent` e `swap_used_percent`, mentre i modelli di Windows raccolgono i parametri `Memory % Committed Bytes In Use` e `Paging File % Usage`.

  Per modificare questi modelli per raccogliere altri parametri, modifica la seguente sezione del modello. L'esempio seguente è preso dal modello per i server Linux. Segui il formato e la sintassi del file di configurazione dell'agente per apportare le modifiche. Per ulteriori informazioni, consulta la pagina Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.

```
{
   "metrics":{
      "append_dimensions":{
         "AutoScalingGroupName":"${!aws:AutoScalingGroupName}",
         "ImageId":"${!aws:ImageId}",
         "InstanceId":"${!aws:InstanceId}",
         "InstanceType":"${!aws:InstanceType}"
      },
      "metrics_collected":{
         "mem":{
            "measurement":[
               "mem_used_percent"
            ]
         },
         "swap":{
            "measurement":[
               "swap_used_percent"
            ]
         }
      }
   }
}
```

> **Note**
>
> Nei modelli inline, tutte le variabili segnaposto devono essere precedute da un punto esclamativo (!) come un carattere di escape. Ciò è visualizzato nel modello di esempio. Se aggiungi altre variabili segnaposto, assicurati di aggiungere un punto esclamativo prima del nome.

- I modelli nella directory `ssm` caricano un file di configurazione dell'agente da Parameter Store. Per utilizzare questi modelli, è necessario innanzitutto creare un file di configurazione e caricarlo in Parameter Store. Occorre quindi fornire il nome Parameter Store del file nel modello. Puoi creare il file di configurazione manualmente o utilizzando la procedura guidata. Per ulteriori informazioni, consulta Creare il file di configurazione CloudWatch dell'agente.

È possibile utilizzare entrambi i tipi di modelli per installare l' CloudWatch agente e aggiornare la configurazione dell'agente.

Tutorial: installa e configura l' CloudWatch agente utilizzando un AWS CloudFormation modello in linea

Questo tutorial illustra come AWS CloudFormation installare l' CloudWatch agente su una nuova istanza Amazon EC2. Questo tutorial esegue l'installazione in una nuova istanza che esegue Amazon Linux 2 utilizzando i modelli inline, che non richiedono l'uso del file di configurazione JSON o di Parameter Store. Il modello inline include la configurazione dell'agente nel modello. In questo tutorial viene utilizzata la configurazione dell'agente predefinita contenuta nel modello.

Dopo la procedura di installazione dell'agente, il tutorial prosegue con la procedura di aggiornamento dell'agente.

Da utilizzare AWS CloudFormation per installare l' CloudWatch agente su una nuova istanza

1. Scarica il modello da GitHub. In questo tutorial, scaricare il modello inline per Amazon Linux 2 come segue:

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/inline/amazon_linux.template
   ```

2. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
3. Seleziona Crea pila.
4. In Choose a template (Scegli un modello), seleziona Upload a template to Amazon S3 (Carica un modello in Amazon S3), scegli il modello scaricato e seleziona Next (Successivo).
5. Nella pagina Specify Details (Specifica dettagli), compili i parametri seguenti e scegli Next (Successivo):

   - Nome dello stack: scegli un nome per lo stack. AWS CloudFormation
   - IamRole: scegli un ruolo IAM che disponga delle autorizzazioni per scrivere CloudWatch metriche, log e tracce. Per ulteriori informazioni, consulta Crea ruoli IAM da utilizzare con l' CloudWatch agente sulle istanze Amazon EC2.
   - InstanceAMI: scegli un'AMI valida nella regione in cui verrà avviato lo stack.
   - InstanceType: Scegliete un tipo di istanza valido.
   - KeyName: per abilitare l'accesso SSH alla nuova istanza, scegli una coppia di key pair Amazon EC2 esistente. Se non disponi di una coppia di chiavi Amazon EC2, puoi crearla nella

AWS Management Console. Per ulteriori informazioni, consulta la sezione relativa alle coppie di chiavi Amazon EC2 nella Guida per l'utente di Amazon EC2 per le istanze Linux.

- SSHLocation: consente di specificare l'intervallo di indirizzi IP che è possibile utilizzare per connettersi all'istanza tramite SSH. L'impostazione predefinita consente l'accesso da qualsiasi indirizzo IP.

6. Nella pagina Options (Opzioni), puoi scegliere di aggiungere tag alle risorse dello stack. Seleziona Successivo.
7. Nella pagina Review (Verifica), esamina le informazioni, riconosci che lo stack può creare risorse &IAM;, quindi scegli Create (Crea).

   Se aggiorni la console, lo stato del nuovo stack è `CREATE_IN_PROGRESS`.
8. Quando viene creata, puoi visualizzare l'istanza nella console Amazon EC2. Se desiderato, è possibile eseguire la connessione all'host e verificare l'avanzamento.

   Utilizza il comando seguente per confermare che l'agente è installato:

   ```
   rpm -qa amazon-cloudwatch-agent
   ```

   Utilizza il comando seguente per confermare che l'agente è in esecuzione:

   ```
   ps aux | grep amazon-cloudwatch-agent
   ```

La procedura successiva illustra l'utilizzo AWS CloudFormation per aggiornare l' CloudWatch agente utilizzando un modello in linea. Il modello inline predefinito raccoglie il parametro `mem_used_percent`. In questo tutorial, puoi modificare la configurazione dell'agente per interrompere la raccolta di tale parametro.

Da utilizzare per AWS CloudFormation aggiornare l'agente CloudWatch

1. Nel modello scaricato nella procedura precedente, rimuovi le righe seguenti e quindi salva il modello:

   ```
   "mem": {

        "measurement": [
            "mem_used_percent"
   ```

```
        ]
 },
```

2. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
3. Nella AWS CloudFormation dashboard, seleziona lo stack che hai creato e scegli Update Stack.
4. In Select Template (Seleziona modello), seleziona Upload a template to Amazon S3 (Carica un modello in Amazon S3), scegli il modello modificato e seleziona Next (Successivo).
5. Nella pagina Options (Opzioni), scegli Next (Successivo), quindi Next (Successivo).
6. Nella pagina Review (Rivedi), esamina le informazioni e scegli Update (Aggiorna).

   Dopo qualche minuto, viene visualizzato `UPDATE_COMPLETE`.

Tutorial: installa l' CloudWatch agente utilizzando AWS CloudFormation e Parameter Store

Questo tutorial illustra come AWS CloudFormation installare l' CloudWatch agente su una nuova istanza Amazon EC2. Questo tutorial esegue l'installazione su una nuova istanza che esegue Amazon Linux 2 utilizzando un file di configurazione dell'agente creato e salvato in Parameter Store.

Dopo la procedura di installazione dell'agente, il tutorial prosegue con la procedura di aggiornamento dell'agente.

Da utilizzare AWS CloudFormation per installare l' CloudWatch agente su una nuova istanza utilizzando una configurazione di Parameter Store

1. Se non l'hai già fatto, scarica il pacchetto dell' CloudWatch agente su uno dei tuoi computer in modo da poter creare il file di configurazione dell'agente. Per ulteriori informazioni e per scaricare l'agente utilizzando Parameter Store, consulta Scarica e configura l' CloudWatch agente. Per ulteriori informazioni su come scaricare il pacchetto tramite la riga di comando, consulta Scarica e configura l' CloudWatchagente utilizzando la riga di comando.
2. Creare il file di configurazione dell'agente e salvarlo in Parameter Store. Per ulteriori informazioni, consulta Creare il file di configurazione CloudWatch dell'agente.
3. Scaricate il modello da GitHub quanto segue:

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/ssm/amazon_linux.template
   ```

4. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
5. Seleziona Crea stack.
6. In Choose a template (Scegli un modello), seleziona Upload a template to Amazon S3 (Carica un modello in Amazon S3), scegli il modello scaricato e seleziona Next (Successivo).
7. Nella pagina Specify Details (Specifica dettagli), compili di conseguenza i seguenti parametri e scegli Next (Successivo):

   - Nome dello stack: scegli un nome per lo stack. AWS CloudFormation
   - IamRole: scegli un ruolo IAM che disponga delle autorizzazioni per scrivere CloudWatch metriche, log e tracce. Per ulteriori informazioni, consulta Crea ruoli IAM da utilizzare con l' CloudWatch agente sulle istanze Amazon EC2.
   - InstanceAMI: scegli un'AMI valida nella regione in cui verrà avviato lo stack.
   - InstanceType: Scegli un tipo di istanza valido.
   - KeyName: per abilitare l'accesso SSH alla nuova istanza, scegli una coppia di key pair Amazon EC2 esistente. Se non disponi di una coppia di chiavi Amazon EC2, puoi crearla nella AWS Management Console. Per ulteriori informazioni, consulta la sezione relativa alle coppie di chiavi Amazon EC2 nella Guida per l'utente di Amazon EC2 per le istanze Linux.
   - SSHLocation: consente di specificare l'intervallo di indirizzi IP che è possibile utilizzare per connettersi all'istanza tramite SSH. L'impostazione predefinita consente l'accesso da qualsiasi indirizzo IP.
   - SSMKey: specifica il file di configurazione dell'agente creato e salvato in Parameter Store.
8. Nella pagina Options (Opzioni), puoi scegliere di aggiungere tag alle risorse dello stack. Seleziona Successivo.
9. Nella pagina Review (Verifica), esamina le informazioni, riconosci che lo stack può creare risorse &IAM;, quindi scegli Create (Crea).

   Se aggiorni la console, lo stato del nuovo stack è `CREATE_IN_PROGRESS`.
10. Quando viene creata, puoi visualizzare l'istanza nella console Amazon EC2. Se desiderato, è possibile eseguire la connessione all'host e verificare l'avanzamento.

    Utilizza il comando seguente per confermare che l'agente è installato:

```
rpm -qa amazon-cloudwatch-agent
```

Utilizza il comando seguente per confermare che l'agente è in esecuzione:

```
ps aux | grep amazon-cloudwatch-agent
```

La procedura successiva illustra l'utilizzo AWS CloudFormation per aggiornare l' CloudWatch agente, utilizzando una configurazione dell'agente salvata in Parameter Store.

Da utilizzare AWS CloudFormation per aggiornare l' CloudWatch agente utilizzando una configurazione in Parameter Store

1. Modificare il file di configurazione dell'agente archiviato in Parameter Store con la nuova configurazione desiderata.
2. Nel AWS CloudFormation modello scaricato nell'the section called “Tutorial: installa l' CloudWatch agente utilizzando AWS CloudFormation Parameter Store”argomento, modifica il numero di versione. Ad esempio, puoi modificare `VERSION=1.0` in `VERSION=2.0`.
3. Apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
4. Nella AWS CloudFormation dashboard, seleziona lo stack che hai creato e scegli Update Stack.
5. In Select Template (Seleziona modello), seleziona Upload a template to Amazon S3 (Carica un modello in Amazon S3), scegli il modello appena modificato e seleziona Next (Successivo).
6. Nella pagina Options (Opzioni), scegli Next (Successivo), quindi Next (Successivo).
7. Nella pagina Review (Rivedi), esamina le informazioni e scegli Update (Aggiorna).

   Dopo qualche minuto, viene visualizzato `UPDATE_COMPLETE`.

Risoluzione dei problemi di installazione dell'agente con CloudWatch AWS CloudFormation

Questa sezione consente di risolvere i problemi relativi all'installazione e all'aggiornamento dell'agente tramite. CloudWatch AWS CloudFormation

Rilevamento di aggiornamento non riuscito

Se si utilizza AWS CloudFormation per aggiornare la configurazione CloudWatch dell'agente e si utilizza una configurazione non valida, l'agente interrompe l'invio di qualsiasi metrica a. CloudWatch Per controllare rapidamente se un aggiornamento della configurazione dell'agente è andato a buon

fine, puoi esaminare il file `cfn-init-cmd.log`. Su un server Linux, il file si trova in `/var/log/cfn-init-cmd.log`. Su un'istanza Windows, il file si trova in `C:\cfn\log\cfn-init-cmd.log`.

Parametri mancanti

Se alcuni parametri previsti dopo l'installazione o l'aggiornamento dell'agente non sono visualizzati, verifica che l'agente sia configurato per raccogliere tale parametro. A questo scopo, controlla il file `amazon-cloudwatch-agent.json` per assicurarti che il parametro sia elencato e verifica di stare effettuando la ricerca nello spazio dei nomi parametro corretto. Per ulteriori informazioni, consulta CloudWatch file e posizioni degli agenti.

CloudWatch preferenza per le credenziali dell'agente

Questa sezione descrive la catena di fornitori di credenziali utilizzata dall' CloudWatch agente per ottenere le credenziali quando comunica con altri servizi e API. AWS L'ordine è il seguente. Le preferenze elencate nei numeri da due a cinque del seguente elenco hanno lo stesso ordine di preferenze definito nell' AWS SDK. Per ulteriori informazioni, consulta Specificazione delle credenziali nella documentazione SDK.

1. File di configurazione e credenziali condivisi come definiti nel file dell'agente. CloudWatch `common-config.toml` Per ulteriori informazioni, consulta (Opzionale) Modifica della configurazione comune delle informazioni relative al proxy o alla regione.
2. AWS Variabili di ambiente SDK

> **Important**
>
> In Linux, se si esegue l' CloudWatch agente utilizzando lo `amazon-cloudwatch-agent-ctl` script, lo script avvia l'agente come `systemd` servizio. In questo caso, le variabili di ambiente come `HOMEAWS_ACCESS_KEY_ID`, e non `AWS_SECRET_ACCESS_KEY` sono accessibili dall'agente.

3. File di configurazione e credenziali condivisi presenti in `$HOME/%USERPROFILE%`

> **Note**
>
> L' CloudWatch agente `$HOME` cerca `.aws/credentials` Linux e macOS e cerca Windows. `%USERPROFILE%` A differenza dell' AWS SDK, l' CloudWatch agente non dispone di metodi di fallback per determinare la home directory se le variabili di

ambiente sono inaccessibili. Questa differenza di comportamento serve a mantenere la retrocompatibilità con le implementazioni precedenti dell'SDK. AWS
Inoltre, a differenza delle credenziali condivise presenti in`common-config.toml`, se le credenziali condivise AWS derivate dall'SDK scadono e vengono ruotate, le credenziali rinnovate non vengono raccolte automaticamente dall' CloudWatch agente e richiedono il riavvio dell'agente per farlo.

4. Un AWS Identity and Access Management ruolo per le attività se è presente un'applicazione che utilizza una definizione di attività di Amazon Elastic Container Service o un'operazione RunTask API.
5. Collegare un profilo dell'istanza a un'istanza Amazon EC2

Come best practice, ti consigliamo di specificare le credenziali nel seguente ordine quando usi l' CloudWatch agente.

1. Usa i ruoli IAM per le attività se la tua applicazione utilizza una definizione di attività di Amazon Elastic Container Service o un'operazione RunTask API.
2. Usa i ruoli IAM se la tua applicazione viene eseguita su un'istanza Amazon EC2.
3. Usa il file dell' CloudWatch agente per specificare il `common-config.toml` file delle credenziali. Questo file di credenziali è lo stesso utilizzato da altri AWS SDK e da. AWS CLI Se stai già utilizzando un file di credenziali condiviso, puoi utilizzarlo anche per questo scopo. Se lo fornite utilizzando il `common-config.toml` file dell' CloudWatch agente, vi assicurate che l'agente utilizzi le credenziali ruotate quando scadono e vengano sostituite senza che sia necessario riavviare l'agente.
4. Utilizzate variabili di ambiente. L'impostazione delle variabili di ambiente è utile se stai eseguendo lavori di sviluppo su un computer diverso da un'istanza Amazon EC2.

> **Note**
>
> Se invii telemetria a un altro account come spiegato inInvio di parametri, log e tracce a un altro account, l' CloudWatch agente utilizza la catena di fornitori di credenziali descritta in questa sezione per ottenere il set iniziale di credenziali. Utilizza quindi tali credenziali quando assume il ruolo IAM specificato da nel file di configurazione dell'agente. `role_arn` CloudWatch

Verifica della firma del pacchetto dell'agente CloudWatch

I file di firma GPG sono inclusi per i pacchetti di CloudWatch agenti sui server Linux. Puoi utilizzare una chiave pubblica per verificare che il file di download dell'agente sia originale e non modificato.

Per Windows Server, puoi utilizzare l'MSI per verificare la firma.

Per i computer macOS, la firma è inclusa nel pacchetto di download dell'agente.

Per trovare il file della firma corretto, consulta la tabella seguente: Per ogni architettura e sistema operativo vengono forniti un collegamento generale e collegamenti per ogni regione. Ad esempio, per Amazon Linux 2023 e Amazon Linux 2 e l'architettura x86-64, tre dei link validi sono:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

> **Note**
>
> Per scaricare l' CloudWatch agente, la connessione deve utilizzare TLS 1.2 o versione successiva.

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig
		https://amazoncloudwatch-agent - *regione* .s3.	*https://amazoncloudwatch-agent*

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
		regione .amazonaws.com/amazon_linux/amd64/latest/ .rpm amazon-cloudwatch-agent	*- regione .s3. regione .amazonaws.com/amazon_linux/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/centos/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/centos/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/redhat/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/redhat/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/suse/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/suse/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/debian/amd64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/debian/amd64/latest/ .deb.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/ubuntu/amd64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/ubuntu/amd64/latest/ .deb.sig* amazon-cloudwatch-agent
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/oracle_linux/amd64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/oracle_linux/amd64/latest/ .rpm.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/darwin/amd64/latest/ .pkg* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/darwin/amd64/latest/ .pkg.sig* amazon-cloudwatch-agent
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/windows/amd64/latest/ .msi* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/windows/amd64/latest/ .msi.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
ARM64	Amazon Linux 2023 e Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/amazon_linux/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/amazon_linux/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/redhat/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/redhat/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent

Architettura	Piattaforma	Collegamento per il download	Collegamento al file di firma
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/ubuntu/arm64/latest/ .deb* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/ubuntu/arm64/latest/ .deb.sig* amazon-cloudwatch-agent
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *regione* .s3. *regione .amazonaws.com/suse/arm64/latest/ .rpm* amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm.sig *https://amazoncloudwatch-agent - regione .s3. regione .amazonaws.com/suse/arm64/latest/ .rpm.sig* amazon-cloudwatch-agent

Per verificare il pacchetto dell'agente su un server Linux CloudWatch

1. Scarica la chiave pubblica.

```
shell$ wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-cloudwatch-agent.gpg
```

2. Importa la chiave pubblica nel tuo keyring.

```
shell$  gpg --import amazon-cloudwatch-agent.gpg
```

```
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

Prendi nota del valore della chiave poiché sarà necessario nella fase successiva. Nell'esempio precedente, il valore della chiave è 3B789C72.

3. Verifica l'impronta eseguendo il comando seguente, sostituendo *key-value* con il valore annotato nella fase precedente:

```
shell$  gpg --fingerprint key-value
pub   2048R/3B789C72 2017-11-14
      Key fingerprint = 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid                  Amazon CloudWatch Agent
```

La stringa dell'impronta deve essere uguale alla seguente:

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

Se la stringa dell'impronta non corrisponde, non installare l'agente. Contatta Amazon Web Services.

Dopo aver verificato l'impronta digitale, è possibile utilizzarla per verificare la firma del pacchetto dell' CloudWatch agente.

4. Scarica il file della firma del pacchetto utilizzando wget. Per determinare il file della firma corretto, consulta la tabella precedente.

```
wget Signature File Link
```

5. Per verificare la firma, esegui gpg --verify.

```
shell$ gpg --verify signature-filename agent-download-filename
gpg: Signature made Wed 29 Nov 2017 03:00:59 PM PST using RSA key ID 3B789C72
gpg: Good signature from "Amazon CloudWatch Agent"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Se l'output include la frase `BAD signature`, controllare di avere eseguito la procedura correttamente. Se continui a ottenere questa risposta, contatta Amazon Web Services ed evita di utilizzare il file scaricato.

Prendi nota dell'avviso sulla trust. Una chiave è considerata attendibile solo se è stata firmata dall'utente o da un firmatario fidato. Questo non significa che la firma non sia valida, ma soltanto che la chiave pubblica non è stata verificata.

Per verificare il pacchetto CloudWatch dell'agente su un server che esegue Windows Server

1. Scarica e installa GnuPG per Windows da https://gnupg.org/download/. Durante l'installazione, includete l'opzione Shell Extension (GpgEx).

   È possibile eseguire i passaggi rimanenti in Windows PowerShell.

2. Scarica la chiave pubblica.

```
PS> wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-cloudwatch-agent.gpg -OutFile amazon-cloudwatch-agent.gpg
```

3. Importa la chiave pubblica nel tuo keyring.

```
PS>  gpg --import amazon-cloudwatch-agent.gpg
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

   Prendere nota del valore della chiave poiché sarà necessario nella fase successiva. Nell'esempio precedente, il valore della chiave è `3B789C72`.

4. Verifica l'impronta eseguendo il comando seguente, sostituendo *`key-value`* con il valore annotato nella fase precedente:

```
PS>  gpg --fingerprint key-value
pub   rsa2048 2017-11-14 [SC]
      9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid           [ unknown] Amazon CloudWatch Agent
```

   La stringa dell'impronta deve essere uguale alla seguente:

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

Se la stringa dell'impronta non corrisponde, non installare l'agente. Contatta Amazon Web Services.

Dopo aver verificato l'impronta digitale, è possibile utilizzarla per verificare la firma del pacchetto dell' CloudWatch agente.

5. Scarica il file della firma del pacchetto tramite wget. Per determinare il file di firma corretto, consulta CloudWatch Agent Download Links.
6. Per verificare la firma, esegui gpg --verify.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Se l'output include la frase `BAD signature`, controllare di avere eseguito la procedura correttamente. Se continui a ottenere questa risposta, contatta Amazon Web Services ed evita di utilizzare il file scaricato.

Prendi nota dell'avviso sulla trust. Una chiave è considerata attendibile solo se è stata firmata dall'utente o da un firmatario fidato. Questo non significa che la firma non sia valida, ma soltanto che la chiave pubblica non è stata verificata.

Per verificare il pacchetto CloudWatch dell'agente su un computer macOS

- Sono disponibili due metodi per la verifica della firma in macOS.
  - Verifica l'impronta digitale eseguendo il seguente comando:

    ```
    pkgutil --check-signature amazon-cloudwatch-agent.pkg
    ```

    Viene visualizzato un risultato simile a quello seguente.

    ```
    Package "amazon-cloudwatch-agent.pkg":
    ```

```
       Status: signed by a developer certificate issued by Apple for
distribution
       Signed with a trusted timestamp on: 2020-10-02 18:13:24 +0000
       Certificate Chain:
       1. Developer ID Installer: AMZN Mobile LLC (94KV3E626L)
       Expires: 2024-10-18 22:31:30 +0000
       SHA256 Fingerprint:
       81 B4 6F AF 1C CA E1 E8 3C 6F FB 9E 52 5E 84 02 6E 7F 17 21 8E FB
       0C 40 79 13 66 8D 9F 1F 10 1C

------------------------------------------------------------------------
       2. Developer ID Certification Authority
       Expires: 2027-02-01 22:12:15 +0000
       SHA256 Fingerprint:
       7A FC 9D 01 A6 2F 03 A2 DE 96 37 93 6D 4A FE 68 09 0D 2D E1 8D 03
       F2 9C 88 CF B0 B1 BA 63 58 7F

------------------------------------------------------------------------
       3. Apple Root CA
       Expires: 2035-02-09 21:40:36 +0000
       SHA256 Fingerprint:
       B0 B1 73 0E CB C7 FF 45 05 14 2C 49 F1 29 5E 6E DA 6B CA ED 7E 2C
       68 C5 BE 91 B5 A1 10 01 F0 24
```

- In alternativa, scarica e usa il file .sig. Per utilizzare questo metodo, procedi come segue.
  - Installa l'applicazione GPG sull'host macOS inserendo il seguente comando.

    ```
    brew install GnuPG
    ```

- Scarica il file della firma del pacchetto utilizzando curl. Per determinare il file di firma corretto, consulta CloudWatch Agent Download Links.
- Per verificare la firma, esegui gpg --verify.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Se l'output include la frase `BAD signature`, controllare di avere eseguito la procedura correttamente. Se continui a ottenere questa risposta, contatta Amazon Web Services ed evita di utilizzare il file scaricato.

Prendi nota dell'avviso sulla trust. Una chiave è considerata attendibile solo se è stata firmata dall'utente o da un firmatario fidato. Questo non significa che la firma non sia valida, ma soltanto che la chiave pubblica non è stata verificata.

Creare il file di configurazione CloudWatch dell'agente

Prima di eseguire l' CloudWatch agente su qualsiasi server, è necessario creare uno o più file di configurazione CloudWatch dell'agente.

Il file di configurazione dell'agente è un file JSON che specifica i parametri, i log e le tracce che l'agente deve raccogliere, inclusi i parametri personalizzati. Puoi crearlo utilizzando la procedura guidata oppure partendo da zero. Puoi anche utilizzare la procedura guidata per creare inizialmente il file di configurazione e quindi modificarlo manualmente. Se decidi di creare o modificare manualmente il file, la procedura risulterà più complessa. Tuttavia, avrai un maggiore controllo sui parametri raccolti e potrai specificare i parametri non disponibili nella procedura guidata.

Ogni volta che modifichi il file di configurazione dell'agente, è necessario riavviare l'agente affinché le modifiche diventino effettive. Per riavviare l'agente, segui le istruzioni descritte in Avvia l'agente CloudWatch .

Dopo avere creato un file di configurazione, puoi salvarlo manualmente come file JSON e utilizzarlo al momento dell'installazione dell'agente nel server. In alternativa, puoi memorizzarlo in Systems Manager Parameter Store se hai intenzione di utilizzare Systems Manager al momento dell'installazione dell'agente nei server.

L' CloudWatch agente supporta l'utilizzo di più file di configurazione. Per ulteriori informazioni, consulta File di configurazione di più CloudWatch agenti.

Le metriche, i log e le tracce raccolti dall' CloudWatch agente comportano costi. Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Indice

Crea il file di configurazione CloudWatch dell'agente con la procedura guidata

La procedura guidata per i file di configurazione dell'agente pone una serie di domande per aiutarvi a configurare l' CloudWatch agente in base alle vostre esigenze. `amazon-cloudwatch-agent-config-wizard`

Credenziali richieste

La procedura guidata può rilevare automaticamente le credenziali e la AWS regione da utilizzare se le AWS credenziali e i file di configurazione sono presenti prima di avviare la procedura guidata. Per ulteriori informazioni su questi file, consulta la pagina relativa ai file di configurazione e delle credenziali nella Guida per l'utente AWS Systems Manager .

Nel file delle AWS credenziali, la procedura guidata verifica le credenziali predefinite e cerca anche una sezione come la seguente: `AmazonCloudWatchAgent`

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

La procedura guidata visualizza le credenziali predefinite, quelle di `AmazonCloudWatchAgent` e un'opzione `Others`. Puoi scegliere il tipo di credenziali da utilizzare. Se scegli `Others`, puoi immettere le credenziali.

Per *`my_access_key`* e *`my_secret_key`*, utilizza le chiavi dell'utente IAM che dispone delle autorizzazioni per la scrittura su Systems Manager Parameter Store. Per ulteriori informazioni sugli utenti IAM necessari per l' CloudWatch agente, consulta. Crea utenti IAM da utilizzare con l'agente su server locali CloudWatch

Nel file di AWS configurazione, puoi specificare la regione a cui l'agente invia le metriche se è diversa dalla `[default]` sezione. L'impostazione predefinita prevede la pubblicazione dei parametri nella regione in cui si trova l'istanza Amazon EC2. Se intendi pubblicare i parametri in una regione differente, specifica qui la regione. Nell'esempio seguente, i parametri vengono pubblicati nella regione `us-west-1`.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

Esegui la procedura guidata di configurazione CloudWatch dell'agente

Per creare il file di configurazione CloudWatch dell'agente

1. Avvia la procedura guidata di configurazione dell' CloudWatch agente inserendo quanto segue:

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
   ```

   Su un server che esegue Windows Server, emetti i seguenti comandi per avviare la procedura guidata:

   ```
   cd "C:\Program Files\Amazon\AmazonCloudWatchAgent"
   ```

   ```
   .\amazon-cloudwatch-agent-config-wizard.exe
   ```

2. Rispondi alle domande per personalizzare il file di configurazione per il tuo server.
3. Se stai archiviando il file di configurazione in locale, il file di configurazione `config.json` viene archiviato in `/opt/aws/amazon-cloudwatch-agent/bin/` su server Linux e viene archiviato in `C:\Program Files\Amazon\AmazonCloudWatchAgent` su Windows Server. Sarà possibile copiare questo file negli altri server in cui si desidera installare l'agente.

   Se si intende utilizzare Systems Manager per installare e configurare l'agente, assicurarti di rispondere Yes (Sì) quando viene richiesto se archiviare il file in Systems Manager Parameter Store. Puoi anche scegliere di archiviare il file in Parameter Store anche se non stai utilizzando l'agente SSM per installare l' CloudWatch agente. Per poter archiviare il file in Parameter Store, devi utilizzare un ruolo IAM con autorizzazioni sufficienti. Per ulteriori informazioni, consulta Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente.

CloudWatch set di metriche predefiniti dell'agente

La procedura guidata è configurata con set predefiniti di parametri, con diversi livelli di dettaglio. Questi set di parametri sono illustrati nelle tabelle seguenti. Per ulteriori informazioni su questi parametri, consulta Metriche raccolte dall'agente CloudWatch.

> **Note**
>
> Parameter Store supporta i parametri nei livelli Standard e Advanced. Questi livelli di parametri non sono correlati ai livelli Basic, Standard e Advanced dei dettagli dei parametri descritti in queste tabelle.

Istanze Amazon EC2 che eseguono Linux

Livello di dettaglio	Parametri inclusi
Base	Mem: mem_used_percent Disk: disk_used_percent I parametri `disk`, ad esempio `disk_used_percent` , hanno una dimension e per `Partition` , il che significa che il numero di parametri personalizzati generati dipende dal numero di partizioni associate all'istanza. Il numero di partizioni disponibili dipende dall'AMI in uso e dal numero di volumi Amazon EBS che colleghi al server.
Standard	CPU: `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_user` , `cpu_usage_system` Disk: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` Mem: `mem_used_percent` Swap: `swap_used_percent`
Avanzato	CPU: `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_user` , `cpu_usage_system` Disk: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads`

Livello di dettaglio	Parametri inclusi
	Mem: `mem_used_percent` Netstat: `netstat_tcp_established` , `netstat_tcp_time_wait` Swap: `swap_used_percent`

Server locali che eseguono Linux

Livello di dettaglio	Parametri inclusi
Base	Disk: `disk_used_percent` Diskio: `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Net: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Swap: `swap_used_percent`
Standard	CPU: `cpu_usage_idle` , `cpu_usage_iowait` Disk: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Net: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Swap: `swap_used_percent`

Livello di dettaglio	Parametri inclusi
Avanzato	CPU: `cpu_usage_guest` , `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_steal` , `cpu_usage_user` , `cpu_usage_system` Disk: `disk_used_percent` , `disk_inodes_free` Diskio: `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem: `mem_used_percent` Net: `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Netstat: `netstat_tcp_established` , `netstat_tcp_time_wait` Swap: `swap_used_percent`

Istanze Amazon EC2 che eseguono Windows Server

> **Note**
>
> I nomi dei parametri riportati in questa tabella mostrano come il parametro viene visualizzato nella console. Il nome effettivo del parametro potrebbe non includere la prima parola. Ad esempio, il nome effettivo del parametro per `LogicalDisk % Free Space` è solo `% Free Space`.

Livello di dettaglio	Parametri inclusi
Base	Memory: `Memory % Committed Bytes In Use` LogicalDisk: `LogicalDisk % Free Space`
Standard	Memory: `Memory % Committed Bytes In Use`

Livello di dettaglio	Parametri inclusi
	Paging: `Paging File % Usage` Processor: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` PhysicalDisk: `PhysicalDisk % Disk Time` LogicalDisk: `LogicalDisk % Free Space`
Avanzato	Memory: `Memory % Committed Bytes In Use` Paging: `Paging File % Usage` Processor: `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` TCP: `TCPv4 Connections Established` , `TCPv6 Connections Established`

Server locale che esegue Windows Server

> **Note**
>
> I nomi dei parametri riportati in questa tabella mostrano come il parametro viene visualizzato nella console. Il nome effettivo del parametro potrebbe non includere la prima parola. Ad esempio, il nome effettivo del parametro per `LogicalDisk % Free Space` è solo `% Free Space`.

Livello di dettaglio	Parametri inclusi
Base	Paging: `Paging File % Usage` Processor: `Processor % Processor Time` LogicalDisk:`LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` Network Interface: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Standard	Paging: `Paging File % Usage` Processor: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` Network Interface: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Avanzato	Paging:`Paging File % Usage`

Livello di dettaglio	Parametri inclusi
	Processor: `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Memory: `Memory % Committed Bytes In Use` Network Interface: `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec` TCP: `TCPv4 Connections Established` , `TCPv6 Connections Established`

Crea o modifica manualmente il file di configurazione dell' CloudWatch agente

Il file di configurazione dell' CloudWatch agente è un file JSON con quattro sezioni`agent`,`metrics`,`logs`, `etraces`, descritte come segue:

- La sezione `agent` include campi per la configurazione generale dell'agente.
- La `metrics` sezione specifica le metriche personalizzate per la raccolta e la pubblicazione su. CloudWatch Se intendi utilizzare l'agente solo per raccogliere log, puoi omettere dal file la sezione `metrics`.
- La `logs` sezione specifica quali file di registro vengono pubblicati in Logs. CloudWatch Questo può includere eventi del log eventi di Windows, se Windows Server è in esecuzione nel server.
- La `traces` sezione specifica le fonti per le tracce che vengono raccolte e inviate. AWS X-Ray

Nelle seguenti sezioni vengono illustrati i campi e la struttura di questo file JSON. Puoi inoltre visualizzare la definizione di schema di questo file di configurazione. La definizione di schema è ubicata in *installation-directory*/doc/amazon-cloudwatch-agent-schema.json sui server Linux e in *installation-directory*/amazon-cloudwatch-agent-schema.json sui server che eseguono Windows Server.

Se crei o modifichi manualmente il file di configurazione dell'agente di , puoi assegnargli qualsiasi nome. Per semplicità nella risoluzione dei problemi, consigliamo di denominarlo /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json su un server Linux e $Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json sui server che eseguono Windows Server. Dopo avere creato il file, potrai copiarlo negli altri server in cui desideri installare l'agente.

> **Note**
>
> Le metriche, i log e le tracce raccolti dall' CloudWatch agente sono a pagamento. Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

CloudWatch file di configurazione dell'agente: sezione Agente

La sezione `agent` può includere i seguenti campi. La procedura guidata non crea una sezione `agent`. Al contrario, la procedura guidata la omette e utilizza i valori predefiniti per tutti i campi della sezione.

- `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta di tutti i parametri specificati in questo file di configurazione. Puoi ignorare questo valore per determinati tipi di parametri.

  Questo valore è specificato in secondi. Ad esempio, il valore 10 causa la raccolta dei parametri ogni 10 secondi; il valore 300 specifica che i parametri devono essere raccolti ogni 5 minuti.

  Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Parametri ad alta risoluzione.

  Il valore predefinito è 60.
- `region`: specifica la regione da utilizzare per l' CloudWatch endpoint quando viene monitorata un'istanza Amazon EC2. I parametri raccolti vengono inviati a questa regione, ad esempio us-

`west-1`. Se ometti questo campo, l'agente invia i parametri alla regione in cui si trova l'istanza Amazon EC2.

Se monitori un server locale, questo campo non viene utilizzato e l'agente legge la regione dal profilo `AmazonCloudWatchAgent` del file di configurazione AWS .

- `credentials`— Speciifica un ruolo IAM da utilizzare per l'invio di metriche, log e tracce a un altro account. AWS Se specificato, questo campo contiene un parametro, `role_arn`.
  - `role_arn`: specifica il nome della risorsa Amazon (ARN) di un ruolo IAM da utilizzare per l'autenticazione durante l'invio di parametri, log e tracce a un account AWS diverso. Per ulteriori informazioni, consulta Invio di parametri, log e tracce a un altro account.
- `debug` Facoltativo. Specifica l'esecuzione dell' CloudWatch agente con messaggi di registro di debug. Il valore predefinito è `false`.
- `aws_sdk_log_level` Facoltativo. Supportato solo nelle versioni 1.247350.0 e successive dell'agente. CloudWatch

  È possibile specificare questo campo per consentire all'agente di eseguire la registrazione per gli endpoint SDK. AWS Il valore di questo campo può includere una o più opzioni tra le seguenti. Separare più opzioni con il carattere `|`.
  - `LogDebug`
  - `LogDebugWithSigning`
  - `LogDebugWithHTTPBody`
  - `LogDebugRequestRetries`
  - `LogDebugWithEventStreamBody`

  Per ulteriori informazioni su queste opzioni, consulta. LogLevelType
- `logfile`— Speciifica la posizione in cui l' CloudWatch agente scrive i messaggi di registro. Se specifichi una stringa vuota, il log passa a stderr. Se non specifichi questa opzione, le ubicazioni predefinite sono le seguenti:
  - Linux: `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`
  - Windows Server: `c:\\ProgramData\\Amazon\\CloudWatchAgent\\Logs\\amazon-cloudwatch-agent.log`

  L' CloudWatch agente ruota automaticamente il file di registro che crea. Un file di log viene ruotato in uscita fino a quando raggiunge la dimensione di 100 MB. L'agente mantiene i file di log ruotati fino a sette giorni e mantiene cinque file di log di backup che sono stati ruotati in uscita. Al nome

dei file di backup dei log viene aggiunto un timestamp, che mostra la data e l'ora in cui il file è stato rimosso: ad esempio, `amazon-cloudwatch-agent-2018-06-08T21-01-50.247.log.gz`.

- `omit_hostname` Facoltativo. Per impostazione predefinita, il nome host viene pubblicato come una dimensione di parametri che sono raccolti dall'agente, a meno che non venga usato il campo `append_dimensions` nella sezione `metrics`. Imposta da `omit_hostname` a `true` per impedire che il nome host venga pubblicato come una dimensione anche se non si utilizza `append_dimensions`. Il valore predefinito è `false`.
- `run_as_user` Facoltativo. Speciifica un utente da utilizzare per eseguire l' CloudWatch agente. Se non si specifica questo parametro, viene utilizzato l'utente root. Questa opzione è valida solo su server Linux.

  Se si specifica questa opzione, l'utente deve esistere prima di avviare l' CloudWatch agente. Per ulteriori informazioni, consulta Esecuzione dell' CloudWatch agente come utente diverso.
- `user_agent` Facoltativo. Speciifica la `user-agent` stringa utilizzata dall' CloudWatch agente quando effettua chiamate API al CloudWatch backend. Il valore predefinito è una stringa costituita dalla versione dell'agente, dalla versione del linguaggio di programmazione Go utilizzata per compilare l'agente, dal sistema operativo di default runtime e dall'architettura, dal tempo di compilazione e dai plug-in abilitati.
- `usage_data`: opzionale. Per impostazione predefinita, l' CloudWatch agente invia dati sullo stato e sulle prestazioni su se stesso CloudWatch ogni volta che pubblica metriche o esegue i log. CloudWatch Questi dati non comportano alcun costo. È possibile impedire all'agente di inviare questi dati specificando `false` per `usage_data`. Se si omette questo parametro, viene utilizzata l'impostazione predefinita di `true` e l'agente invia i dati dell'integrità e delle prestazioni.

  Se si imposta questo valore su `false`, affinché la modifica abbia effetto sarà necessario arrestare e riavviare l'agente.

Di seguito è riportato un esempio di una sezione `agent`.

```
"agent": {
   "metrics_collection_interval": 60,
   "region": "us-west-1",
   "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log",
   "debug": false,
   "run_as_user": "cwagent"
  }
```

CloudWatch file di configurazione dell'agente: sezione Metrics

Campi comuni a Linux e Windows

Nei server in cui è in esecuzione Linux o Windows Server, la sezione `metrics` include i seguenti campi:

- `namespace` Facoltativo. Lo spazio dei nomi da utilizzare per i parametri raccolti dall'agente. Il valore predefinito è `CWAgent`. La lunghezza massima è 255 caratteri. Di seguito è riportato un esempio:

```
{
  "metrics": {
    "namespace": "Development/Product1Metrics",
   ......
   },
}
```

- `append_dimensions` Facoltativo. Aggiunge le dimensioni dei parametri Amazon EC2 a tutti i parametri raccolti dall'agente. In questo modo l'agente non pubblica il nome host come dimensione.

  Le uniche coppie chiave-valore supportate per `append_dimensions` sono mostrate nell'elenco seguente. Qualsiasi altra coppia chiave-valore viene ignorata. L'agente supporta queste coppie chiave-valore esattamente come sono mostrate nell'elenco seguente. Non è possibile modificare i valori della chiave per pubblicare nomi di dimensione diversi.

  - `"ImageId":"${aws:ImageId}"` imposta l'ID AMI dell'istanza come valore della dimensione `ImageId`.
  - `"InstanceId":"${aws:InstanceId}"` imposta l'ID istanza dell'istanza come valore della dimensione `InstanceId`.
  - `"InstanceType":"${aws:InstanceType}"` imposta il tipo di istanza dell'istanza come valore della dimensione `InstanceType`.
  - `"AutoScalingGroupName":"${aws:AutoScalingGroupName}"` imposta il nome del gruppo Auto Scaling dell'istanza come valore della dimensione `AutoScalingGroupName`.

  Se si desidera aggiungere dimensioni ai parametri con coppie chiave-valore arbitrarie, utilizza il parametro `append_dimensions` nel campo per quel particolare tipo di parametro.

  Se specifichi un valore che dipende da metadati Amazon EC2 e utilizzi proxy, devi assicurarti che il server possa accedere all'endpoint per Amazon EC2. Per ulteriori informazioni su questi endpoint,

consulta Amazon Elastic Compute Cloud (Amazon EC2) nella Riferimenti generali di Amazon Web Services.

- `aggregation_dimensions` Facoltativo. Specifica le dimensioni su cui devono essere aggregati i parametri raccolti. Ad esempio, se esegui il rollup dei parametri nella dimensione `AutoScalingGroupName`, vengono aggregati i parametri da tutte le istanze in ogni gruppo Auto Scaling, che possono essere visualizzati complessivamente.

  Puoi eseguire il rollup dei parametri lungo una o più dimensioni. Ad esempio, specificando `[["InstanceId"], ["InstanceType"], ["InstanceId","InstanceType"]]` aggrega i parametri per ID istanza singolarmente, tipo di istanza singolarmente e per la combinazione delle due dimensioni.

  Puoi inoltre specificare `[]` per eseguire il rollup di tutti i parametri in un'unica raccolta, ignorando tutte le dimensioni.

- `endpoint_override`: specifica un endpoint FIPS o un collegamento privato da utilizzare come endpoint in cui l'agente invia i parametri. La specifica relativa e l'impostazione di un collegamento privato consente di inviare i parametri a un endpoint Amazon VPC. Per ulteriori informazioni, consulta Cos'è Amazon VPC?.

  Il valore di `endpoint_override` deve essere una stringa che è un URL.

  Ad esempio, la parte seguente della sezione dei parametri del file di configurazione imposta l'agente per utilizzare un endpoint VPC durante l'invio dei parametri.

  ```
  {
    "metrics": {
      "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.monitoring.us-
  east-1.vpce.amazonaws.com",
     ......
     },
  }
  ```

- `metrics_collected`: obbligatorio. Specifica i parametri da raccogliere, inclusi quelli personalizzati raccolti tramite `StatsD` o `collectd`. Questa sezione include varie sottosezioni.

  Il contenuto della sezione `metrics_collected` varia se il file di configurazione è per un server che esegue Linux o Windows Server.

- `force_flush_interval`: specifica in secondi la quantità massima di tempo in cui i parametri rimangono nel buffer di memoria prima di essere inviati al server. Indipendentemente

dall'impostazione, se la dimensione dei parametri nel buffer raggiunge 1 MB o 1.000 parametri differenti, i parametri vengono immediatamente inviati al server.

Il valore predefinito è 60.

- `credentials`: specifica un ruolo IAM da utilizzare durante l'invio di parametri a un account diverso. Se specificato, questo campo contiene un parametro, `role_arn`.
  - `role_arn`: specifica l'ARN di un ruolo IAM da utilizzare per l'autenticazione durante l'invio di parametri a un account diverso. Per ulteriori informazioni, consulta la pagina Invio di parametri, log e tracce a un altro account. Se specificato qui, questo valore sostituisce il `role_arn` specificato nella sezione `agent` del file di configurazione, se presente.

Sezione Linux

Nei server con Linux in esecuzione, la sezione `metrics_collected` del file di configurazione può anche includere i seguenti campi.

Molti di questi campi possono includere una sezione `measurement` che elenca i parametri che si desidera raccogliere per quella risorsa. Queste sezioni `measurement` possono specificare il nome di parametro completo, ad esempio `swap_used`, oppure solo la parte del nome parametro che verrà aggiunta al tipo della risorsa. Ad esempio, specificando `reads` nella sezione `measurement` della sezione `diskio` si causa la raccolta del parametro `diskio_reads`.

- `collectd` Facoltativo. Specifica che desideri recuperare i parametri personalizzati utilizzando il protocollo `collectd`. Si utilizza `collectd` un software per inviare le metriche all' CloudWatch agente. Per ulteriori informazioni sulle opzioni di configurazione disponibili per collectd, consulta Recupero dei parametri personalizzati con collectd.
- `cpu` Facoltativo. Specifica che è necessario raccogliere i parametri della CPU. Questa sezione è valida solo per le istanze Linux. È necessario includere almeno uno dei campi `totalcpu` e `resources` per tutti i parametri CPU da raccogliere. Questa sezione può includere i seguenti campi.
  - `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono

invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.

- `resources`: opzionale. Specifica questo campo con il valore * per far sì che i parametri per CPU vengano raccolti. L'unico valore consentito è *.
- `totalcpu` Facoltativo. Specifica se segnalare i parametri di CPU aggregati in tutti i core CPU. Il valore predefinito è true.
- `measurement`: specifica la matrice di parametri cpu da raccogliere. I valori possibili sono `time_active`, `time_guest`, `time_guest_nice`, `time_idle`, `time_iowait`, `time_irq`, `time_nice`, `time_softirq`, `time_steal`, `time_system`, `time_user`, `usage_active`, `usage_guest`, `usage_guest_nice`, `usage_idle`, `usage_iowait`, `usage_irq`, `usage_nice`, `usage_softirq`, `usage_steal`, `usage_system` e `usage_user`. Il campo è obbligatorio se includi `cpu`.

  Per impostazione predefinita, l'unità di misura dei parametri `cpu_usage_*` è `Percent`, mentre i parametri `cpu_time_*` non hanno un'unità di misura.

  Nelle voci di ciascun parametro individuale, potresti specificare uno o più dei seguenti valori:

  - `rename`: specifica un nome diverso per questo parametro.
  - `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di `None` per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come elencato nella `Unit` descrizione in. MetricDatum
- `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri di CPU, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

  Questo valore è specificato in secondi. Ad esempio, il valore 10 causa la raccolta dei parametri ogni 10 secondi; il valore 300 specifica che i parametri devono essere raccolti ogni 5 minuti.

  Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Parametri ad alta risoluzione.
- `append_dimensions` Facoltativo. Le dimensioni aggiuntive da utilizzare solo per i parametri di CPU. Se specifichi questo campo, verrà utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions` globale, usato per tutti i tipi di parametri raccolti dall'agente.

- `disk` Facoltativo. Specifica che i parametri del disco devono essere raccolti. Questa sezione è valida solo per le istanze Linux. Questa sezione può includere i seguenti campi.

- `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.
- `resources` Facoltativo. Specifica una matrice di punti di montaggio su disco. Questo campo si limita CloudWatch a raccogliere le metriche solo dai punti di montaggio elencati. Puoi specificare `*` come valore per raccogliere parametri da tutti i punti di montaggio. Il valore predefinito raccoglie i parametri da tutti i punti di montaggio.
- `measurement`: specifica la matrice di parametri disk da raccogliere. I valori possibili sono `free`, `total`, `used`, `used_percent`, `inodes_free`, `inodes_used` e `inodes_total`. Il campo è obbligatorio se includi `disk`.

  > **Note**
  >
  > I parametri `disk` hanno una dimensione per `Partition`, il che significa che il numero di parametri personalizzati generati dipende dal numero di partizioni associate all'istanza. Il numero di partizioni disponibili dipende dall'AMI in uso e dal numero di volumi Amazon EBS che colleghi al server.

  Per visualizzare le unità di misura predefinite per ciascun parametro `disk`, consulta Metriche raccolte dall' CloudWatchagente sulle istanze Linux e macOS.

  Nelle voci di ciascun parametro individuale, potresti specificare uno o più dei seguenti valori:

  - `rename`: specifica un nome diverso per questo parametro.
  - `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di None di None per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come elencato nella `Unit` descrizione in. MetricDatum
- `ignore_file_system_types`: specifica i tipi di file system da escludere durante la raccolta di parametri disk. Tra i valori validi figurano `sysfs`, `devtmpfs` e così via.
- `drop_device`: impostando questa opzione su `true`, `Device` non viene incluso come dimensione per i parametri disk.

Impedire che `Device` venga utilizzato come dimensione può essere utile sulle istanze che utilizzano il sistema Nitro perché su tali istanze i nomi dei dispositivi cambiano per ogni montaggio del disco quando l'istanza viene riavviata. Ciò può causare dati incoerenti nei parametri e che gli allarmi basati su tali parametri vadano in stato `INSUFFICIENT DATA`.

Il valore predefinito è `false`.

- `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri del disco, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

  Questo valore è specificato in secondi.

  Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni, consulta la pagina Parametri ad alta risoluzione.

- `append_dimensions` Facoltativo. Le dimensioni aggiuntive da utilizzare solo per i parametri del disco. Se specifichi questo campo, viene utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions`, usato per tutti i tipi di parametri raccolti dall'agente.

- `diskio` Facoltativo. Specifica che è necessario raccogliere i parametri disk i/o. Questa sezione è valida solo per le istanze Linux. Questa sezione può includere i seguenti campi.
  - `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.
  - `resources`: opzionale. Se specifichi una serie di dispositivi, CloudWatch raccoglie le metriche solo da tali dispositivi. In caso contrario, verranno raccolti parametri da tutti i dispositivi. Puoi inoltre specificare * come valore per raccogliere parametri da tutti i dispositivi.
  - `measurement`: specifica la matrice di parametri diskio da raccogliere. I valori possibili sono `reads`, `writes`, `read_bytes`, `write_bytes`, `read_time`, `write_time`, `io_time` e `iops_in_progress`. Il campo è obbligatorio se includi `diskio`.

    Nelle voci di ciascun parametro individuale, potresti specificare uno o più dei seguenti valori:

    - `rename`: specifica un nome diverso per questo parametro.

    - `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di `None` di `None` per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come indicato nella `Unit` descrizione in. MetricDatum
  - `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri diskio, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

    Questo valore è specificato in secondi.

    Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Parametri ad alta risoluzione.
  - `append_dimensions` Facoltativo. Le dimensioni aggiuntive da utilizzare solo per i parametri diskio. Se specifichi questo campo, viene utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions`, usato per tutti i tipi di parametri raccolti dall'agente.
- `swap` Facoltativo. Specifica che i parametri di memoria di scambio devono essere raccolti. Questa sezione è valida solo per le istanze Linux. Questa sezione può includere i seguenti campi.
  - `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.
  - `measurement`: specifica la matrice di parametri swap da raccogliere. I valori possibili sono `free`, `used` e `used_percent`. Il campo è obbligatorio se includi `swap`.

    Per visualizzare le unità di misura predefinite per ciascun parametro `swap`, consulta Metriche raccolte dall' CloudWatchagente sulle istanze Linux e macOS.

    Nelle voci di ciascun parametro individuale, potresti specificare uno o più dei seguenti valori:

    - `rename`: specifica un nome diverso per questo parametro.
    - `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di `None` di `None` per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come elencato nella `Unit` descrizione in. MetricDatum

- `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri di scambio, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

  Questo valore è specificato in secondi.

  Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Parametri ad alta risoluzione.

- `append_dimensions` Facoltativo. Le dimensioni aggiuntive da utilizzare solo per i parametri di scambio. Se specifichi questo campo, viene utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions` globale, usato per tutti i tipi di parametri raccolti dall'agente. Viene raccolto come parametro ad alta risoluzione.

- `mem` Facoltativo. Specifica che i parametri di memoria devono essere raccolti. Questa sezione è valida solo per le istanze Linux. Questa sezione può includere i seguenti campi.

  - `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.

  - `measurement`: specifica la matrice di parametri memory da raccogliere. I valori possibili sono `active`, `available`, `available_percent`, `buffered`, `cached`, `free`, `inactive`, `total`, `used` e `used_percent`. Il campo è obbligatorio se includi `mem`.

    Per visualizzare le unità di misura predefinite per ciascun parametro `mem`, consulta Metriche raccolte dall' CloudWatchagente sulle istanze Linux e macOS.

    Nelle voci di ciascun parametro individuale, potresti specificare uno o più dei seguenti valori:

    - `rename`: specifica un nome diverso per questo parametro.
    - `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di `None` per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come elencato nella `Unit` descrizione in. MetricDatum

  - `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri di memoria, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

    Questo valore è specificato in secondi.

    Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Parametri ad alta risoluzione.
  - `append_dimensions` Facoltativo. Le dimensioni aggiuntive da utilizzare solo per i parametri di memoria. Se specifichi questo campo, verrà utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions`, usato per tutti i tipi di parametri raccolti dall'agente.
- `net` Facoltativo. Specifica che i parametri di rete devono essere raccolti. Questa sezione è valida solo per le istanze Linux. Questa sezione può includere i seguenti campi.
  - `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.
  - `resources`: opzionale. Se specifichi una serie di interfacce di rete, CloudWatch raccoglie le metriche solo da tali interfacce. In caso contrario, verranno raccolti parametri da tutti i dispositivi. Puoi inoltre specificare `*` come valore per raccogliere parametri da tutte le interfacce.
  - `measurement`: specifica la matrice di parametri di rete da raccogliere. I valori possibili sono `bytes_sent`, `bytes_recv`, `drop_in`, `drop_out`, `err_in`, `err_out`, `packets_sent` e `packets_recv`. Il campo è obbligatorio se includi `net`.

    Per visualizzare le unità di misura predefinite per ciascun parametro `net`, consulta Metriche raccolte dall' CloudWatchagente sulle istanze Linux e macOS.

    Nelle voci di ciascun parametro individuale, potresti specificare uno o più dei seguenti valori:

    - `rename`: specifica un nome diverso per questo parametro.

    - `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di `None` per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come elencato nella descrizione in. `Unit` MetricDatum
  - `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri di rete, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

    Questo valore è specificato in secondi. Ad esempio, il valore 10 causa la raccolta dei parametri ogni 10 secondi; il valore 300 specifica che i parametri devono essere raccolti ogni 5 minuti.

    Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Parametri ad alta risoluzione.
  - `append_dimensions` Facoltativo. Le dimensioni aggiuntive da utilizzare solo per i parametri di rete. Se specifichi questo campo, viene utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions`, usato per tutti i tipi di parametri raccolti dall'agente.
- `netstat` Facoltativo. Specifica che i parametri della connessione TCP e della connessione UDP devono essere raccolti. Questa sezione è valida solo per le istanze Linux. Questa sezione può includere i seguenti campi.
  - `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.
  - `measurement`: specifica la matrice di parametri netstat da raccogliere. I valori possibili sono `tcp_close`, `tcp_close_wait`, `tcp_closing`, `tcp_established`, `tcp_fin_wait1`, `tcp_fin_wait2`, `tcp_last_ack`, `tcp_listen`, `tcp_none`, `tcp_syn_sent`, `tcp_syn_recv`, `tcp_time_wait` e `udp_socket`. Il campo è obbligatorio se includi `netstat`.

    Per visualizzare le unità di misura predefinite per ciascun parametro `netstat`, consulta Metriche raccolte dall' CloudWatchagente sulle istanze Linux e macOS.

    Nelle voci di ciascun parametro individuale, potresti specificare uno o più dei seguenti valori:

    - `rename`: specifica un nome diverso per questo parametro.
    - `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di `None` per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come elencato nella `Unit` descrizione in. MetricDatum
  - `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri netstat, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

    Questo valore è specificato in secondi.

    Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Parametri ad alta risoluzione.
  - `append_dimensions` Facoltativo. Le dimensioni aggiuntive da utilizzare solo per i parametri di netstat. Se specifichi questo campo, viene utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions`, usato per tutti i tipi di parametri raccolti dall'agente.
- `processes` Facoltativo. Specifica che i parametri di processo devono essere raccolti. Questa sezione è valida solo per le istanze Linux. Questa sezione può includere i seguenti campi.
  - `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.
  - `measurement`: specifica la matrice di parametri processes da raccogliere. I valori possibili sono `blocked`, `dead`, `idle`, `paging`, `running`, `sleeping`, `stopped`, `total`, `total_threads`, `wait` e `zombies`. Il campo è obbligatorio se includi `processes`.

    Per tutti i parametri `processes`, l'unità di misura predefinita è `None`.

    Nelle voci di ciascun parametro individuale, potresti specificare uno o più dei seguenti valori:
    - `rename`: specifica un nome diverso per questo parametro.

- `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di `None` per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come elencato nella `Unit` descrizione in. MetricDatum

- `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri di processo, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

  Questo valore è specificato in secondi. Ad esempio, il valore 10 causa la raccolta dei parametri ogni 10 secondi; il valore 300 specifica che i parametri devono essere raccolti ogni 5 minuti.

  Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni, consulta la pagina Parametri ad alta risoluzione.

- `append_dimensions` Facoltativo. Le dimensioni aggiuntive da utilizzare solo per i parametri di processo. Se specifichi questo campo, viene utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions`, usato per tutti i tipi di parametri raccolti dall'agente.

- `nvidia_gpu` Facoltativo. Specifica che i parametri della GPU NVIDIA devono essere raccolti. Questa sezione è valida solo per le istanze Linux su host configurati con un acceleratore della GPU NVIDIA e hanno NVIDIA System Management Interface (nvidia-smi) installato.

  I parametri della GPU NVIDIA raccolti hanno come prefisso la stringa `nvidia_smi_` per distinguerli da quelli raccolti per altri tipi di acceleratori. Questa sezione può includere i seguenti campi.

  - `drop_original_metrics` Facoltativo. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.

  - `measurement`: specifica la matrice dei parametri della GPU NVIDIA da raccogliere. Per un elenco dei possibili valori da utilizzare, consulta la colonna Metric (Parametri) della tabella in Raccolta dei parametri della GPU NVIDIA.

    Nelle voci di ciascun parametro individuale, puoi definire uno o più dei seguenti valori:

    - `rename`: specifica un nome diverso per questo parametro.

    - `unit`: specifica l'unità di misura da utilizzare per questo parametro, la quale sostituisce l'unità di misura predefinita di `None` per il parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come elencato nella `Unit` descrizione in. MetricDatum
    - `metrics_collection_interval`: opzionale. Specifica la frequenza di raccolta dei parametri della GPU NVIDIA, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.
- `procstat` Facoltativo. Specifica che si desidera recuperare i parametri dai singoli processi. Per ulteriori informazioni sulle opzioni di configurazione disponibili per procstat, consulta Raccolta di parametri dei processi con il plug-in procstat.
- `statsd` Facoltativo. Specifica che desideri recuperare i parametri personalizzati utilizzando il protocollo `StatsD`. L' CloudWatch agente funge da demone per il protocollo. Utilizzi qualsiasi `StatsD` client standard per inviare le metriche all'agente. CloudWatch Per ulteriori informazioni sulle opzioni di configurazione disponibili per StatsD, consulta Recupero dei parametri personalizzati con StatsD .
- `ethtool` Facoltativo. Specifica che desideri recuperare i parametri di rete utilizzando il plug-in `ethtool`. Questo plug-in può importare sia i parametri raccolti dall'utilità standard ethtool, sia i parametri delle prestazioni di rete dalle istanze Amazon EC2. Per ulteriori informazioni sulle opzioni di configurazione disponibili per ethtool, consulta Raccolta di parametri sulle prestazioni di rete.

Di seguito è riportato un esempio di una sezione `metrics` per un server Linux. In questo esempio vengono raccolti tre parametri CPU, tre parametri netstat, tre parametri di processo e un parametro del disco e l'agente è impostato per ricevere ulteriori parametri da un client `collectd`.

```
"metrics": {
    "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId",
 "InstanceType"],[]],
    "metrics_collected": {
      "collectd": {},
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit": "Percent"},
          {"name": "cpu_usage_nice", "unit": "Percent"},
          "cpu_usage_guest"
        ],
        "totalcpu": false,
```

```
        "drop_original_metrics": [ "cpu_usage_guest" ],
        "metrics_collection_interval": 10,
        "append_dimensions": {
          "test": "test1",
          "date": "2017-10-01"
        }
      },
      "netstat": {
        "measurement": [
          "tcp_established",
          "tcp_syn_sent",
          "tcp_close"
        ],
        "metrics_collection_interval": 60
      },
       "disk": {
        "measurement": [
          "used_percent"
        ],
        "resources": [
          "*"
        ],
        "drop_device": true
      },
      "processes": {
        "measurement": [
          "running",
          "sleeping",
          "dead"
        ]
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    }
  }
```

Windows Server

Nella sezione `metrics_collected` per Windows Server, puoi avere sottosezioni per ciascun oggetto prestazione di Windows, ad esempio `Memory`, `Processor` e `LogicalDisk`. Per informazioni sugli oggetti e sui contatori disponibili, consulta Contatori delle prestazioni nella documentazione di Microsoft Windows.

All'interno della sottosezione per ciascun oggetto, è necessario specificare una matrice `measurement` dei contatori da raccogliere. La matrice `measurement` è obbligatoria per ciascun oggetto specificato nel file di configurazione. Puoi inoltre specificare un campo `resources` per denominare le istanze da cui raccogliere i parametri. Puoi inoltre specificare `*` per `resources` per raccogliere parametri separati per ogni istanza. Se ometti `resources` per i contatori che hanno istanze, i dati per tutte le istanze vengono aggregati in un unico set. Se ometti `resources` per i contatori che non hanno istanze, i contatori non vengono raccolti dall'agente. CloudWatch Per determinare se i contatori dispongono della istanza, è possibile utilizzare uno dei seguenti comandi.

Powershell:

```
Get-Counter -ListSet *
```

Riga di comando (non Powershell):

```
TypePerf.exe -q
```

All'interno di ogni sezione di oggetti, puoi inoltre specificare i seguenti campi facoltativi:

- `metrics_collection_interval` Facoltativo. Specifica la frequenza di raccolta dei parametri di questo oggetto, sostituendo il valore di `metrics_collection_interval` globale specificato nella sezione `agent` del file di configurazione.

  Questo valore è specificato in secondi. Ad esempio, il valore 10 causa la raccolta dei parametri ogni 10 secondi; il valore 300 specifica che i parametri devono essere raccolti ogni 5 minuti.

  Se imposti questo valore al di sotto di 60 secondi, ogni parametro viene raccolto come parametro ad alta risoluzione. Per ulteriori informazioni, consulta la pagina Parametri ad alta risoluzione.
- `append_dimensions` Facoltativo. Specifica le dimensioni aggiuntive da utilizzare solo per i parametri di questo oggetto. Se specifichi questo campo, verrà utilizzato in aggiunta alle dimensioni specificate nel campo `append_dimensions` globale, usato per tutti i tipi di parametri raccolti dall'agente.

- `drop_original_metrics`: opzionale. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.

All'interno di ogni sezione di contatori, puoi inoltre specificare i seguenti campi facoltativi:

- `rename`: specifica un nome diverso da utilizzare per questa metrica CloudWatch .
- `unit`: specifica l'unità di misura da utilizzare per questo parametro. L'unità specificata deve essere un'unità CloudWatch metrica valida, come indicato nella `Unit` descrizione in. MetricDatum

Esistono altre due sezioni opzionali che puoi includere in `metrics_collected`:

- `statsd`: consente di recuperare i parametri personalizzati utilizzando il protocollo `StatsD`. L' CloudWatch agente funge da demone per il protocollo. Utilizzi qualsiasi `StatsD` client standard per inviare le metriche all'agente. CloudWatch Per ulteriori informazioni, consulta Recupero dei parametri personalizzati con StatsD .
- `procstat`: consente di recuperare i parametri dai singoli processi. Per ulteriori informazioni, consulta la pagina Raccolta di parametri dei processi con il plug-in procstat.

Di seguito è riportato un esempio di una sezione `metrics` per l'uso su Windows Server. In questo esempio, vengono raccolti molti parametri di Windows e il computer è impostato per ricevere ulteriori parametri da un client `StatsD`.

```
"metrics": {
    "metrics_collected": {
      "statsd": {},
      "Processor": {
        "measurement": [
          {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
          "% Interrupt Time",
          "% User Time",
          "% Processor Time"
```

```
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      },
      "LogicalDisk": {
        "measurement": [
          {"name": "% Idle Time", "unit": "Percent"},
          {"name": "% Disk Read Time", "rename": "DISK_READ"},
          "% Disk Write Time"
        ],
        "resources": [
          "*"
        ]
      },
      "Memory": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Available Bytes",
          "Cache Faults/sec",
          "Page Faults/sec",
          "Pages/sec"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "Network Interface": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Bytes Received/sec",
          "Bytes Sent/sec",
          "Packets Received/sec",
          "Packets Sent/sec"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d3": "win_bo"
```

```
        }
      },
      "System": {
        "measurement": [
          "Context Switches/sec",
          "System Calls/sec",
          "Processor Queue Length"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    },
    "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],[]]
    }
  }
```

CloudWatch file di configurazione dell'agente: sezione Logs

La sezione `logs` include i seguenti campi:

- `logs_collected`: obbligatorio se è inclusa la sezione `logs`. Specifica quali file di log e log di eventi Windows devono essere raccolti dal server. Può includere due campi: `files` e `windows_events`.
  - `files`— Speciifica quali file di registro regolari l' CloudWatch agente deve raccogliere. Include il campo `collect_list`, che definisce ulteriormente questi file.
    - `collect_list`: campo obbligatorio se `files` è incluso. Contiene una matrice di voci, ciascuna delle quali specifica un file di log da raccogliere. Ciascuna di queste voci può includere i seguenti campi:
      - `file_path`— specifica il percorso del file di registro da caricare su Logs. CloudWatch Le regole di corrispondenza glob Standard Unix sono accettate, con l'aggiunta di `**` come super asterisco. Ad esempio, specificando `/var/log/**.log`, tutti i file `.log` nella

struttura di directory `/var/log` verranno raccolti. Per ulteriori esempi, consulta la pagina Web Glob Library.

Puoi anche utilizzare l'asterisco standard come carattere jolly. Ad esempio, `/var/log/system.log*` corrisponde ai file, ad esempio `system.log_1111`, `system.log_2222` e così via in `/var/log`.

Solo il file più recente viene inviato ai CloudWatch registri in base all'ora di modifica del file. Ti consigliamo di utilizzare i caratteri jolly per specificare una serie di file dello stesso tipo, ad esempio `access_log.2018-06-01-01` e `access_log.2018-06-01-02`, ma non file di più tipi, ad esempio `access_log_80` e `access_log_443`. Per specificare più tipi di file, aggiungi un'altra voce di flusso di log al file di configurazione dell'agente in modo che ciascun tipo di file di log abbia come destinazione un flusso di log distinto.

- `auto_removal` Facoltativo. In tal caso`true`, l' CloudWatch agente elimina automaticamente questo file di registro dopo averlo letto ed è stato ruotato. In genere i file di registro vengono eliminati dopo che l'intero contenuto è stato caricato su CloudWatch Logs, ma se l'agente raggiunge l'EOF (fine del file) e rileva anche un altro file di registro più recente che corrisponde allo stesso`file_path`, l'agente elimina il VECCHIO file, quindi è necessario assicurarsi di aver finito di scrivere sul file VECCHIO prima di creare il NUOVO file. La libreria di tracciamento RUST presenta un'incompatibilità nota perché potenzialmente creerà un NUOVO file di registro e quindi tenterà comunque di scrivere sul VECCHIO file di registro.

  L'agente rimuove solo i file completi dai log che creano più file, ad esempio i log che creano file separati per ogni data. Se un log scrive continuamente in un singolo file, non viene rimosso.

  Se hai già un metodo di rotazione o rimozione dei file di log, si consiglia di omettere questo campo o impostarlo su `false`.

  Se non compili questo campo, verrà utilizzato il valore predefinito `false`.

- `log_group_name` Facoltativo. Speciifica cosa usare come nome del gruppo di log in Logs. CloudWatch

  È consigliabile usare questo campo per specificare un nome del gruppo di log per evitare confusione. Se `log_group_name` non è specificato, viene utilizzato come nome del gruppo di log il valore del `file_path`, fino al punto finale. Ad esempio, se il percorso del file

è `/tmp/TestLogFile.log.2017-07-11-14`, il nome del gruppo di log sarà `/tmp/TestLogFile.log`.

Se specifichi un nome del gruppo di log, puoi utilizzare `{instance_id}`, `{hostname}`, `{local_hostname}` e `{ip_address}` come variabili all'interno del nome. `{hostname}` recupera il nome host dai metadati EC2, mentre `{local_hostname}` utilizza il nome host dal file di configurazione di rete.

Se utilizzi queste variabili per creare diversi gruppi di log, devi tenere presente il limite di 1.000.000 gruppi di log per Regione in ogni account.

I caratteri consentiti includono a-z, A-Z, 0-9, '_' (trattino basso), '-' (trattino), '/' (barra) e '.' (punto).

- `log_group_class`: opzionale. Specifica quale classe di gruppo di log usare per il nuovo gruppo di log. Per ulteriori informazioni sulle classi dei gruppi di log, consulta Classi di log.

  I valori validi sono `STANDARD` e `INFREQUENT_ACCESS`. Se ometti questo campo, verrà utilizzato il valore predefinito `STANDARD`.

  > **Important**
  >
  > Dopo la creazione di un gruppo di log, la relativa classe non può essere modificata.

- `log_stream_name`: opzionale. Specifica cosa usare come nome del flusso di log in Logs. CloudWatch Nel nome, puoi utilizzare `{instance_id}`, `{hostname}`, `{local_hostname}` e `{ip_address}` come variabili all'interno del nome. `{hostname}` recupera il nome host dai metadati EC2, mentre `{local_hostname}` utilizza il nome host dal file di configurazione di rete.

  Se ometti questo campo, viene utilizzato il valore del parametro `log_stream_name` nella sezione `logs` globale. Se anche questo viene omesso, viene utilizzato il valore predefinito di `{instance_id}`.

  Se non esiste, il flusso di log verrà creato automaticamente.

- `retention_in_days` Facoltativo. Specifica il numero di giorni in cui mantenere gli eventi di log nel gruppo di log specificato.
  - Se l'agente sta creando questo gruppo di log e si omette questo campo, la conservazione non avrà scadenza.

- Se questo gruppo di flussi di log esiste già e si specifica questo campo, viene utilizzata la nuova conservazione specificata. Se si omette questo campo per un gruppo di flussi di log già esistente, la conservazione del gruppo non viene modificata.

  La procedura guidata dell' CloudWatch agente utilizza `-1` come valore predefinito per questo campo quando viene utilizzato per creare il file di configurazione dell'agente e non si specifica un valore per la conservazione dei log. Questo `-1` valore impostato dalla procedura guidata specifica che gli eventi nel gruppo di log non scadranno mai. Tuttavia, la modifica manuale di questo valore su `-1` non ha alcun effetto.

I valori validi sono 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288 e 3653.

Se configuri l'agente in modo che scriva più flussi di log nello stesso gruppo, la specificazione del parametro `retention_in_days` in un'unica posizione imposterà la conservazione dei log per l'intero gruppo. Se definisci il parametro `retention_in_days` per lo stesso gruppo di log in più posizioni, la conservazione viene impostata se tutti i valori sono uguali. Tuttavia, se sono specificati valori del parametro `retention_in_days` diversi per lo stesso gruppo di log in più posizioni, la conservazione dei log non verrà impostata e l'agente si fermerà, restituendo un errore.

> **Note**
>
> Per poter impostare le policy di conservazione, il ruolo o l'utente IAM dell'agente deve disporre della `logs:PutRetentionPolicy`. Per ulteriori informazioni, consulta la pagina Consentire all' CloudWatch agente di impostare una politica di conservazione dei log.

> **Warning**
>
> Se hai impostato la `retention_in_days` per un gruppo di log già esistente, tutti i log del gruppo pubblicati prima del numero di giorni definito verranno eliminati. Ad esempio, se si imposta il valore su 3, sarebbero eliminati tutti i log antecedenti agli ultimi 3 giorni.

- `filters`: opzionale. Può contenere una matrice di voci, ognuna delle quali definisce un'espressione regolare e un tipo di filtro per definire se pubblicare o eliminare le voci

di log corrispondenti al filtro. Se si omette questo campo, tutti i registri del file di registro vengono pubblicati in Logs. CloudWatch Se si include questo campo, l'agente elabora ogni messaggio di registro con tutti i filtri specificati e solo gli eventi di registro che superano tutti i filtri vengono pubblicati in Logs. CloudWatch Le voci di registro che non superano tutti i filtri rimarranno comunque nel file di registro dell'host, ma non verranno inviate a CloudWatch Logs.

Ogni voce nella matrice dei filtri può includere i seguenti campi:

- `type`: indica il tipo di filtro. I valori validi sono `include` e `exclude`. Con`include`, la voce di registro deve corrispondere all'espressione da pubblicare in CloudWatch Logs. Con`exclude`, ogni voce di registro che corrisponde al filtro non viene inviata a CloudWatch Logs.
- `expression`: una stringa di espressione regolare che segue la sintassi RE2.

> **Note**
>
> L' CloudWatch agente non controlla le prestazioni di alcuna espressione regolare fornita né limita il tempo di esecuzione della valutazione delle espressioni regolari. Consigliamo di fare attenzione a non scrivere un'espressione dispendiosa da valutare. Per ulteriori informazioni sui possibili problemi, vedete Regular expression Denial of Service - S ReDo

Ad esempio, il seguente estratto del file di configurazione dell' CloudWatch agente pubblica i log che sono richieste PUT e POST nei registri, ma CloudWatch escludono i log provenienti da Firefox.

```
"collect_list": [
  {
    "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
    "log_group_name": "test.log",
    "log_stream_name": "test.log",
    "filters": [
      {
        "type": "exclude",
        "expression": "Firefox"
      },
      {
        "type": "include",
        "expression": "P(UT|OST)"
```

```
      }
    ]
  },
  .....
]
```

> **Note**
>
> L'ordine dei filtri nel file di configurazione influisce sulle prestazioni. Nell'esempio precedente, l'agente elimina tutti i log corrispondenti a `Firefox` prima di iniziare a valutare il secondo filtro. Per fare in modo che vengano valutate meno voci di log da più di un filtro, inserisci prima il filtro che prevedi escluda più log nel file di configurazione.

- `timezone` Facoltativo. Specifica il fuso orario da utilizzare per l'inserimento di timestamp negli eventi di log. I valori validi sono `UTC` e `Local`. Il valore predefinito è `Local`.

  Questo parametro viene ignorato se non specifichi un valore per `timestamp_format`.

- `timestamp_format` Facoltativo. Specifica il formato del timestamp, utilizzando testo normale e simboli speciali che iniziano con %. Se ometti questo campo, verrà utilizzata l'ora corrente. Se utilizzi questo campo, puoi utilizzare i simboli del seguente elenco nel formato.

  Se una singola voce di log contiene due timestamp che corrispondono al formato, viene utilizzato il primo timestamp.

  Questo elenco di simboli è diverso dall'elenco utilizzato dal precedente agente Logs. CloudWatch Per un riepilogo delle differenze, consulta Differenze nel timestamp tra l'agente unificato e il precedente CloudWatch agente Logs CloudWatch .

  `%y`

  Anno senza secolo come numero decimale a cui è aggiunto uno zero Ad esempio, `19` per rappresentare 2019.

  `%Y`

  Anno con secolo come numero decimale Ad esempio, `2019`.

  `%b`

  Mese come il nome abbreviato nella lingua locale

`%B`

Mese come il nome completo nella lingua locale

`%m`

Mese come numero decimale a cui è aggiunto uno zero

`%-m`

Mese come numero decimale (senza l'aggiunta di uno zero)

`%d`

Giorno del mese come numero decimale a cui è aggiunto uno zero

`%-d`

Giorno del mese come numero decimale (senza l'aggiunta di uno zero)

`%A`

Nome completo di giorno feriale, ad esempio `Monday`

`%a`

Abbreviazione di giorno feriale, ad esempio `Mon`

`%H`

Ora (formato di 24 ore) come numero decimale a cui è aggiunto uno zero

`%I`

Ora (formato di 12 ore) come numero decimale a cui è aggiunto uno zero

`%-I`

Ora (formato di 12 ore) come numero decimale (senza l'aggiunta di uno zero)

`%p`

AM o PM

`%M`

Minuti come numero decimale a cui è aggiunto uno zero

`%-M`

Minuti come numero decimale (senza l'aggiunta di uno zero)

`%S`

Secondi come numero decimale a cui è aggiunto uno zero

`%-S`

Secondi come numero decimale (senza l'aggiunta di uno zero)

`%f`

Secondi frazionati come numero decimale (1-9 cifre), con l'aggiunta di zero a sinistra.

`%Z`

Fuso orario, ad esempio `PST`

`%z`

Fuso orario, espresso come la differenza tra il fuso orario locale e UTC. Ad esempio, `-0700`. È supportato solo questo formato. Ad esempio, `-07:00` non è un formato valido.

- `multi_line_start_pattern`: specifica il modello per identificare l'inizio di un messaggio di log. Un messaggio di log è composto da una riga corrispondente al modello e da tutte le righe successive non corrispondenti al modello.

  Se questo campo viene omesso, la modalità a più righe viene disabilitata e le righe che iniziano con caratteri diversi da spazi vuoti chiudono il messaggio di log precedente e iniziano un nuovo messaggio di log.

  Se includi questo campo, puoi specificare `{timestamp_format}` per utilizzare la stessa espressione regolare come formato del timestamp. Altrimenti, è possibile specificare un'espressione regolare diversa per CloudWatch Logs da utilizzare per determinare le righe iniziali delle voci su più righe.
- `encoding`: specifica la codifica del file di log in modo che il file possa essere letto correttamente. Se specifichi una codifica non corretta, potrebbe verificarsi una perdita di dati, in quanto i caratteri che non possono essere decodificati saranno sostituiti da altri caratteri.

  Il valore predefinito è `utf-8`. Di seguito sono riportati i valori possibili:

`ascii, big5, euc-jp, euc-kr, gbk, gb18030, ibm866, iso2022-jp, iso8859-2, iso8859-3, iso8859-4, iso8859-5, iso8859-6, iso8859-7, iso8859-8, iso8859-8-i, iso8859-10, iso8859-13, iso8859-14, iso8859-15, iso8859-16, koi8-r, koi8-u, macintosh, shift_jis, utf-8, utf-16, utf-16le, UTF-16, UTF-16LE, windows-874, windows-1250, windows-1251, windows-1252, windows-1253, windows-1254, windows-1255, windows-1256, windows-1257, windows-1258, x-mac-cyrillic`

- La sezione `windows_events` specifica il tipo di eventi Windows da raccogliere dai server con Windows Server in esecuzione. Include i seguenti campi:
  - `collect_list`: campo obbligatorio se `windows_events` è incluso. Specifica i tipi e i livelli di eventi Windows da raccogliere. Ciascun log da raccogliere presenta una voce in questa sezione, che può includere i seguenti campi:
    - `event_name`: specifica il tipo di eventi Windows da registrare. Equivale al nome del canale del log degli eventi di Windows, ad esempio `System`, `Security`, `Application` e così via. Questo campo è obbligatorio per ogni tipo di evento Windows da registrare.

      > **Note**
      >
      > Quando CloudWatch recupera i messaggi da un canale di registro di Windows, cerca il canale di registro in base alla sua proprietà. `Full Name` Nel frattempo, nel riquadro di spostamento del Visualizzatore eventi di Windows vengono visualizzate le proprietà `Log Name` dei canali di log. `Full Name` e `Log Name` non sempre corrispondono. Per verificare il `Full Name` di un canale, fare clic con il pulsante destro del mouse nel Visualizzatore eventi di Windows e aprire Properties (Proprietà).

    - `event_levels`: specifica i livelli di evento da registrare. È necessario specificare ciascun livello da registrare. I valori possibili sono `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` e `VERBOSE`. Questo campo è obbligatorio per ogni tipo di evento Windows da registrare.
    - `log_group_name`: obbligatorio. Specifica cosa usare come nome del gruppo di log in CloudWatch Logs.
    - `log_stream_name`: opzionale. Specifica cosa usare come nome del flusso di log in Logs. CloudWatch Nel nome, puoi utilizzare `{instance_id}`, `{hostname}`, `{local_hostname}` e `{ip_address}` come variabili all'interno del nome. `{hostname}`

recupera il nome host dai metadati EC2, mentre `{local_hostname}` utilizza il nome host dal file di configurazione di rete.

Se ometti questo campo, viene utilizzato il valore del parametro `log_stream_name` nella sezione `logs` globale. Se anche questo viene omesso, viene utilizzato il valore predefinito di `{instance_id}`.

Se non esiste, il flusso di log verrà creato automaticamente.

- `event_format` Facoltativo. Speciifica il formato da utilizzare per la memorizzazione degli eventi di Windows nei registri. CloudWatch `xml`utilizza il formato XML come in Windows Event Viewer. `text`utilizza il formato legacy dell'agente CloudWatch Logs.
- `retention_in_days`: opzionale. Specifica il numero di giorni in cui mantenere gli eventi di Windows nel gruppo di log specificato.
  - Se l'agente sta creando questo gruppo di log e si omette questo campo, la conservazione non avrà scadenza.
  - Se questo gruppo di flussi di log esiste già e si specifica questo campo, viene utilizzata la nuova conservazione specificata. Se si omette questo campo per un gruppo di flussi di log già esistente, la conservazione del gruppo non viene modificata.

    La procedura guidata dell' CloudWatch agente utilizza `-1` come valore predefinito per questo campo quando viene utilizzato per creare il file di configurazione dell'agente e non si specifica un valore per la conservazione dei log. Questo valore `-1` specificato dalla procedura guidata specifica che gli eventi nel gruppo di log non scadono. Tuttavia, la modifica manuale di questo valore su `-1` non ha alcun effetto.

  I valori validi sono 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288 e 3653.

  Se configuri l'agente in modo che scriva più flussi di log nello stesso gruppo, la specificazione del parametro `retention_in_days` in un'unica posizione imposterà la conservazione dei log per l'intero gruppo. Se definisci il parametro `retention_in_days` per lo stesso gruppo di log in più posizioni, la conservazione viene impostata se tutti i valori sono uguali. Tuttavia, se sono specificati valori del parametro `retention_in_days` diversi per lo stesso gruppo di log in più posizioni, la conservazione dei log non verrà impostata e l'agente si fermerà, restituendo un errore.

> **Note**
>
> Per poter impostare le policy di conservazione, il ruolo o l'utente IAM dell'agente deve disporre della `logs:PutRetentionPolicy`. Per ulteriori informazioni, consulta la pagina Consentire all' CloudWatch agente di impostare una politica di conservazione dei log.

> **Warning**
>
> Se hai impostato la `retention_in_days` per un gruppo di log già esistente, tutti i log del gruppo pubblicati prima del numero di giorni definito verranno eliminati. Ad esempio, se si imposta il valore su 3, sarebbero eliminati tutti i log antecedenti agli ultimi 3 giorni.

- `log_stream_name`: obbligatorio. Specifica il nome del flusso di log predefinito da utilizzare per tutti i log o gli eventi Windows che non dispongono di nomi del flusso di log individuali definiti nel parametro `log_stream_name` all'interno della voce in `collect_list`.
- `endpoint_override`: specifica un endpoint FIPS o un collegamento privato da utilizzare come endpoint in cui l'agente invia i log. Specificando questo campo e impostando un collegamento privato potrai inviare i log a un endpoint Amazon VPC. Per ulteriori informazioni, consulta Cos'è Amazon VPC?.

  Il valore di `endpoint_override` deve essere una stringa che è un URL.

  Ad esempio, la parte seguente della sezione dei log del file di configurazione imposta l'agente per utilizzare un endpoint VPC durante l'invio dei log.

```
{
  "logs": {
    "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.logs.us-
east-1.vpce.amazonaws.com",
   ......
   },
}
```

- `force_flush_interval`: specifica in secondi la quantità massima di tempo in cui i log rimangono nel buffer di memoria prima di essere inviati al server. Indipendentemente

dall'impostazione di questo campo, se la dimensione dei log nel buffer raggiunge 1 MB, i log vengono immediatamente inviati al server. Il valore predefinito è 5.

Se utilizzi l'agente per segnalare parametri ad alta risoluzione in formato dei parametri incorporati e stai impostando allarmi su tali parametri, mantieni questo parametro impostato sul valore predefinito di 5. In caso contrario, i parametri vengono segnalati con un ritardo che può causare allarmi su dati parziali o incompleti.

- `credentials`— Speciifica un ruolo IAM da utilizzare per l'invio dei log a un account diverso. AWS Se specificato, questo campo contiene un parametro, `role_arn`.
  - `role_arn`— specifica l'ARN di un ruolo IAM da utilizzare per l'autenticazione quando si inviano i log a un account diverso. AWS Per ulteriori informazioni, consulta Invio di parametri, log e tracce a un altro account. Se specificato qui, sostituisce il `role_arn` specificato nella sezione `agent` del file di configurazione, se presente.
- `metrics_collected`— Questo campo può contenere sezioni per specificare che l'agente deve raccogliere i log per abilitare casi d'uso come CloudWatch Application Signals e Container Insights con osservabilità migliorata per Amazon EKS.
  - `app_signals`(Facoltativo) Indica che desideri abilitare CloudWatch Application Signals. Per ulteriori informazioni su questa configurazione, consulta. Abilita CloudWatch Application Signals
  - `kubernetes`: questo campo può contenere un parametro `enhanced_container_insights`, che può essere utilizzato per abilitare Approfondimenti sui container con osservabilità migliorata per Amazon EKS.
    - `enhanced_container_insights`: imposta questo valore su `true` per abilitare Approfondimenti sui container con osservabilità migliorata per Amazon EKS. Per ulteriori informazioni, consulta Approfondimenti sui container con osservabilità migliorata per Amazon EKS.
    - `accelerated_compute_metrics`— Imposta questa opzione `false` per disattivare la raccolta dei parametri delle GPU Nvidia sui cluster Amazon EKS. Per ulteriori informazioni, consulta Metriche della GPU NVIDIA.
  - `emf`: per raccogliere i parametri integrati nei log, non è più necessario aggiungere questo campo `emf`. Questo è un campo legacy che specifica che l'agente deve raccogliere i log con formato del parametro integrato. È possibile generare dati di parametro da questi log. Per ulteriori informazioni, consulta la pagina Incorporamento dei parametri nei log.

Di seguito è riportato un esempio di una sezione `logs`.

```
"logs":
   {
       "logs_collected": {
           "files": {
               "collect_list": [
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\amazon-cloudwatch-agent.log",
                       "log_group_name": "amazon-cloudwatch-agent.log",
                       "log_stream_name": "my_log_stream_name_1",
                       "timestamp_format": "%H: %M: %S%y%b%-d"
                   },
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\test.log",
                       "log_group_name": "test.log",
                       "log_stream_name": "my_log_stream_name_2"
                   }
               ]
           },
           "windows_events": {
               "collect_list": [
                   {
                       "event_name": "System",
                       "event_levels": [
                           "INFORMATION",
                           "ERROR"
                       ],
                       "log_group_name": "System",
                       "log_stream_name": "System"
                   },
                   {
                       "event_name": "CustomizedName",
                       "event_levels": [
                           "INFORMATION",
                           "ERROR"
                       ],
                       "log_group_name": "CustomizedLogGroup",
                       "log_stream_name": "CustomizedLogStream"
                   }
               ]
           }
       },
```

```
       "log_stream_name": "my_log_stream_name",
       "metrics_collected": {
           "kubernetes": {
                "enhanced_container_insights": true
      }
    }
  }
```

CloudWatch file di configurazione dell'agente: sezione Traces

Aggiungendo una `traces` sezione al file di configurazione dell' CloudWatch agente, è possibile abilitare CloudWatch Application Signals o raccogliere tracce da X-Ray e dall'SDK di OpenTelemetry strumentazione e inviarle a X-Ray.

> **Important**
>
> Il ruolo IAM dell'agente o l'utente IAM deve disporre della AWSXrayWriteOnlyAccesspolicy per inviare i dati di traccia a X-Ray. Per ulteriori informazioni, consulta Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente.

Per iniziare rapidamente a raccogliere le tracce, puoi aggiungere solo quanto segue al file di configurazione dell' CloudWatch agente.

```
"traces_collected": {
        "xray": {
        },
        "otlp": {
        }
      }
```

Se si aggiunge la sezione precedente al file di configurazione dell' CloudWatch agente e si riavvia l'agente, l'agente inizia a raccogliere le tracce utilizzando le seguenti opzioni e valori predefiniti. Per ulteriori informazioni su questi parametri, consulta le definizioni dei parametri più avanti in questa sezione.

```
"traces_collected": {
        "xray": {
            "bind_address": "127.0.0.1:2000",
            "tcp_proxy": {
```

```
            "bind_address": "127.0.0.1:2000"
        }
    },
    "otlp": {
        "grpc_endpoint": "127.0.0.1:4317",
        "http_endpoint": "127.0.0.1:4318"
    }
}
```

La sezione `traces` può includere i seguenti campi:

- `traces_collected`: obbligatorio se è inclusa la sezione `traces`. Specifica da quali SDK raccogliere le tracce. Include i seguenti campi:
  - `app_signals`: opzionale. Specifica che si desidera abilitare CloudWatch Application Signals. Per ulteriori informazioni su questa configurazione, vedereAbilita CloudWatch Application Signals.
  - `xray`: opzionale. Specifica che desideri raccogliere le tracce dall'SDK X-Ray. Questa sezione può includere i seguenti campi.
    - `bind_address` Facoltativo. Speciifica l'indirizzo UDP dell' CloudWatch agente da utilizzare per ascoltare le tracce X-Ray. Il formato è `ip:port`. Questo indirizzo deve corrispondere all'indirizzo impostato nell'SDK X-Ray.

      Se ometti questo campo, verrà utilizzato il valore predefinito `127.0.0.1:2000`.
    - `tcp_proxy`: opzionale. Configura l'indirizzo di un proxy utilizzato per supportare il campionamento remoto di X-Ray. Per ulteriori informazioni, consulta Configurazione delle regole di campionamento nella console di X-Ray.

      Questa sezione può contenere i seguenti campi.
      - `bind_address`: opzionale. Speciifica l'indirizzo TCP su cui l' CloudWatch agente deve configurare il proxy. Il formato è `ip:port`. Questo indirizzo deve corrispondere all'indirizzo impostato nell'SDK X-Ray.

        Se ometti questo campo, verrà utilizzato il valore predefinito `127.0.0.1:2000`.
- `otlp`: opzionale. Specifica che si desidera raccogliere tracce dall'SDK. OpenTelemetry Per ulteriori informazioni su AWS Distro for OpenTelemetry, consulta AWS Distro for. OpenTelemetry Per ulteriori informazioni su AWS Distro for OpenTelemetry SDK, consulta Introduzione.

  Questa sezione può includere i seguenti campi.

    - `grpc_endpoint` Facoltativo. Speciifica l'indirizzo che l' CloudWatch agente deve utilizzare per ascoltare le OpenTelemetry tracce inviate utilizzando le chiamate di procedura remota gRPC. Il formato è `ip:port`. Questo indirizzo deve corrispondere all'indirizzo impostato per l'esportatore gRPC nell'SDK. OpenTelemetry

      Se ometti questo campo, verrà utilizzato il valore predefinito `127.0.0.1:4317`.
    - `http_endpoint`: opzionale. Speciifica l'indirizzo che l' CloudWatch agente deve utilizzare per ascoltare le tracce OTLP inviate tramite HTTP. Il formato è `ip:port`. Questo indirizzo deve corrispondere all'indirizzo impostato per l'esportatore HTTP nell'SDK. OpenTelemetry

      Se ometti questo campo, verrà utilizzato il valore predefinito `127.0.0.1:4318`.
- `concurrency`: opzionale. Specifica il numero massimo di chiamate simultanee a X-Ray che possono essere utilizzate per caricare tracce. Il valore predefinito è `8`.
- `local_mode`: opzionale. Se `true`, l'agente non raccoglie i metadati delle istanze Amazon EC2. Il valore predefinito è `false`.
- `endpoint_override`: opzionale. Speciifica un endpoint FIPS o un link privato da utilizzare come endpoint in cui l'agente invia le tracce. CloudWatch Specificando questo campo e impostando un collegamento privato potrai inviare le tracce a un endpoint Amazon VPC. Per ulteriori informazioni, consulta Cos'è Amazon VPC?.

  Il valore di `endpoint_override` deve essere una stringa che è un URL.
- `region_override`: opzionale. Specifica la Regione da utilizzare per l'endpoint X-Ray. L' CloudWatch agente invia le tracce a X-Ray nella regione specificata. Se ometti questo campo, l'agente invia i parametri alla Regione in cui si trova l'istanza Amazon EC2.

  Se si specifica una Regione qui, questa avrà la precedenza sull'impostazione del parametro `region` nella sezione `agent` del file di configurazione.
- `proxy_override`: opzionale. Speciifica l'indirizzo del server proxy per l' CloudWatch agente da utilizzare per l'invio di richieste a X-Ray. Il protocollo del server proxy deve essere specificato come parte di questo indirizzo.
- `credentials`— Speciifica un ruolo IAM da utilizzare per l'invio di tracce a un account diverso. AWS Se specificato, questo campo contiene un parametro, `role_arn`.
    - `role_arn`— specifica l'ARN di un ruolo IAM da utilizzare per l'autenticazione quando si inviano tracce a un AWS account diverso. Per ulteriori informazioni, consulta Invio di parametri, log e tracce a un altro account. Se specificato qui, sostituisce il `role_arn` specificato nella sezione `agent` del file di configurazione, se presente.

CloudWatch file di configurazione dell'agente: esempi completi

Di seguito è riportato un esempio di file completo di configurazione CloudWatch dell'agente per un server Linux.

Gli elementi elencati nelle sezioni `measurement` per i parametri che si desidera raccogliere possono specificare il nome parametro completo oppure solo la parte del nome del parametro che verrà aggiunta al tipo della risorsa. Ad esempio, specificando `reads` o `diskio_reads` nella sezione `measurement` della sezione `diskio` si causa la raccolta del parametro `diskio_reads`.

Questo esempio include entrambi i modi di specificare i parametri nella `measurement` sezione.

```
   {
     "agent": {
       "metrics_collection_interval": 10,
       "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
     },
     "metrics": {
       "namespace": "MyCustomNamespace",
       "metrics_collected": {
         "cpu": {
           "resources": [
             "*"
           ],
           "measurement": [
             {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit":
"Percent"},
             {"name": "cpu_usage_nice", "unit": "Percent"},
             "cpu_usage_guest"
           ],
           "totalcpu": false,
           "metrics_collection_interval": 10,
           "append_dimensions": {
             "customized_dimension_key_1": "customized_dimension_value_1",
             "customized_dimension_key_2": "customized_dimension_value_2"
           }
         },
         "disk": {
           "resources": [
             "/",
             "/tmp"
           ],
           "measurement": [
```

```
            {"name": "free", "rename": "DISK_FREE", "unit": "Gigabytes"},
            "total",
            "used"
          ],
           "ignore_file_system_types": [
            "sysfs", "devtmpfs"
          ],
          "metrics_collection_interval": 60,
          "append_dimensions": {
            "customized_dimension_key_3": "customized_dimension_value_3",
            "customized_dimension_key_4": "customized_dimension_value_4"
          }
        },
        "diskio": {
          "resources": [
            "*"
          ],
          "measurement": [
            "reads",
            "writes",
            "read_time",
            "write_time",
            "io_time"
          ],
          "metrics_collection_interval": 60
        },
        "swap": {
          "measurement": [
            "swap_used",
            "swap_free",
            "swap_used_percent"
          ]
        },
        "mem": {
          "measurement": [
            "mem_used",
            "mem_cached",
            "mem_total"
          ],
          "metrics_collection_interval": 1
        },
        "net": {
          "resources": [
            "eth0"
```

```
            ],
            "measurement": [
              "bytes_sent",
              "bytes_recv",
              "drop_in",
              "drop_out"
            ]
          },
          "netstat": {
            "measurement": [
              "tcp_established",
              "tcp_syn_sent",
              "tcp_close"
            ],
            "metrics_collection_interval": 60
          },
          "processes": {
            "measurement": [
              "running",
              "sleeping",
              "dead"
            ]
          }
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
 ["d1"],[]],
        "force_flush_interval" : 30
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "log_stream_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
```

```
              },
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
                "log_group_name": "test.log",
                "log_stream_name": "test.log",
                "timezone": "Local"
              }
            ]
          }
        },
        "log_stream_name": "my_log_stream_name",
        "force_flush_interval" : 15,
        "metrics_collected": {
           "kubernetes": {
                "enhanced_container_insights": true
      }
    }
  }
}
```

Di seguito è riportato un esempio di file di configurazione completo CloudWatch dell'agente per un server che esegue Windows Server.

```
{
      "agent": {
        "metrics_collection_interval": 60,
        "logfile": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\\amazon-
cloudwatch-agent.log"
      },
      "metrics": {
        "namespace": "MyCustomNamespace",
        "metrics_collected": {
          "Processor": {
            "measurement": [
              {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
              "% Interrupt Time",
              "% User Time",
              "% Processor Time"
            ],
            "resources": [
              "*"
            ],
            "append_dimensions": {
```

```
            "customized_dimension_key_1": "customized_dimension_value_1",
            "customized_dimension_key_2": "customized_dimension_value_2"
          }
        },
        "LogicalDisk": {
          "measurement": [
            {"name": "% Idle Time", "unit": "Percent"},
            {"name": "% Disk Read Time", "rename": "DISK_READ"},
            "% Disk Write Time"
          ],
          "resources": [
            "*"
          ]
        },
        "customizedObjectName": {
          "metrics_collection_interval": 60,
          "customizedCounterName": [
            "metric1",
            "metric2"
          ],
          "resources": [
            "customizedInstances"
          ]
        },
        "Memory": {
          "metrics_collection_interval": 5,
          "measurement": [
            "Available Bytes",
            "Cache Faults/sec",
            "Page Faults/sec",
            "Pages/sec"
          ]
        },
        "Network Interface": {
          "metrics_collection_interval": 5,
          "measurement": [
            "Bytes Received/sec",
            "Bytes Sent/sec",
            "Packets Received/sec",
            "Packets Sent/sec"
          ],
          "resources": [
            "*"
          ],
```

```
            "append_dimensions": {
              "customized_dimension_key_3": "customized_dimension_value_3"
            }
          },
          "System": {
            "measurement": [
              "Context Switches/sec",
              "System Calls/sec",
              "Processor Queue Length"
            ]
          }
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
 ["d1"],[]]
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\amazon-cloudwatch-agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\test.log",
                "log_group_name": "test.log",
                "timezone": "Local"
              }
            ]
          },
          "windows_events": {
            "collect_list": [
              {
                "event_name": "System",
                "event_levels": [
```

```
                "INFORMATION",
                "ERROR"
              ],
              "log_group_name": "System",
              "log_stream_name": "System",
              "event_format": "xml"
            },
            {
              "event_name": "CustomizedName",
              "event_levels": [
                "WARNING",
                "ERROR"
              ],
              "log_group_name": "CustomizedLogGroup",
              "log_stream_name": "CustomizedLogStream",
              "event_format": "xml"
            }
          ]
        }
      },
      "log_stream_name": "example_log_stream_name"
    }
  }
```

Salvate il file di configurazione CloudWatch dell'agente manualmente

Se si crea o si modifica manualmente il file di configurazione dell' CloudWatch agente, è possibile assegnargli un nome qualsiasi. Per semplicità nella risoluzione dei problemi, consigliamo di denominarlo `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` su un server Linux e `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json` sui server che eseguono Windows Server. Dopo avere creato il file, potrai copiarlo negli altri server in cui desideri eseguire l'agente.

Caricamento del file di configurazione CloudWatch dell'agente in Systems Manager Parameter Store

Se si prevede di utilizzare l'agente SSM per installare l' CloudWatch agente sui server, dopo aver modificato manualmente il file di configurazione dell' CloudWatch agente, è possibile caricarlo in Systems Manager Parameter Store. Per eseguire questa operazione, utilizza il comando `put-parameter` di Systems Manager.

Per poter archiviare il file in Parameter Store, devi utilizzare un ruolo IAM con autorizzazioni sufficienti. Per ulteriori informazioni, consulta la pagina Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente.

Utilizza il seguente comando, in cui *parameter name* è il nome da utilizzare per questo file in Parameter Store e *configuration_file_pathname* corrisponde al percorso e al nome del file di configurazione modificato.

```
aws ssm put-parameter --name "parameter name" --type "String" --value
 file://configuration_file_pathname
```

Abilita CloudWatch Application Signals

Utilizzate CloudWatch Application Signals per strumentare automaticamente le vostre applicazioni AWS in modo da monitorare le prestazioni delle applicazioni rispetto agli obiettivi aziendali. Application Signals offre una visione unificata e incentrata sulle applicazioni delle applicazioni Java, delle loro dipendenze e dei loro edge. Per ulteriori informazioni, consulta Application Signals.

CloudWatch Application Signals sfrutta l' CloudWatch agente per ricevere metriche e tracce dalle applicazioni con strumentazione automatica, applicare facoltativamente regole per ridurre la cardinalità elevata e quindi pubblicare la telemetria elaborata su. CloudWatch È possibile fornire una configurazione personalizzata all' CloudWatch agente specificamente per Application Signals utilizzando il file di configurazione dell'agente. Innanzitutto, la presenza di una `app_signals` sezione sotto la sezione all'interno della `metrics_collected` sezione del `logs` file di configurazione dell'agente specifica che l' CloudWatch agente riceverà le metriche dalle applicazioni con strumentazione automatica. Allo stesso modo, la presenza di una `app_signals` sezione sotto la `traces_collected` sezione all'interno della `traces` sezione del file di configurazione dell'agente specifica che l' CloudWatch agente è abilitato a ricevere tracce dalle applicazioni con strumentazione automatica. Inoltre, puoi facoltativamente inoltrate regole di configurazione personalizzate per ridurre la pubblicazione di telemetria ad alta cardinalità, come indicato in questa sezione.

- Per i cluster Amazon EKS, quando installi il componente aggiuntivo Amazon CloudWatch Observability EKS, l' CloudWatch agente è abilitato per impostazione predefinita a ricevere sia metriche che tracce dalle tue applicazioni con strumentazione automatica. Se desideri facoltativamente inoltrare regole di configurazione personalizzate, puoi farlo inoltrando una configurazione dell'agente personalizzata al componente aggiuntivo Amazon EKS quando lo crei o lo aggiorni utilizzando una configurazione aggiuntiva, come indicato in (Facoltativo) Configurazione aggiuntiva.

- Per altre piattaforme supportate, tra cui Amazon EC2, è necessario avviare l' CloudWatch agente con una configurazione dell'agente che abiliti Application Signals specificando le `app_signals` sezioni e, facoltativamente, eventuali regole di configurazione personalizzate, come descritto più avanti in questa sezione.

Di seguito è riportata una panoramica dei campi del file di configurazione dell' CloudWatch agente relativi ad Application Signals. CloudWatch

- `logs`
  - `metrics_collected`— Questo campo può contenere sezioni per specificare che l'agente deve raccogliere i log per abilitare casi d'uso come CloudWatch Application Signals e Container Insights con osservabilità migliorata per Amazon EKS.

    > **Note**
    >
    > In precedenza, questa sezione veniva utilizzata anche per specificare che l'agente deve raccogliere i log con formato del parametro incorporato. Queste impostazioni non sono più necessarie.

    - `app_signals`(Facoltativo) Specificate che desiderate abilitare CloudWatch Application Signals a ricevere metriche dalle vostre applicazioni con strumentazione automatica per facilitare i segnali applicativi. CloudWatch
      - `rules` (Facoltativo) Una serie di regole per selezionare parametri e tracce in modo condizionale e applicare azioni per gestire scenari ad alta cardinalità. Ogni regola può contenere i seguenti campi:
        - `rule_name` (Facoltativo) Il nome della regola.
        - `selectors` (Facoltativo) Una serie di corrispondenze tra le dimensioni di parametri e tracce. Ogni selettore deve fornire i campi riportati di seguito:
          - `dimension` Obbligatorio se `selectors` non è vuoto. Questo specifica la dimensione dei parametri e delle tracce da utilizzare come filtro.
          - `match` Obbligatorio se `selectors` non è vuoto. Un modello jolly utilizzato per i valori corrispondenti della dimensione specificata.

- `action` (Facoltativo) L'operazione da applicare ai parametri e alle tracce che corrispondono ai selettori specificati. `action` deve essere una delle seguenti parole chiave.
  - `keep`Speciifica di inviare solo le metriche e le tracce a, se corrisponde a. CloudWatch `selectors`
  - `drop` Specifica di eliminare il parametro e le tracce che corrispondono ai `selectors`.
  - `replace` Specifica di sostituire le dimensioni dei parametri e delle tracce che corrispondono ai `selectors`. Le sostituzioni sono effettuate in base alla sezione `replacements`.
- `replacements` Obbligatorio se `action` è `replace`. Una serie di coppie di dimensioni e valori che verranno applicate a parametri e tracce che corrispondono ai `selectors` specificati quando `action` è `replace`. Ogni sostituzione deve fornire i campi riportati di seguito:
  - `target_dimension` Obbligatorio se `replacements` non è vuoto. Specifica la dimensione che deve essere sostituita.
  - `value` Obbligatorio se `replacements` non è vuoto. Il valore con cui sostituire il valore originale di `target_dimension`.
- `limiter`(Facoltativo) Utilizzate questa sezione per limitare il numero di metriche e dimensioni a cui vengono inviate Application Signals CloudWatch, per ottimizzare i costi.
  - `disabled`(Facoltativo) Se`true`, la funzione di limitazione delle metriche è disabilitata. Il valore predefinito è `false`.
  - `drop_threshold`(Facoltativo) Il numero massimo di metriche distinte per servizio in un intervallo di rotazione che possono essere esportate da un agente. CloudWatch L'impostazione predefinita è 500.
  - `rotation_interval`(Facoltativo) L'intervallo in base al quale il limitatore reimposta i record metrici per il conteggio delle distinzioni. Viene espresso come una stringa con una sequenza di numeri e un suffisso unitario. Le frazioni sono supportate. I suffissi di unità supportati sono`s`,`m`, `h`, e `ms` `us` `ns`

    L'impostazione predefinita è `1h` un'ora.
  - `log_dropped_metrics`(Facoltativo) Speciifica se l'agente deve scrivere i log nei log dell' CloudWatch agente quando le metriche di Application Signals vengono eliminate. Il valore predefinito è `false`.

> **Note**
>
> Per attivare questa registrazione, il `debug` parametro nella `agent` sezione deve essere impostato anche su. `true`

- `traces`
  - `traces_collected`
    - `app_signals` Facoltativo. Specificalo per consentire all' CloudWatch agente di ricevere tracce dalle tue applicazioni con strumentazione automatica per facilitare CloudWatch i segnali applicativi.

> **Note**
>
> Anche se le regole `app_signals` personalizzate sono specificate nella sezione `metrics_collected` contenuta nella sezione `logs`, si applicano implicitamente anche alla sezione `traces_collected`. Lo stesso insieme di regole si applicherà sia ai parametri che alle tracce.

Quando sono presenti più regole con operazioni diverse, queste si applicano nella seguente sequenza: `keep`, poi `drop`, quindi `replace`.

Di seguito è riportato un esempio di file di configurazione completo CloudWatch dell'agente che applica regole personalizzate.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "rules": [
          {
            "rule_name": "keep01",
            "selectors": [
              {
                "dimension": "Service",
                "match": "pet-clinic-frontend"
              },
              {
```

```
                "dimension": "RemoteService",
                "match": "customers-service"
              }
            ],
            "action": "keep"
          },
          {
            "rule_name": "drop01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "GET /api/customer/owners/*"
              }
            ],
            "action": "drop"
          },
          {
            "rule_name": "replace01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "PUT /api/customer/owners/*/pets/*"
              },
              {
                "dimension": "RemoteOperation",
                "match": "PUT /owners"
              }
            ],
            "replacements": [
              {
                "target_dimension": "Operation",
                "value": "PUT /api/customer/owners/{ownerId}/pets{petId}"
              }
            ],
            "action": "replace"
          }
        ]
      }
    }
  },
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
```

```
  }
}
```

Per il file di configurazione di esempio precedente, le `rules` vengono elaborate come segue:

1. La regola `keep01` assicura che tutte i parametri e le tracce con la dimensione `Service` come `pet-clinic-frontend` e la dimensione `RemoteService` come `customers-service` vengano mantenute.
2. Per i parametri e le tracce elaborati dopo l'applicazione di `keep01`, la regola `drop01` garantisce che i parametri e le tracce con la dimensione `Operation` come `GET /api/customer/owners/*` vengano eliminati.
3. Per i parametri e le tracce elaborati dopo l'applicazione di `drop01`, la regola `replace01` aggiorna i parametri e le tracce che hanno la dimensione `Operation` come `PUT /api/customer/owners/*/pets/*` e la dimensione `RemoteOperation` come `PUT /owners` in modo che la dimensione `Operation` venga ora sostituita da `PUT /api/customer/owners/{ownerId}/pets{petId}`.

Di seguito è riportato un esempio completo di file di CloudWatch configurazione che gestisce la cardinalità in Application Signals modificando il limite delle metriche a 100, abilitando la registrazione delle metriche eliminate e impostando l'intervallo di rotazione su due ore.

```
{
    "logs": {
        "metrics_collected": {
            "app_signals": {
                "limiter": {
                    "disabled": false,
                    "drop_threshold": 100,
                    "rotation_interval": "2h",
                    "log_dropped_metrics": true
                }
            }
        },
        "traces": {
            "traces_collected": {
                "app_signals": {}
            }
        }
    }
```

```
}
```

Raccolta di parametri sulle prestazioni di rete

Le istanze EC2 in esecuzione su Linux che utilizzano Elastic Network Adapter (ENA) pubblicano parametri delle prestazioni di rete. La versione 1.246396.0 e successive dell' CloudWatch agente consentono di importare queste metriche delle prestazioni di rete in. CloudWatch Quando si importano queste metriche delle prestazioni di rete in CloudWatch, vengono addebitate come metriche personalizzate. CloudWatch

Per ulteriori informazioni sul driver ENA, consulta Abilitazione delle reti avanzate con Elastic Network Adapter (ENA) sulle istanze Linux e Abilitazione delle reti avanzate con Elastic Network Adapter (ENA) sulle istanze Windows.

La modalità di impostazione della raccolta dei parametri delle prestazioni di rete varia nei server Linux e nei server Windows.

Nella tabella seguente sono elencati i parametri delle prestazioni di rete abilitati dall'adattatore ENA. Quando l' CloudWatch agente importa queste metriche CloudWatch da istanze Linux, queste vengono anteposte all'inizio di ciascuna `ethtool_` di queste metriche.

Parametro	Descrizione
Nome su server Linux: **`bw_in_allowance_exceeded`** Nome su server Windows: **`Aggregate inbound BW allowance exceeded`**	Il numero di pacchetti accordati e/o rilasciati perché la larghezza di banda aggregata in ingresso ha superato il valore massimo per l'istanza. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione dell'`metrics_collected` agente. CloudWatch Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna
Nome su server Linux: **`bw_out_allowance_exceeded`**	Il numero di pacchetti accodati e/o rilasciati perché la larghezza di banda aggregata in uscita ha superato il valore massimo per l'istanza.

Parametro	Descrizione
Nome su server Windows: **`Aggregate outbound BW allowance exceeded`**	Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione `metrics_collected` dell' CloudWatch agente. Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna
Nome su server Linux: **`conntrack_allowance_available`** Nome su server Windows: **`Available connection tracking allowance`**	Riporta il numero di connessioni tracciate che possono essere stabilite dall'istanza prima di raggiungere il limite Connessioni tracciate di quel tipo di istanza. Questa metrica è disponibile solo sulle istanze EC2 basate su Nitro che utilizzano il driver Linux per Elastic Network Adapter (ENA) a partire dalla versione 2.8.1 e sui computer che utilizzano il driver Windows per Elastic Network Adapter (ENA) a partire dalla versione 2.6.0. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione dell'agente. `metrics_collected` CloudWatch Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna

Parametro	Descrizione
Nome su server Linux: **ena_srd_mode** Nome su server Windows: **ena srd mode**	Descrive quali funzionalità di ENA Express sono abilitate. Per ulteriori informazioni su ENA Express, consulta Migliorare le prestazioni di rete con ENA Express sulle istanze Linux I valori sono i seguenti: • 0 = ENA Express disattivato, UDP disattivato • 1 = ENA Express attivato, UDP disattivato • 2 = ENA Express disattivato, UDP attivato **Note** Ciò accade solo quando ENA Express era originariamente abilitato e UDP era configurato per utilizzarlo. Il valore precedente viene mantenuto per il traffico UDP. • 3 = ENA Express attivato, UDP attivato
Nome su server Linux: **ena_srd_eligible_tx_pkts** Nome su server Windows: **ena srd eligible tx pkts**	Il numero di pacchetti di rete inviati in un determinato periodo di tempo che soddisfano i requisiti di idoneità dello AWS Scalable Reliable Datagram (SRD), come segue: • Sono supportati i tipi sia delle istanze di invio sia di quelle di ricezione. • Sia le istanze di invio sia quelle di ricezione devono avere ENA Express configurato. • Le istanze di invio e ricezione devono trovarsi sulla stessa sottorete. • Il percorso di rete tra le istanze non deve includere box middleware (software intermediario). ENA Express attualmente non supporta i box middlewar e (software intermediario).

Parametro	Descrizione
Nome su server Linux: **ena_srd_tx_pkts** Nome su server Windows: **ena srd tx pkts**	Il numero di pacchetti SRD trasmessi in un determinato periodo di tempo.
Nome su server Linux: **ena_srd_rx_pkts** Nome su server Windows: **ena srd rx pkts**	Il numero di pacchetti SRD ricevuti in un determinato periodo di tempo.
Nome su server Linux: **ena_srd_resource_utilization** Nome su server Windows: **ena srd resource utilization**	La percentuale di utilizzo massimo consentito della memoria per le connessioni SRD simultanee utilizzate dall'istanza.
Nome su server Linux: **linklocal_allowance_exceeded** Nome su server Windows: **Link local packet rate allowance exceeded**	Il numero di pacchetti accodati o rilasciati perché il PPS del traffico verso i servizi proxy locali ha superato il valore massimo per l'interfaccia di rete. Ciò influisce sul traffico verso il servizio DNS, il servizio di metadati dell'istanza e il servizio Amazon Time Sync. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione dell'`metrics_collected` agente. CloudWatch Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna

Parametro	Descrizione
Nome su server Linux: **`linklocal_allowance_exceeded`** Nome su server Windows: **`Link local packet rate allowance exceeded`**	Il numero di pacchetti accodati o rilasciati perché il PPS del traffico verso i servizi proxy locali ha superato il valore massimo per l'interfaccia di rete. Ciò influisce sul traffico verso il servizio DNS, il servizio di metadati dell'istanza e il servizio Amazon Time Sync. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione `metrics_collected` dell' CloudWatch agente. Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna
Nome su server Linux: **`pps_allowance_exceeded`** Nome su server Windows: **`PPS allowance exceeded`**	Il numero di pacchetti accodati e/o rilasciati perché il PPS bidirezionale ha superato il valore massimo per l'istanza. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione `metrics_collected` dell' CloudWatch agente. Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna

Configurazione di Linux

Sui server Linux, il plugin ethtool consente di importare le metriche delle prestazioni di rete in. CloudWatch

ethtool è un'utility Linux standard in grado di raccogliere statistiche sui dispositivi Ethernet sui server Linux. Le statistiche raccolte dipendono dal dispositivo di rete e dal driver. Esempi di queste

statistiche includono `tx_cnt`, `rx_bytes`, `tx_errors` e `align_errors`. Quando utilizzi il plugin ethtool con l' CloudWatch agente, puoi anche importare queste statistiche CloudWatch, insieme alle metriche delle prestazioni di rete EC2 elencate in precedenza in questa sezione.

> **Tip**
>
> Per trovare le statistiche disponibili sul nostro sistema operativo e dispositivo di rete, usa il comando `ethtool -S`.

Quando l' CloudWatch agente importa le metriche in CloudWatch, aggiunge un `ethtool_` prefisso ai nomi di tutte le metriche importate. Quindi viene richiamata la statistica standard di ethtool e `rx_bytes` viene richiamata `ethtool_rx_bytes` la metrica delle prestazioni di rete EC2. CloudWatch b`w_in_allowance_exceeded` `ethtool_bw_in_allowance_exceeded` CloudWatch

Sui server Linux, per importare le metriche ethtool, aggiungi una `ethtool` sezione alla sezione del file di configurazione dell'agente`metrics_collected`. CloudWatch La sezione `ethtool` può includere le seguenti sottosezioni:

- interface_include: l'inclusione di questa sezione fa sì che l'agente raccolga i parametri solo dalle interfacce con i nomi elencati in questa sezione. Se si omette questa sezione, i parametri vengono raccolti da tutte le interfacce Ethernet non elencate in `interface_exclude`.

  L'interfaccia ethernet predefinita è `eth0`.
- interface_exclude: se si include questa sezione, elencare le interfacce Ethernet da cui non si desidera raccogliere i parametri.

  Il plug-in ethtool ignora sempre le interfacce di loopback.
- metrics_include — Questa sezione elenca le metriche in cui importare. CloudWatch Può includere sia le statistiche standard raccolte da ethtool che i parametri di rete ad alta risoluzione di Amazon EC2.

L'esempio seguente visualizza parte del file di configurazione dell'agente. CloudWatch Questa configurazione raccoglie i parametri standard ethtool `rx_packets` e `tx_packets` e i parametri sulle prestazioni rete Amazon EC2 provenienti solo dall'interfaccia `eth1`.

Per ulteriori informazioni sul file di configurazione CloudWatch dell'agente, vedere Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.

```
"metrics": {
    "append_dimensions": {
      "InstanceId": "${aws:InstanceId}"
    },
    "metrics_collected": {
      "ethtool": {
        "interface_include": [
          "eth1"
        ],
        "metrics_include": [
          "rx_packets",
          "tx_packets",
          "bw_in_allowance_exceeded",
          "bw_out_allowance_exceeded",
          "conntrack_allowance_exceeded",
          "linklocal_allowance_exceeded",
          "pps_allowance_exceeded"
         ]
      }
   }
}
```

Configurazione su Windows

Sui server Windows, le metriche delle prestazioni di rete sono disponibili tramite Windows Performance Counters, da cui l' CloudWatch agente raccoglie già le metriche. Non è quindi necessario alcun plug-in per raccogliere queste metriche dai server Windows.

Di seguito è riportato un file di configurazione di esempio per raccogliere le metriche delle prestazioni di rete da Windows. Per ulteriori informazioni sulla modifica del file di configurazione dell' CloudWatch agente, vedere. Crea o modifica manualmente il file di configurazione dell' CloudWatch agente

```
{
    "metrics": {
        "append_dimensions": {
            "InstanceId": "${aws:InstanceId}"
        },
        "metrics_collected": {
            "ENA Packets Shaping": {
```

```
                "measurement": [
                    "Aggregate inbound BW allowance exceeded",
                    "Aggregate outbound BW allowance exceeded",
                    "Connection tracking allowance exceeded",
                    "Link local packet rate allowance exceeded",
                    "PPS allowance exceeded"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            }
        }
    }
}
```

Visualizzazione dei parametri sulle prestazioni di rete

Dopo aver importato le metriche delle prestazioni di rete in CloudWatch, puoi visualizzare queste metriche come grafici di serie temporali e creare allarmi in grado di monitorare queste metriche e avvisarti se superano una soglia specificata. La procedura seguente mostra come visualizzare i parametri ethtool come grafici di serie temporali. Per ulteriori informazioni sull'impostazione degli allarmi, consulta Utilizzo degli CloudWatch allarmi Amazon.

Poiché tutte queste metriche sono contatori aggregati, puoi utilizzare funzioni matematiche metriche, ad esempio per calcolare la frequenza di queste CloudWatch metriche nei grafici o `RATE(METRICS())` utilizzarle per impostare allarmi. Per ulteriori informazioni sulle funzioni matematiche dei parametri, consulta Utilizzare la matematica dei parametri

Per visualizzare le metriche delle prestazioni di rete nella console CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Scegli lo spazio dei nomi da utilizzare per i parametri raccolti dall'agente. Per impostazione predefinita, si tratta di CWAgent, ma è possibile che sia stato specificato uno spazio dei nomi diverso nel file di configurazione dell' CloudWatch agente.
4. Scegli una dimensione di parametro (ad esempio, Per-Instance Metrics (Parametri per istanza)).
5. La scheda All metrics (Tutti i parametri) visualizza tutti i parametri per tale dimensione nello spazio dei nomi. Puoi eseguire le operazioni indicate di seguito:

a. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.

b. Per ordinare la tabella, utilizza l'intestazione della colonna.

c. Per filtrare in base a una risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).

d. Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).

6. (Facoltativo) Per aggiungere questo grafico a una CloudWatch dashboard, scegli Azioni, quindi scegli Aggiungi alla dashboard.

Raccolta dei parametri della GPU NVIDIA

È possibile utilizzare l' CloudWatch agente per raccogliere i parametri della GPU NVIDIA dai server Linux. Per configurarlo, aggiungi una `nvidia_gpu` sezione all'interno della `metrics_collected` sezione del file di configurazione dell' CloudWatch agente. Per ulteriori informazioni, consulta Sezione Linux.

Inoltre, sull'istanza deve essere installato un driver NVIDIA. I driver NVIDIA sono preinstallati su alcune Amazon Machine Image (AMI). In caso contrario, il driver può essere installato manualmente. Per ulteriori informazioni, consulta Installazione dei driver NVIDIA sulle istanze Linux.

È possibile raccogliere i seguenti parametri. Tutte queste metriche vengono raccolte senza CloudWatch `Unit`, ma è possibile specificare un'unità per ogni metrica aggiungendo un parametro al file di configurazione dell' CloudWatch agente. Per ulteriori informazioni, consulta Sezione Linux.

Parametro	Nome della metrica in CloudWatch	Descrizione
`utilization_gpu`	`nvidia_smi_utilization_gpu`	La percentuale di tempo nell'ultimo periodo di campionamento in cui erano in esecuzione uno o più kernel sulla GPU.
`temperature_gpu`	`nvidia_smi_temperature_gpu`	La temperatura del core della GPU in gradi Celsius.

Parametro	Nome della metrica in CloudWatch	Descrizione
power_draw	nvidia_smi_power_draw	L'ultimo assorbimento di potenza misurato per l'intera scheda, in watt.
utilization_memory	nvidia_smi_utilization_memory	La percentuale di tempo nell'ultimo periodo di campionamento in cui la memoria globale (dispositivo) veniva letta o scritta.
fan_speed	nvidia_smi_fan_speed	La percentuale di velocità massima attualmente prevista per il funzionamento della ventola del dispositivo.
memory_total	nvidia_smi_memory_total	Memoria totale riportata, in MB.
memory_used	nvidia_smi_memory_used	Memoria utilizzata, in MB.
memory_free	nvidia_smi_memory_free	Memoria libera, in MB.
pcie_link_gen_current	nvidia_smi_pcie_link_gen_current	L'attuale generazione del collegamento.
pcie_link_width_current	nvidia_smi_pcie_link_width_current	L'attuale larghezza del collegamento.
encoder_stats_session_count	nvidia_smi_encoder_stats_session_count	Il numero attuale di sessioni dell'encoder.
encoder_stats_average_fps	nvidia_smi_encoder_stats_average_fps	La media mobile dei fotogrammi di codifica al secondo.

Parametro	Nome della metrica in CloudWatch	Descrizione
`encoder_stats_average_latency`	`nvidia_smi_encoder_stats_average_latency`	La media mobile della latenza di codifica in microsecondi.
`clocks_current_graphics`	`nvidia_smi_clocks_current_graphics`	L'attuale frequenza di clock della scheda video (shader).
`clocks_current_sm`	`nvidia_smi_clocks_current_sm`	L'attuale frequenza di clock dello Streaming Multiprocessor (SM).
`clocks_current_memory`	`nvidia_smi_clocks_current_memory`	L'attuale frequenza di clock della memoria.
`clocks_current_video`	`nvidia_smi_clocks_current_video`	L'attuale frequenza di clock del video (encoder più decoder).

Tutti questi parametri vengono raccolti con le seguenti dimensioni:

Dimensione	Descrizione
`index`	Un identificatore univoco per la GPU su questo server. Rappresenta l'indice NVIDIA Management Library (NVML) del dispositivo.

Dimensione	Descrizione
`name`	Il tipo di GPU. Ad esempio, `NVIDIA Tesla A100`
`host`	Il nome host del server.

Raccolta di parametri dei processi con il plug-in procstat

Il plug-in procstat consente di recuperare i parametri dai singoli processi. È supportato su server Linux e su server che eseguono una versione supportata di Windows Server.

Argomenti

- Configurazione dell' CloudWatch agente per procstat
- Parametri raccolti da Procstat
- Visualizzazione delle metriche di processo importate dall'agente CloudWatch

Configurazione dell' CloudWatch agente per procstat

Per utilizzare il plugin procstat, aggiungi una `procstat` sezione nella sezione del file di configurazione dell'`metrics_collected`agente. CloudWatch Esistono tre modi per specificare i processi da monitorare. Puoi usare solo uno di questi metodi, ma puoi specificare uno o più processi da monitorare.

- `pid_file`: consente di selezionare i processi in base ai nomi dei file PID creati.
- `exe`: consente di selezionare i processi i cui nomi corrispondono alla stringa specificata, utilizzando regole di corrispondenza delle espressioni regolari. La corrispondenza è di tipo "contiene", ovvero se si specifica `agent` come il termine da abbinare, i processi con nomi come `cloudwatchagent` corrispondono al termine. Per ulteriori informazioni, consulta la pagina Syntax.
- `pattern`: consente di selezionare i processi in base alla righe di comando utilizzate per avviare i processi. Le righe di comando di tutti i processi selezionati corrispondono alla stringa specificata utilizzando regole di corrispondenza delle espressioni regolari. L'intera riga di comando è selezionata, inclusi i parametri e le opzioni utilizzati con il comando.

La corrispondenza è di tipo "contiene", ovvero se si specifica `-config` come il termine da abbinare, i processi con parametri come `-c` corrispondono al termine.

- `drop_original_metrics`: opzionale. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non vuoi che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.

L' CloudWatch agente utilizza solo uno di questi metodi, anche se includi più di una delle sezioni precedenti. Se si specifica più di una sezione, l' CloudWatch agente utilizza la `pid_file` sezione se presente. In caso contrario, utilizza la sezione `exe`.

Su Linux server, le stringhe specificate in una sezione `exe` o `pattern` vengono valutate come espressioni regolari. Su server che eseguono Windows Server, queste stringhe vengono valutate come query WMI. Ad esempio, sarebbe `pattern: "%apache%"`. Per ulteriori informazioni, consulta la pagina LIKE Operator.

Indipendentemente dal metodo utilizzato, puoi includere un parametro `metrics_collection_interval` opzionale, che specifica la frequenza in secondi di raccolta di tali parametri. Se non specifichi questo parametro, verrà utilizzato il valore predefinito di 60 secondi.

Negli esempi nelle sezioni seguenti, la sezione `procstat` è la sola inclusa nella sezione `metrics_collected` del file di configurazione dell'agente. I file di configurazione effettivi possono anche includere altre sezioni in `metrics_collected`. Per ulteriori informazioni, consulta la pagina Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.

Configurazione con pid_file

La sezione `procstat` di esempio seguente monitora i processi che creano i file PID `example1.pid` e `example2.pid`. Ogni processo raccogli parametri diversi. I parametri raccolti dal processo che crea `example2.pid` vengono raccolti ogni 10 secondi, mentre i parametri raccolti dal processo `example1.pid` vengono raccolti ogni 60 secondi, ovvero il valore predefinito.

```
{
    "metrics": {
        "metrics_collected": {
```

```
            "procstat": [
                {
                    "pid_file": "/var/run/example1.pid",
                    "measurement": [
                        "cpu_usage",
                        "memory_rss"
                    ]
                },
                {
                    "pid_file": "/var/run/example2.pid",
                    "measurement": [
                        "read_bytes",
                        "read_count",
                        "write_bytes"
                    ],
                    "metrics_collection_interval": 10
                }
            ]
        }
    }
}
```

Configurazione con exe

La sezione `procstat` di esempio seguente monitora tutti i processi con nomi che corrispondono alle stringhe `agent` o `plugin`. Ogni processo raccoglie gli stessi parametri.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "exe": "agent",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                },
                {
                    "exe": "plugin",
                    "measurement": [
                        "cpu_time",
```

```
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                }
            ]
        }
    }
}
```

Configurazione con modello

La sezione `procstat` di esempio seguente monitora tutti i processi con righe di comando che corrispondono alle stringhe `config` o `-c`. Ogni processo raccoglie gli stessi parametri.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pattern": "config",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                },
                {
                    "pattern": "-c",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                }
            ]
        }
    }
}
```

Parametri raccolti da Procstat

La tabella seguente elenca i parametri che puoi raccogliere con il plug-in `procstat`.

L' CloudWatch agente aggiunge `procstat` all'inizio dei seguenti nomi di metriche. La sintassi varia a seconda che la raccolta venga eseguita da un server Linux o da un server che esegue Windows Server. Ad esempio, il parametro `cpu_time` viene visualizzato come `procstat_cpu_time` quando viene raccolto da Linux e come `procstat cpu_time` quando viene raccolto da Windows Server.

Nome parametro	Disponibile su	Descrizione
`cpu_time`	Linux	Il tempo di utilizzo della CPU. Questo parametro è misurato in centesimi di secondo. Unità: numero
`cpu_time_guest`	Linux	Il periodo di tempo durante il quale il processo si trova in modalità guest. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna
`cpu_time_guest_nice`	Linux	Il tempo in cui il processo è

Nome parametro	Disponibile su	Descrizione
		in esecuzione su un host nice. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna
`cpu_time_idle`	Linux	Il periodo di tempo durante il quale il processo si trova in modalità di inattività. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`cpu_time_iowait`	Linux	Il periodo di tempo durante il quale il processo è in attesa di operazioni di I/O da completare. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna
`cpu_time_irq`	Linux	Il periodo di tempo durante il quale il processo lavora sulle interruzioni. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`cpu_time_nice`	Linux	Il periodo di tempo durante il quale il processo si trova in modalità nice. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna
`cpu_time_soft_irq`	Linux	Il periodo di tempo durante il quale il processo lavora sulle interruzioni del software. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`cpu_time_steal`	Linux	Il tempo impiegato per l'esecuzione in altri sistemi operativi durante l'esecuzione in un ambiente virtualizzato. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`cpu_time_stolen`	Linux, Windows Server	Il periodo di tempo durante il quale il processo si trova nello stato di stolen time (tempo rubato), ovvero il tempo impiegato su altri sistemi operativi in un ambiente virtualizzato. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`cpu_time_system`	Linux, Windows Server, macOS	Il periodo di tempo durante il quale il processo si trova in modalità di sistema. Questo parametro è misurato in centesimi di secondo. Tipo: Float Unità: numero
`cpu_time_user`	Linux, Windows Server, macOS	Il periodo di tempo durante il quale il processo si trova in modalità utente. Questo parametro è misurato in centesimi di secondo. Unità: numero

Nome parametro	Disponibile su	Descrizione
`cpu_usage`	Linux, Windows Server, macOS	La percentuale di tempo durante il quale il processo è attivo su qualsiasi capacità. Unità: percentuale
`memory_data`	Linux, macOS	La quantità di memoria utilizzata dal processo per i dati. Unità: byte
`memory_locked`	Linux, macOS	La quantità di memoria bloccata dal processo. Unità: byte
`memory_rss`	Linux, Windows Server, macOS	La quantità di memoria reale (set residente) utilizzata dal processo. Unità: byte

Nome parametro	Disponibile su	Descrizione
`memory_stack`	Linux, macOS	La quantità di memoria stack utilizzata dal processo. Unità: byte
`memory_swap`	Linux, macOS	La quantità di memoria swap utilizzata dal processo. Unità: byte
`memory_vms`	Linux, Windows Server, macOS	La quantità di memoria virtuale utilizzata dal processo. Unità: byte
`num_fds`	Linux	Il numero di descrittori di file aperti da questo processo. Unità: nessuna
`num_threads`	Linux, Windows, macOS	Il numero di thread nel processo. Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`pid`	Linux, Windows Server, macOS	Identificatore processo (ID). Unità: nessuna
`pid_count`	Linux, Windows Server, macOS	Il numero di ID processo associati al processo. Sui server Linux e sui computer macOS il nome completo di questa metrica è `procstat_lookup_pid_count` e su Windows Server è `procstat_lookup pid_count` . Unità: nessuna
`read_bytes`	Linux, Windows Server	Il numero di byte letti dal processo dai dischi. Unità: byte

Nome parametro	Disponibile su	Descrizione
`write_bytes`	Linux, Windows Server	Il numero di byte scritti dal processo sui dischi. Unità: byte
`read_count`	Linux, Windows Server	Il numero di operazioni di lettura del disco eseguite dal processo. Unità: nessuna
`rlimit_realtime_priority_hard`	Linux	Il limite hard alla priorità in tempo reale che può essere impostato per questo processo. Unità: nessuna
`rlimit_realtime_priority_soft`	Linux	Il limite soft alla priorità in tempo reale che può essere impostato per questo processo. Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`rlimit_signals_pending_hard`	Linux	Il limite hard al numero massimo di segnali che possono essere messi in coda da questo processo. Unità: nessuna
`rlimit_signals_pending_soft`	Linux	Il limite soft al numero massimo di segnali che possono essere messi in coda da questo processo. Unità: nessuna
`rlimit_nice_priority_hard`	Linux	Il limite hard alla priorità nice massima che può essere impostata da questo processo. Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`rlimit_nice_priority_soft`	Linux	Il limite soft alla priorità nice massima che può essere impostata da questo processo. Unità: nessuna
`rlimit_num_fds_hard`	Linux	Il limite del numero massimo di descrittori di file che un processo può avere aperti. Unità: nessuna
`rlimit_num_fds_soft`	Linux	Il limite soft del numero massimo di descrittori di file che un processo può avere aperti. Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`write_count`	Linux, Windows Server	Il numero di operazioni di scrittura del disco eseguite dal processo. Unità: nessuna
`involuntary_context_switches`	Linux	Il numero di volte in cui si è verificato un passaggio di contesto involontario del processo. Unità: nessuna
`voluntary_context_switches`	Linux	Il numero di volte in cui si è verificato un passaggio di contesto volontario del processo. Unità: nessuna
`realtime_priority`	Linux	L'utilizzo corrente di priorità real-time per il processo. Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`nice_priority`	Linux	L'utilizzo corrente di priorità nice per il processo. Unità: nessuna
`signals_pending`	Linux	Il numero di segnali in attesa di essere gestiti dal processo. Unità: nessuna
`rlimit_cpu_time_hard`	Linux	Il limite di risorsa di tempo CPU hard per il processo. Unità: nessuna
`rlimit_cpu_time_soft`	Linux	Il limite di risorsa di tempo CPU soft per il processo. Unità: nessuna

Nome parametro	Disponibile su	Descrizione
`rlimit_file_locks_hard`	Linux	Il limite di risorsa dei blocchi file hard per il processo. Unità: nessuna
`rlimit_file_locks_soft`	Linux	Il limite di risorsa dei blocchi file soft per il processo. Unità: nessuna
`rlimit_memory_data_hard`	Linux	Il limite di risorse hard sul processo per memoria utilizzata per i dati. Unità: byte
`rlimit_memory_data_soft`	Linux	Il limite di risorse soft sul processo per memoria utilizzata per i dati. Unità: byte

Nome parametro	Disponibile su	Descrizione
`rlimit_memory_locked_hard`	Linux	Il limite di risorse hard sul processo per memoria bloccata. Unità: byte
`rlimit_memory_locked_soft`	Linux	Il limite di risorse soft sul processo per memoria bloccata. Unità: byte
`rlimit_memory_rss_hard`	Linux	Il limite di risorse hard sul processo per memoria fisica. Unità: byte
`rlimit_memory_rss_soft`	Linux	Il limite di risorse soft sul processo per memoria fisica. Unità: byte
`rlimit_memory_stack_hard`	Linux	Il limite di risorsa hard sullo stack del processo. Unità: byte

Nome parametro	Disponibile su	Descrizione
`rlimit_memory_stack_soft`	Linux	Il limite di risorsa soft sullo stack del processo. Unità: byte
`rlimit_memory_vms_hard`	Linux	Il limite di risorse hard sul processo per memoria virtuale. Unità: byte
`rlimit_memory_vms_soft`	Linux	Il limite di risorse soft sul processo per memoria virtuale. Unità: byte

Visualizzazione delle metriche di processo importate dall'agente CloudWatch

Dopo aver importato le metriche di processo in CloudWatch, puoi visualizzare queste metriche come grafici di serie temporali e creare allarmi in grado di monitorare queste metriche e avvisarti se superano una soglia specificata. La procedura seguente mostra come visualizzare i parametri del processo come grafico delle serie temporali. Per ulteriori informazioni sull'impostazione degli allarmi, consulta Utilizzo degli CloudWatch allarmi Amazon.

Per visualizzare le metriche di processo nella console CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.

3. Scegli lo spazio dei nomi da utilizzare per i parametri raccolti dall'agente. Per impostazione predefinita, si tratta di CWAgent, ma è possibile che sia stato specificato uno spazio dei nomi diverso nel file di configurazione dell' CloudWatch agente.
4. Scegli una dimensione di parametro (ad esempio, Per-Instance Metrics (Parametri per istanza)).
5. La scheda All metrics (Tutti i parametri) visualizza tutti i parametri per tale dimensione nello spazio dei nomi. Puoi eseguire le operazioni indicate di seguito:

   a. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.

   b. Per ordinare la tabella, utilizza l'intestazione della colonna.

   c. Per filtrare per risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).

   d. Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).
6. (Facoltativo) Per aggiungere questo grafico a un CloudWatch pannello di controllo, scegli Azioni, Aggiungi al pannello di controllo.

Recupero dei parametri personalizzati con StatsD

Puoi recuperare metriche personalizzate aggiuntive dalle tue applicazioni o servizi utilizzando l' CloudWatchagente con il protocollo. `StatsD` StatSD è una soluzione open source popolare in grado di raccogliere parametri da un'ampia varietà di applicazioni. StatSD è particolarmente utile per la strumentazione dei parametri. Per un esempio di utilizzo congiunto dell' CloudWatch agente e StatSD, consulta Come monitorare meglio i parametri delle applicazioni personalizzate utilizzando Amazon Agent. CloudWatch

`StatsD`è supportato sia su server Linux che su server che eseguono Windows Server. CloudWatch supporta il seguente `StatsD` formato:

```
MetricName:value|type|@sample_rate|#tag1:
  value,tag1...
```

- `MetricName`: una stringa senza virgola, barre, caratteri # o @.
- `value`: questo può essere un numero intero o float.
- `type`: specifica `c` per il contatore, `g` per il misuratore, `ms` per il timer, `h` per l'istogramma o `s` per il set.

- `sample_rate` (facoltativo) Un float compreso tra 0 e 1, inclusi. Utilizza solo per contatore, istogramma e parametri timer. Il valore predefinito è 1 (campionamento 100% del tempo).
- `tags`— (Facoltativo) Un elenco di tag separati da virgole. `StatsD`i tag sono simili alle dimensioni di. CloudWatch Utilizza i due punti per tag chiave/valore, ad esempio `env:prod`.

Puoi utilizzare qualsiasi `StatsD` client che segue questo formato per inviare le metriche all' CloudWatch agente. Per ulteriori informazioni su alcuni dei `StatsD` client disponibili, vedere la pagina del client StatSD su. GitHub

Per raccogliere questi parametri personalizzati, aggiungi una riga `"statsd": {}` alla sezione `metrics_collected` del file di configurazione dell'agente. È possibile aggiungere questa riga manualmente. Se utilizzi la procedura guidata per creare il file di configurazione, è già tutto previsto. Per ulteriori informazioni, consulta la pagina Creare il file di configurazione CloudWatch dell'agente.

La configurazione predefinita `StatsD` funziona per la maggior parte degli utenti. È possibile aggiungere dei campi facoltativi alla sezione statsd del file di configurazione dell'agente in base alle esigenze:

- `service_address`— L'indirizzo del servizio che l' CloudWatch agente deve ascoltare. Il formato è *`ip`*`:`*`port`*. Se ometti l'indirizzo IP, l'agente ascolta su tutte le interfacce disponibili. È supportato solo il formato UDP, perciò non devi specificare un prefisso UDP.

  Il valore predefinito è `:8125`.
- `metrics_collection_interval`: la frequenza in secondi con cui il plug-in `StatsD` viene eseguito e raccoglie i parametri. Il valore predefinito è 10 secondi. L'intervallo varia tra 1 e 172.000.
- `metrics_aggregation_interval`— Con che frequenza, in secondi, CloudWatch aggrega le metriche in singoli punti dati. Il valore predefinito è 60 secondi.

  Ad esempio, se `metrics_collection_interval` è 10 ed `metrics_aggregation_interval` è 60, CloudWatch raccoglie i dati ogni 10 secondi. Dopo ogni minuto, le sei letture dei dati di quel minuto vengono aggregate in un unico punto dati, che viene inviato a. CloudWatch

  L'intervallo varia tra 0 e 172.000. Impostando `metrics_aggregation_interval` su 0 si disabilita l'aggregazione dei parametri `StatsD`.
- `allowed_pending_messages`: il numero di messaggi UDP che possono essere messi in coda. Quando la coda è piena, il server StatsD inizia a eliminare i pacchetti. Il valore predefinito è 10000.

- `drop_original_metrics`: opzionale. Se utilizzi il campo `aggregation_dimensions` nella sezione `metrics` per raggruppare i parametri in risultati aggregati, per impostazione predefinita l'agente invia sia i parametri aggregati che i parametri originali separati per ogni valore della dimensione. Se non desideri che le metriche originali vengano inviate a CloudWatch, puoi specificare questo parametro con un elenco di metriche. Le metriche specificate insieme a questo parametro non hanno le relative metriche per dimensione riportate. CloudWatch Vengono invece riportati solo i parametri aggregati. Ciò riduce il numero di parametri raccolti dall'agente, riducendo i costi.

Di seguito è riportato un esempio della sezione statsd del file di configurazione dell'agente, utilizzando la porta di default e intervalli personalizzati di raccolta e aggregazione.

```
{
   "metrics":{
      "metrics_collected":{
         "statsd":{
            "service_address":":8125",
            "metrics_collection_interval":60,
            "metrics_aggregation_interval":300
         }
      }
   }
}
```

Visualizzazione delle metriche StatsD importate dall'agente CloudWatch

Dopo aver importato le metriche StatsD CloudWatch in, puoi visualizzare queste metriche come grafici di serie temporali e creare allarmi in grado di guardare queste metriche e avvisarti se superano una soglia specificata. La procedura seguente mostra come visualizzare le parametri StatsD come grafico di serie temporali. Per ulteriori informazioni sull'impostazione degli allarmi, consulta Utilizzo degli CloudWatch allarmi Amazon.

Per visualizzare le metriche StatsD nella console CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Scegli lo spazio dei nomi da utilizzare per i parametri raccolti dall'agente. Per impostazione predefinita, si tratta di CWAgent, ma è possibile che sia stato specificato uno spazio dei nomi diverso nel file di configurazione dell' CloudWatch agente.

4. Scegli una dimensione di parametro (ad esempio, Per-Instance Metrics (Parametri per istanza)).
5. La scheda All metrics (Tutti i parametri) visualizza tutti i parametri per tale dimensione nello spazio dei nomi. Puoi eseguire le operazioni indicate di seguito:

   a. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.

   b. Per ordinare la tabella, utilizza l'intestazione della colonna.

   c. Per filtrare per risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).

   d. Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).
6. (Facoltativo) Per aggiungere questo grafico a un CloudWatch pannello di controllo, scegli Azioni, Aggiungi al pannello di controllo.

Recupero dei parametri personalizzati con collectd

Puoi recuperare metriche aggiuntive dalle tue applicazioni o servizi utilizzando l' CloudWatchagente con il protocollo collectd, supportato solo sui server Linux. collectd è una popolare soluzione open source con plugin in grado di raccogliere statistiche di sistema per un'ampia varietà di applicazioni. Combinando le metriche di sistema che l' CloudWatch agente può già raccogliere con le metriche aggiuntive di collectd, puoi monitorare, analizzare e risolvere meglio i tuoi sistemi e le tue applicazioni. Per ulteriori informazioni su collectd, consulta collectd - Il daemon di raccolta delle statistiche di sistema.

Si utilizza il software collectd per inviare le metriche all'agente. CloudWatch Per le metriche collectd, l' CloudWatch agente funge da server mentre il plugin collectd funge da client.

Il software collectd non è installato automaticamente su ogni server. Su un server che esegue Amazon Linux 2, attieniti alla seguente procedura per installare collectd

```
sudo amazon-linux-extras install collectd
```

Per informazioni sull'installazione di collectd su altri sistemi, consulta la pagina dei download per collectd.

Per raccogliere questi parametri personalizzati, aggiungi una riga "collectd": {} alla sezione metrics_collected del file di configurazione dell'agente. È possibile aggiungere questa riga

manualmente. Se utilizzi la procedura guidata per creare il file di configurazione, è già tutto previsto. Per ulteriori informazioni, consulta la pagina Creare il file di configurazione CloudWatch dell'agente.

Sono disponibili anche parametri opzionali. Se utilizzi collectd e non utilizzi `/etc/collectd/auth_file` come collectd_auth_file, è necessario impostare alcune di queste opzioni.

- service_address: l'indirizzo del servizio che l'agente deve ascoltare. CloudWatch Il formato è `"udp://`*`ip`*`:`*`port`*. Il valore predefinito è `udp://127.0.0.1:25826`.
- name_prefix: Un prefisso da allegare all'inizio del nome di ogni parametro collectd. Il valore predefinito è `collectd_`. La lunghezza massima è 255 caratteri.
- collectd_security_level: Consente di specificare il livello di protezione per la comunicazione di rete. Il valore predefinito è encrypt.

  encrypt specifica che vengono accettati solo i dati criptati. sign specifica che vengono accettati solo i dati firmati e criptati. none specifica che vengono accettati tutti i dati. Se specifichi un valore per collectd_auth_file, i dati criptati vengono decriptati, se possibile.

  Per ulteriori informazioni, consulta Configurazione del client e Possibili interazioni nei Wiki collectd.
- collectd_auth_file Imposta un file con la mappatura tra nomi utente e password. Queste password vengono utilizzate per verificare le firme e decriptare i pacchetti di rete criptati. Se presenti, i dati firmati vengono verificati e i pacchetti criptati vengono decriptati. In caso contrario, i dati firmati vengono accettati senza controllare la firma e i dati criptati non possono essere decriptati.

  Il valore predefinito è `/etc/collectd/auth_file`.

  Se collectd_security_level è impostato su none, questo è facoltativo. Se si è impostato collectd_security_level su `encrypt` o sign, è necessario specificare collectd_auth_file.

  Per il formato del file di autorizzazione, ogni riga è un nome utente seguito da due punti e qualsiasi numero di spazi seguiti dalla password. Ad esempio:

  ```
  user1: user1_password

  user2: user2_password
  ```
- collectd_typesdb: un elenco di uno o più file che contengono le descrizioni dei set di dati. L'elenco deve essere circondato da parentesi, anche se c'è una sola voce nell'elenco. Ogni voce dell'elenco deve essere inclusa nelle virgolette doppie. Se sono presenti più voci, separale con le virgole. Il valore predefinito nei server Linux è `["/usr/share/collectd/types.db"]`. L'impostazione

predefinita nei computer macOs dipende dalla versione di collectd. Ad esempio, `["/usr/local/Cellar/collectd/5.12.0/share/collectd/types.db"]`.

Per ulteriori informazioni, consulta https://www.collectd.org/documentation/manpages/types.db.html.

- metrics_aggregation_interval: con quale frequenza, in secondi, aggrega le metriche in singoli punti dati. CloudWatch Il valore predefinito è 60 secondi. L'intervallo è compreso tra 0 e 172,000. Se si imposta il valore su 0 si disabilita l'aggregazione dei parametri collectd.

Di seguito è riportato un esempio della sezione collectd di un file di configurazione dell'agente.

```
{
   "metrics":{
      "metrics_collected":{
         "collectd":{
            "name_prefix":"My_collectd_metrics_",
            "metrics_aggregation_interval":120
         }
      }
   }
}
```

Visualizzazione delle metriche raccolte CloudWatch importate dall'agente

Dopo aver importato le metriche collectd in CloudWatch, puoi visualizzare queste metriche come grafici di serie temporali e creare allarmi in grado di controllare queste metriche e avvisarti se superano una soglia specificata. La procedura seguente mostra come visualizzare i parametri collectd come grafici delle serie temporali. Per ulteriori informazioni sull'impostazione degli allarmi, consulta Utilizzo degli CloudWatch allarmi Amazon.

CloudWatch Per visualizzare le metriche raccolte nella console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Scegli lo spazio dei nomi da utilizzare per i parametri raccolti dall'agente. Per impostazione predefinita, si tratta di CWAgent, ma è possibile che sia stato specificato uno spazio dei nomi diverso nel file di configurazione dell' CloudWatch agente.
4. Scegli una dimensione di parametro (ad esempio, Per-Instance Metrics (Parametri per istanza)).

5. La scheda All metrics (Tutti i parametri) visualizza tutti i parametri per tale dimensione nello spazio dei nomi. Puoi eseguire le operazioni indicate di seguito:

   a. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.

   b. Per ordinare la tabella, utilizza l'intestazione della colonna.

   c. Per filtrare per risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).

   d. Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).

6. (Facoltativo) Per aggiungere questo grafico a un CloudWatch pannello di controllo, scegli Azioni, Aggiungi al pannello di controllo.

Configurare e configurare la raccolta dei parametri Prometheus su istanze Amazon EC2

Le sezioni seguenti spiegano come installare l' CloudWatch agente con il monitoraggio Prometheus sulle istanze EC2 e come configurare l'agente per lo scraping di destinazioni aggiuntive. Fornisce inoltre esercitazioni per impostare carichi di lavoro di esempio per l'utilizzo dei test con il monitoraggio Prometheus.

Per informazioni sui sistemi operativi supportati dall'agente, consulta CloudWatch Raccogli metriche, log e tracce con l'agente CloudWatch

Requisiti del gruppo di sicurezza VPC

Se utilizzi un VPC, si applicano i seguenti requisiti.

- Le regole di ingresso dei gruppi di sicurezza per i carichi di lavoro Prometheus devono aprire le porte CloudWatch Prometheus all'agente per lo scraping delle metriche di Prometheus tramite l'IP privato.
- Le regole di uscita del gruppo di sicurezza per l' CloudWatch agente devono consentire all'agente di connettersi alla CloudWatch porta dei carichi di lavoro Prometheus tramite IP privato.

Argomenti

- CloudWatch Fase 1: Installare l'agente

- Passaggio 2: scraping delle origini Prometheus e importazione dei parametri
- Esempio: impostazione di carichi di lavoro di esempio Java/JMX per il test dei parametri Prometheus

CloudWatch Fase 1: Installare l'agente

Il primo passaggio consiste nell'installare l' CloudWatch agente sull'istanza EC2. Per istruzioni, consulta Installazione dell'agente CloudWatch .

Passaggio 2: scraping delle origini Prometheus e importazione dei parametri

L' CloudWatch agente con monitoraggio Prometheus necessita di due configurazioni per analizzare le metriche di Prometheus. Una è per le configurazioni standard Prometheus come documentato in <scrape_config> nella documentazione di Prometheus. L'altra è per la configurazione dell'agente. CloudWatch

Configurazione di Prometheus Scrape

<scrape_config>L' CloudWatch agente supporta le configurazioni scrape standard di Prometheus come documentato nella documentazione di Prometheus.https://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config È possibile modificare questa sezione per aggiornare le configurazioni già presenti in questo file e aggiungere ulteriori destinazioni di scraping Prometheus. Un file di configurazione di esempio contiene le seguenti righe di configurazione globali:

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
- job_name: MY_JOB
  sample_limit: 10000
  file_sd_configs:
    - files: ["C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_1.yaml",
 "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_2.yaml"]
```

La sezione `global` specifica i parametri validi in tutti i contesti di configurazione. Servono anche come valori predefiniti per altre sezioni di configurazione. Contiene i seguenti parametri:

- `scrape_interval`: definisce con quale frequenza recuperare le destinazioni.
- `scrape_timeout`: definisce quanto tempo attendere prima che una richiesta di scrape scada.

La sezione `scrape_configs` specifica un insieme di destinazioni e parametri che definiscono come eseguire lo scraping. Contiene i seguenti parametri:

- `job_name`: il nome del processo assegnato ali parametri raschiate per impostazione predefinita.
- `sample_limit`: limite di scraping sul numero di campioni sottoposti a scraping che verranno accettati.
- `file_sd_configs`: elenco delle configurazioni di individuazione del servizio file. Legge un insieme di file contenenti un elenco di zero o più configurazioni statiche. La sezione `file_sd_configs` contiene un parametro `files` che definisce i modelli per i file da cui vengono estratti i gruppi di destinazione.

L'agente supporta i seguenti tipi di configurazione di rilevamento dei servizi. CloudWatch

`static_config` Consente di specificare un elenco di destinazioni e un set di etichette comuni per loro. È il modo canonico per specificare destinazioni statiche in una configurazione scrape.

Di seguito è riportato un esempio di configurazione statica per lo scraping dei parametri Prometheus da un host locale. I parametri possono anche essere sottoposti a scraping da altri server se la porta Prometheus è aperta al server in cui viene eseguito l'agente.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_sd_1.yaml
- targets:
    - 127.0.0.1:9404
  labels:
    key1: value1
    key2: value2
```

Questo connettore fornisce i seguenti parametri:

- `targets`: le destinazioni sottoposte a scraping dalla configurazione statica.
- `labels`: etichette assegnate a tutti i parametri sottoposti a scraping dalle destinazioni.

`ec2_sd_config` Consente di recuperare le destinazioni di scraping dalle istanze Amazon EC2. Di seguito è riportato un esempio `ec2_sd_config` per lo scraping dei parametri Prometheus da un elenco di istanze EC2. Le porte Prometheus di queste istanze devono essere aperte sul server su cui viene eseguito l'agente. CloudWatch Il ruolo IAM per l'istanza EC2 in cui viene eseguito l' CloudWatch agente deve includere l'autorizzazione. `ec2:DescribeInstance` Ad esempio, puoi

allegare la policy gestita AmazonEC2 all'istanza su cui è ReadOnlyAccess in esecuzione l'agente. CloudWatch

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: MY_JOB
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - i-98765432109876543
              - i-12345678901234567
```

Questo connettore fornisce i seguenti parametri:

- `region`— La AWS regione in cui si trova l'istanza EC2 di destinazione. Se si lascia vuoto, verrà utilizzata la regione dai metadati dell'istanza.
- `port`: la porta da cui eseguire lo scraping dei parametri.
- `filters`: filtri facoltativi da utilizzare per filtrare l'elenco delle istanze. L'esempio viene filtrato in base agli ID di istanza EC2. Per ulteriori criteri in base ai quali filtrare, consulta DescribeInstances.

CloudWatch configurazione dell'agente per Prometheus

Il file di configurazione CloudWatch dell'agente include `prometheus` sezioni in entrambe le sezioni. `logs` `metrics_collected` Include i seguenti parametri.

- cluster_name: specifica il nome del cluster da aggiungere come etichetta nell'evento log. Questo campo è facoltativo.
- log_group_name: specifica il nome del gruppo di log per i parametri Prometheus.
- prometheus_config_path: specifica il percorso del file di configurazione di scraping di Prometheus.
- emf_processor: specifica la configurazione del processore con formato metrico incorporato. Per ulteriori informazioni sul formato della metrica incorporata, vedere Incorporamento dei parametri nei log.

La sezione `emf_processor` può contenere i parametri seguenti:

- metric_declaration_dedup: se impostato su true, la funzione di deduplicazione dei parametri con formato metrico incorporato è abilitata.
- metric_namespace: specifica lo spazio dei nomi delle metriche per le metriche emesse. CloudWatch
- metric_unit: specifica la mappa metric name:metric unit. Per informazioni sulle unità metriche supportate, MetricDatumconsulta.
- metric_declaration: sono sezioni che specificano la matrice di log con formato metrico incorporato da generare. Esistono `metric_declaration` sezioni per ogni sorgente Prometheus da cui l'agente importa per impostazione predefinita CloudWatch. Ciascuna di queste sezioni include i seguenti campi:
  - `source_labels` specifica il valore delle etichette controllate dalla riga `label_matcher`.
  - `label_matcher` è un'espressione regolare che controlla il valore delle etichette elencate in `source_labels`. Le metriche corrispondenti sono abilitate per l'inclusione nel formato metrico incorporato inviato a. CloudWatch
  - `metric_selectors`è un'espressione regolare che specifica le metriche da raccogliere e a cui inviare. CloudWatch
  - `dimensions`è l'elenco di etichette da utilizzare come CloudWatch dimensioni per ogni metrica selezionata.

Di seguito è riportato un esempio di configurazione CloudWatch dell'agente per Prometheus.

```
{
   "logs":{
      "metrics_collected":{
         "prometheus":{
            "cluster_name":"prometheus-cluster",
            "log_group_name":"Prometheus",
            "prometheus_config_path":"C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\prometheus.yaml",
            "emf_processor":{
               "metric_declaration_dedup":true,
               "metric_namespace":"CWAgent-Prometheus",
               "metric_unit":{
                  "jvm_threads_current": "Count",
                  "jvm_gc_collection_seconds_sum": "Milliseconds"
```

```
            },
            "metric_declaration":[
               {
                  "source_labels":[
                     "job", "key2"
                  ],
                  "label_matcher":"MY_JOB;^value2",
                  "dimensions":[
                     [
                        "key1", "key2"
                     ],
                     [
                        "key2"
                     ]
                  ],
                  "metric_selectors":[
                     "^jvm_threads_current$",
                     "^jvm_gc_collection_seconds_sum$"
                  ]
               }
            ]
         }
      }
   }
}
```

Nell'esempio precedente viene configurata una sezione di formato metrica incorporata da inviare come evento di log se sono soddisfatte le seguenti condizioni:

- Il valore dell'etichetta `job` è `MY_JOB`
- Il valore dell'etichetta `key2` è `value2`
- I parametri Prometheus `jvm_threads_current` e `jvm_gc_collection_seconds_sum` contengono le etichette `job` e `key2`.

L'evento log inviato include la seguente sezione evidenziata.

```
{
    "CloudWatchMetrics": [
        {
            "Metrics": [
```

```
                {
                    "Unit": "Count",
                    "Name": "jvm_threads_current"
                },
                {
                    "Unit": "Milliseconds",
                    "Name": "jvm_gc_collection_seconds_sum"
                }
            ],
            "Dimensions": [
                [
                    "key1",
                    "key2"
                ],
                [
                    "key2"
                ]
            ],
            "Namespace": "CWAgent-Prometheus"
        }
    ],
    "ClusterName": "prometheus-cluster",
    "InstanceId": "i-0e45bd06f196096c8",
    "Timestamp": "1607966368109",
    "Version": "0",
    "host": "EC2AMAZ-PDDOIUM",
    "instance": "127.0.0.1:9404",
    "jvm_threads_current": 2,
    "jvm_gc_collection_seconds_sum": 0.006000000000000002,
    "prom_metric_type": "gauge",
    ...
}
```

Esempio: impostazione di carichi di lavoro di esempio Java/JMX per il test dei parametri Prometheus

JMX Exporter è un esportatore ufficiale di Prometheus che può recuperare ed esporre JMX mBeans JMX come metriche Prometheus. Per ulteriori informazioni, vedere prometheus/jmx_exporter.

L' CloudWatch agente può raccogliere metriche Prometheus predefinite da Java Virtual Machine (JVM), Hjava e Tomcat (Catalina), da un esportatore JMX su istanze EC2.

CloudWatch Fase 1: Installare l'agente

Il primo passaggio consiste nell'installare l' CloudWatch agente sull'istanza EC2. Per istruzioni, consulta la pagina Installazione dell'agente CloudWatch .

Passaggio 2: avvio del carico di lavoro Java/JMX

Il prossimo passaggio consiste nell'avviare il carico di lavoro Java/JMX.

Per prima cosa, scarica il file jar di JMX Exporter più recente dal seguente percorso:prometheus/jmx_exporter.

Uso del file jar per l'applicazione di esempio

I comandi di esempio nelle seguenti sezioni usano `SampleJavaApplication-1.0-SNAPSHOT.jar` come file jar. Sostituisci queste parti dei comandi con il file jar per la tua applicazione.

Preparazione della configurazione di JMX Exporter

Il file `config.yaml` è il file di configurazione JMX Exporter. Per ulteriori informazioni, consulta la sezione relativa alla configurazione nella documentazione di JMX Exporter.

Ecco una configurazione di esempio per Java e Tomcat.

```
---
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
```

```
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

Avvio dell'applicazione Java con Prometheus Exporter

Avvio dell'applicazione di esempio. Questo invierà i parametri Prometheus alla porta 9404. Assicurati di sostituire il punto di ingresso `com.gubupt.sample.app.App` con le informazioni corrette per l'applicazione Java di esempio.

In Linux immetti il seguente comando.

```
$ nohup java -javaagent:./jmx_prometheus_javaagent-0.14.0.jar=9404:./config.yaml -cp
  ./SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App &
```

In Windows immetti il seguente comando.

```
PS C:\> java -javaagent:.\jmx_prometheus_javaagent-0.14.0.jar=9404:.\config.yaml -cp  .
\SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App
```

Verifica dell'invio dei parametri Prometheus

Verifica che vengano inviati i parametri Prometheus.

In Linux immetti il seguente comando.

```
$ curl localhost:9404
```

In Windows immetti il seguente comando.

```
PS C:\> curl  http://localhost:9404
```

Esempio di output su Linux:

```
StatusCode        : 200
StatusDescription : OK
Content           : # HELP jvm_classes_loaded The number of classes that are currently
 loaded in the JVM
                    # TYPE jvm_classes_loaded gauge
                    jvm_classes_loaded 2526.0
                    # HELP jvm_classes_loaded_total The total number of class...
RawContent        : HTTP/1.1 200 OK
                    Content-Length: 71908
                    Content-Type: text/plain; version=0.0.4; charset=utf-8
                    Date: Fri, 18 Dec 2020 16:38:10 GMT

                    # HELP jvm_classes_loaded The number of classes that are
 currentl...
Forms             : {}
Headers           : {[Content-Length, 71908], [Content-Type, text/plain; version=0.0.4;
 charset=utf-8], [Date, Fri, 18
                    Dec 2020 16:38:10 GMT]}
Images            : {}
```

```
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 71908
```

Fase 3: Configurare l' CloudWatch agente per acquisire le metriche di Prometheus

Quindi, configura la configurazione dello scrape Prometheus nel file di configurazione dell'agente. CloudWatch

Per impostare la configurazione di scraping di Prometheus per l'esempio Java/JMX

1. Impostare la configurazione per `file_sd_config` e `static_config`.

   In Linux immetti il seguente comando.

   ```
   $ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus.yaml
   global:
     scrape_interval: 1m
     scrape_timeout: 10s
   scrape_configs:
     - job_name: jmx
       sample_limit: 10000
       file_sd_configs:
         - files: [ "/opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml" ]
   ```

   In Windows immetti il seguente comando.

   ```
   PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
   global:
     scrape_interval: 1m
     scrape_timeout: 10s
   scrape_configs:
     - job_name: jmx
       sample_limit: 10000
       file_sd_configs:
         - files: [ "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
   \prometheus_file_sd.yaml" ]
   ```

2. Impostare la configurazione delle destinazioni di scraping.

   In Linux immetti il seguente comando.

```
$ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: linux
```

In Windows immetti il seguente comando.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: windows
```

3. Impostare la configurazione di scraping di Prometheus tramite `ec2_sc_config`. Sostituire *`your-ec2-istance-id`* con l'ID dell'istanza EC2 corretto.

   In Linux immetti il seguente comando.

```
$ cat .\prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - your-ec2-instance-id
```

   In Windows immetti il seguente comando.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
```

```
  labels:
    application: sample_java_app
    os: windows
```

4. Imposta la configurazione dell'agente. CloudWatch Innanzitutto, passa alla directory corretta. In Linux è `/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json`. In Windows è `C:\ProgramData\Amazon\AmazonCloudWatchAgent\cwagent-config.json`.

   Di seguito è riportata una configurazione di esempio con i parametri Prometheus Java/JHX definiti. Assicurarti di sostituire *`path-to-Prometheus-Scrape-Configuration-file`* con il percorso corretto.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
      "prometheus": {
        "cluster_name": "my-cluster",
        "log_group_name": "prometheus-test",
        "prometheus_config_path": "path-to-Prometheus-Scrape-Configuration-file",
        "emf_processor": {
          "metric_declaration_dedup": true,
          "metric_namespace": "PrometheusTest",
          "metric_unit":{
            "jvm_threads_current": "Count",
            "jvm_classes_loaded": "Count",
            "java_lang_operatingsystem_freephysicalmemorysize": "Bytes",
            "catalina_manager_activesessions": "Count",
            "jvm_gc_collection_seconds_sum": "Seconds",
            "catalina_globalrequestprocessor_bytesreceived": "Bytes",
            "jvm_memory_bytes_used": "Bytes",
            "jvm_memory_pool_bytes_used": "Bytes"
          },
          "metric_declaration": [
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["instance"]],
              "metric_selectors": [
                "^jvm_threads_current$",
                "^jvm_classes_loaded$",
```

```
                "^java_lang_operatingsystem_freephysicalmemorysize$",
                "^catalina_manager_activesessions$",
                "^jvm_gc_collection_seconds_sum$",
                "^catalina_globalrequestprocessor_bytesreceived$"
              ]
            },
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["area"]],
              "metric_selectors": [
                "^jvm_memory_bytes_used$"
              ]
            },
            {
              "source_labels": ["job"],
              "label_matcher": "^jmx$",
              "dimensions": [["pool"]],
              "metric_selectors": [
                "^jvm_memory_pool_bytes_used$"
              ]
            }
          ]
        }
      }
    },
    "force_flush_interval": 5
  }
}
```

5. Riavviare l' CloudWatch agente immettendo uno dei seguenti comandi.

   In Linux immetti il seguente comando.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json
   ```

   In Windows immetti il seguente comando.

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
    -a fetch-config -m ec2 -s -c file:C:\ProgramData\Amazon\AmazonCloudWatchAgent
   \cwagent-config.json
   ```

Visualizzazione dei parametri e dei log Prometheus

È ora possibile visualizzare i parametri Java/JMX raccolti.

Per visualizzare i parametri per il carico di lavoro Java/JMX

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nella regione in cui è in esecuzione il cluster, scegli Metrics (Parametri) nel pannello di navigazione sinistro. Trova lo spazio dei PrometheusTestnomi per visualizzare le metriche.
3. Per visualizzare gli eventi di CloudWatch Logs, scegli Registra gruppi nel riquadro di navigazione. Gli eventi sono nel gruppo di log prometheus-test.

Installa l' CloudWatch agente utilizzando il componente aggiuntivo Amazon CloudWatch Observability EKS

Il componente aggiuntivo Amazon CloudWatch Observability EKS installa l' CloudWatch agente e l'agente Fluent-bit su un cluster Amazon EKS, con l'osservabilità migliorata di Container Insights per Amazon EKS e CloudWatch Application Signals abilitata per impostazione predefinita. Utilizzando il componente aggiuntivo, è possibile raccogliere i parametri dell'infrastruttura, la telemetria delle prestazioni delle applicazioni i log dei container dal cluster Amazon EKS.

Con Approfondimenti sui container con osservabilità migliorata per Amazon EKS, i parametri di Approfondimenti sui container vengono addebitati per osservazione anziché per parametro archiviato o log importato. Per Application Signals, la fatturazione si basa sulle richieste in entrata alle applicazioni, sulle richieste in uscita dalle applicazioni e su ogni obiettivo del livello di servizio (SLO) configurato. Ogni richiesta in entrata ricevuta e ogni richiesta in uscita effettuata genera un segnale di applicazione. Ogni SLO crea due segnali di applicazione per ciascun periodo di misurazione. Per ulteriori informazioni sui CloudWatch prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Il componente aggiuntivo Amazon EKS abilita Container Insights su nodi di lavoro Linux e Windows nel cluster Amazon EKS. Per abilitare Container Insights su Windows, devi utilizzare la versione 1.5.0 o successiva del componente aggiuntivo Amazon EKS. Attualmente, Application Signals non è supportato su Windows nei cluster Amazon EKS.

Il componente aggiuntivo Amazon CloudWatch Observability EKS è supportato sui cluster Amazon EKS in esecuzione con Kubernetes versione 1.23 o successiva.

Quando installi il componente aggiuntivo, devi anche concedere le autorizzazioni IAM per consentire all' CloudWatch agente di inviare metriche, log e tracce a. CloudWatch Ci sono due modi per effettuare questa operazione:

- Collega una policy al ruolo IAM dei nodi di lavoro. Questa opzione concede ai nodi di lavoro le autorizzazioni a cui inviare telemetria. CloudWatch
- Utilizza un ruolo IAM per gli account di servizio per i pod dell'agente e collega la policy a questo ruolo. Questo metodo funziona solo per i cluster Amazon EKS. Questa opzione consente l' CloudWatch accesso solo ai pod degli agenti appropriati.

Opzione 1: installazione con autorizzazioni IAM sui nodi worker

Per utilizzare questo metodo, collega innanzitutto la policy CloudWatchAgentServerPolicyIAM ai nodi di lavoro inserendo il seguente comando. In questo comando, sostituiscilo *my-worker-node-role*con il ruolo IAM utilizzato dai nodi di lavoro Kubernetes.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Quindi installa il componente aggiuntivo Amazon CloudWatch Observability EKS. Per installare il componente aggiuntivo, puoi utilizzare la AWS CLI console o Terraform AWS CloudFormation.

AWS CLI

Per utilizzare il AWS CLI componente aggiuntivo Amazon CloudWatch Observability EKS

Inserire il seguente comando. Sostituisci *my-cluster-name* con il nome del cluster.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-name my-cluster-name
```

Amazon EKS console

Per utilizzare la console Amazon EKS per aggiungere il componente aggiuntivo Amazon CloudWatch Observability EKS

1. Apri la console Amazon EKS all'indirizzo https://console.aws.amazon.com/eks/home#/clusters.

2. Nel pannello di navigazione a sinistra, seleziona Cluster.
3. Scegli il nome del cluster per cui desideri configurare il componente aggiuntivo Amazon CloudWatch Observability EKS.
4. Seleziona la scheda Componenti aggiuntivi.
5. Scegli Ottieni altri componenti aggiuntivi.
6. Nella pagina Seleziona componenti aggiuntivi, procedi come segue:
   a. Nella sezione Amazon EKS-Addons, seleziona la casella di controllo Amazon CloudWatch Observability.
   b. Seleziona Successivo.
7. Nella pagina Configura le impostazioni dei componenti aggiuntivi selezionati, procedi come segue:
   a. Seleziona la Versione che desideri utilizzare.
   b. Per Seleziona ruolo IAM, scegli Eredita dal nodo.
   c. (Facoltativo) È possibile espandere le Impostazioni di configurazione facoltative. Se selezioni Sostituisci per Metodo di risoluzione dei conflitti, una o più impostazioni per il componente aggiuntivo esistente possono essere sovrascritte con le impostazioni del componente aggiuntivo di Amazon EKS. Se rimuovi questa opzione e c'è un conflitto con le impostazioni esistenti, l'operazione non va a buon fine e viene visualizzato un messaggio di errore per aiutarti a risolvere il conflitto. Prima di selezionare questa opzione, assicurarsi che il componente aggiuntivo Amazon EKS non gestisca le impostazioni che devono essere gestite dall'utente.
   d. Seleziona Successivo.
8. Nella pagina Rivedi e aggiungi, scegli Crea. Una volta completata l'installazione, viene visualizzato il componente aggiuntivo.

AWS CloudFormation

Da utilizzare AWS CloudFormation per installare il componente aggiuntivo Amazon CloudWatch Observability EKS

Sostituisci *my-cluster-name* con il nome del cluster. Per ulteriori informazioni, consulta. AWS::EKS::Addon

```
{
```

```
    "Resources": {
        "EKSAddOn": {
            "Type": "AWS::EKS::Addon",
            "Properties": {
                "AddonName": "amazon-cloudwatch-observability",
                "ClusterName": "my-cluster-name"
            }
        }
    }
}
```

Terraform

Per utilizzare Terraform per installare il componente aggiuntivo Amazon CloudWatch Observability EKS

Sostituisci *my-cluster-name* con il nome del cluster. Per ulteriori informazioni, consulta la sezione Risorsa: aws_eks_addon.

```
resource "aws_eks_addon" "example" {
  addon_name   = "amazon-cloudwatch-observability"
  cluster_name = "my-cluster-name"
}
```

Opzione 2: installazione tramite il ruolo dell'account di servizio IAM

Prima di utilizzare questo metodo, verifica di soddisfare i seguenti prerequisiti:

- Disponi di un cluster Amazon EKS funzionale con nodi collegati in una delle Regioni AWS che supportano Approfondimenti sui container. Per l'elenco delle regioni supportate, consulta Container Insights.
- `kubectl` deve essere installato e configurato per il cluster. Per ulteriori informazioni, consulta la pagina relativa all'installazione di `kubectl` nella Guida per l'utente di Amazon EKS.
- `eksctl` deve essere installato. Per ulteriori informazioni, consulta la pagina Installing or updating `eksctl` nella Guida per l'utente di Amazon EKS.

Per installare il componente aggiuntivo Amazon CloudWatch Observability EKS utilizzando il ruolo dell'account del servizio IAM

1. Immetti il comando seguente per creare un provider OpenID Connect (OIDC), se il cluster non ne ha già uno. Per ulteriori informazioni, consulta la pagina Configuring a Kubernetes service account to assume an IAM role nella Guida per l'utente di Amazon EKS.

```
eksctl utils associate-iam-oidc-provider --cluster my-cluster-name --approve
```

2. Immetti il seguente comando per creare il ruolo IAM con la CloudWatchAgentServerPolicypolicy allegata e configura l'account del servizio dell'agente per assumere quel ruolo utilizzando OIDC. Sostituiscilo *my-cluster-name*con il nome del cluster e sostituiscilo *my-service-account-role*con il nome del ruolo a cui desideri associare l'account di servizio. Se il ruolo non esiste ancora, `eksctl` lo crea per tuo conto.

```
eksctl create iamserviceaccount \
  --name cloudwatch-agent \
  --namespace amazon-cloudwatch --cluster my-cluster-name \
  --role-name my-service-account-role \
  --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
  --role-only \
  --approve
```

3. Installa il componente aggiuntivo immettendo il seguente comando. Sostituiscilo *my-cluster-name*con il nome del cluster, sostituisci *111122223333* con l'ID dell'account e sostituiscilo *my-service-account-role*con il ruolo IAM creato nel passaggio precedente.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-
name my-cluster-name --service-account-role-arn arn:aws:iam::111122223333:role/my-
service-account-role
```

(Facoltativo) Configurazione aggiuntiva

Disattiva la raccolta dei log dei container

Per impostazione predefinita, il componente aggiuntivo utilizza Fluent Bit per raccogliere i log dei contenitori da tutti i pod e quindi invia i log a Logs. CloudWatch Per informazioni su quali log vengono raccolti, consulta la pagina Configurazione di Fluent Bit.

Per disattivare la raccolta dei log dei contenitori, utilizza la seguente opzione quando crei o aggiorni il componente aggiuntivo:

```
--configuration-values '{ "containerLogs": { "enabled": false } }'
```

Disattiva la raccolta di metriche delle GPU NVIDIA

A partire dalla versione 1.300034.0 dell' CloudWatch agente, Container Insights raccoglie per impostazione predefinita le metriche delle GPU NVIDIA dai carichi di lavoro EKS. Queste metriche sono elencate nella tabella in. Metriche della GPU NVIDIA

Puoi scegliere di non raccogliere i parametri della GPU NVIDIA impostando l'`accelerated_compute_metrics`opzione nel file di configurazione dell' CloudWatch agente su. `false` Questa opzione si trova nella `kubernetes` sezione della `metrics_collected` sezione del file di configurazione. CloudWatch Di seguito è riportato un esempio di configurazione di opt-out.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
      "emf": {
      },
      "kubernetes": {
        "enhanced_container_insights": true,
        "accelerated_compute_metrics": false
      }
    },
    "force_flush_interval": 5,
  }
}
```

Utilizza una configurazione personalizzata CloudWatch dell'agente

Per raccogliere altre metriche, log o tracce utilizzando l' CloudWatch agente, puoi specificare una configurazione personalizzata mantenendo attivi Container Insights e CloudWatch Application Signals. A tale scopo, incorpora il file di configurazione dell' CloudWatch agente nella chiave di configurazione sotto la chiave dell'agente della configurazione avanzata che puoi utilizzare durante la creazione o l'aggiornamento del componente aggiuntivo EKS. Di seguito è rappresentata la configurazione predefinita dell'agente quando non si fornisce alcuna configurazione aggiuntiva.

> **Important**
>
> Qualsiasi configurazione personalizzata fornita utilizzando impostazioni di configurazione aggiuntive ha la precedenza sulla configurazione predefinita utilizzata dall'agente. Fai attenzione a non disattivare involontariamente funzionalità abilitate di default, come Container Insights con osservabilità migliorata e Application Signals. CloudWatch Se è necessario fornire una configurazione personalizzata dell'agente, consigliamo di utilizzare la seguente configurazione predefinita come base e modificarla di secondo le necessità.

```
--configuration-values '{
  "agent": {
    "config": {
      "logs": {
        "metrics_collected": {
          "app_signals": {},
          "kubernetes": {
            "enhanced_container_insights": true
          }
        }
      },
      "traces": {
        "traces_collected": {
          "app_signals": {}
        }
      }
    }

}'
```

L'esempio seguente mostra la configurazione predefinita dell'agente in Windows. CloudWatch L' CloudWatch agente su Windows non supporta la configurazione personalizzata.

```
{
  "logs": {
    "metrics_collected": {
      "kubernetes": {
        "enhanced_container_insights": true
      },
    }
  }
```

```
}
```

Gestisci i certificati webhook di ammissione TLS

Il componente aggiuntivo Amazon CloudWatch Observability EKS sfrutta i webhook di ammissione di Kubernetes per convalidare e modificare le richieste di risorse (CR) `Instrumentation` personalizzate `AmazonCloudWatchAgent` e, facoltativamente, le richieste pod Kubernetes sul cluster se Application Signals è abilitato. CloudWatch In Kubernetes, i webhook richiedono un certificato TLS configurato per stabilire una relazione di attendibilità con il server API per garantire una comunicazione sicura.

Per impostazione predefinita, il componente aggiuntivo Amazon CloudWatch Observability EKS genera automaticamente una CA autofirmata e un certificato TLS firmato da questa CA per proteggere la comunicazione tra il server API e il server webhook. Questo certificato generato automaticamente ha una scadenza predefinita di 10 anni e non viene rinnovato automaticamente alla scadenza. Inoltre, il bundle CA e il certificato vengono rigenerati ogni volta che il componente aggiuntivo viene aggiornato o reinstallato, reimpostando così la scadenza. Se desideri modificare la scadenza predefinita del certificato generato automaticamente, puoi utilizzare le seguenti configurazioni aggiuntive durante la creazione o l'aggiornamento del componente aggiuntivo. Sostituiscilo *`expiry-in-days`*con la durata di scadenza desiderata in giorni.

```
--configuration-values '{ "admissionWebhooks": { "autoGenerateCert":
 { "expiryDays": expiry-in-days } } }'
```

Per una soluzione di autorità di certificazione più sicura e ricca di funzionalità, il componente aggiuntivo offre il supporto opt-in per cert-manager, una soluzione ampiamente adottata per la gestione dei certificati TLS in Kubernetes che semplifica il processo di ottenimento, rinnovo, gestione e utilizzo di tali certificati. Garantisce che i certificati siano validi e aggiornati e tenta di rinnovarli in un momento impostato prima della scadenza. cert-manager facilita anche l'emissione di certificati da una varietà di fonti supportate, tra cui AWS Certificate Manager Private Certificate Authority.

Ti consigliamo di esaminare le best practice per la gestione dei certificati TLS sui tuoi cluster e di scegliere di utilizzare cert-manager per gli ambienti di produzione. Tieni presente che se scegli di abilitare cert-manager per la gestione dei certificati TLS del webhook di ammissione, devi preinstallare cert-manager sul tuo cluster Amazon EKS prima di installare il componente aggiuntivo Amazon Observability EKS. CloudWatch Consulta la documentazione di cert-manager per ulteriori informazioni sulle opzioni di installazione disponibili. Dopo averlo installato, puoi scegliere di utilizzare

cert-manager per la gestione dei certificati webhook di ammissione TLS utilizzando la seguente configurazione aggiuntiva durante la creazione o l'aggiornamento del componente aggiuntivo.

```
--configuration-values '{ "admissionWebhooks": { "certManager": { "enabled":
 true } } }'
```

La configurazione avanzata discussa in questa sezione utilizza per impostazione predefinita un emittente. SelfSigned

Raccolta degli ID dei volumi Amazon EBS

Se desideri raccogliere gli ID dei volumi Amazon EBS nei log delle prestazioni, ti basta aggiungere al ruolo IAM un'altra policy collegata ai nodi worker o all'account di servizio. Aggiungi quanto segue come una policy inline. Per ulteriori informazioni, consulta la pagina Adding and Removing IAM Identity Permissions.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Risoluzione dei problemi del componente aggiuntivo Amazon CloudWatch Observability EKS

Utilizza le seguenti informazioni per risolvere i problemi con il componente aggiuntivo Amazon CloudWatch Observability EKS.

Aggiornamento ed eliminazione del componente aggiuntivo Amazon CloudWatch Observability EKS

Per istruzioni sull'aggiornamento o l'eliminazione del componente aggiuntivo Amazon CloudWatch Observability EKS, consulta Managing Amazon EKS add-on. Usa `amazon-cloudwatch-observability` come nome del componente aggiuntivo.

Verifica la versione dell' CloudWatch agente utilizzato dal componente aggiuntivo Amazon CloudWatch Observability EKS

Il componente aggiuntivo Amazon CloudWatch Observability EKS installa una risorsa personalizzata `AmazonCloudWatchAgent` che controlla il comportamento del daemset dell' CloudWatchagente sul cluster, inclusa la versione dell'agente utilizzata. CloudWatch È possibile ottenere un elenco di tutte le risorse `AmazonCloudWatchAgent` personalizzate installate sul cluster u immettendo il seguente comando:

```
kubectl get amazoncloudwatchagent -A
```

Nell'output di questo comando, dovresti essere in grado di controllare la versione dell'agente. CloudWatch In alternativa, puoi anche descrivere la risorsa `amazoncloudwatchagent` o uno dei pod `cloudwatch-agent-*` in esecuzione sul cluster per ispezionare l'immagine utilizzata.

Gestione di un ConfigurationConflict durante la gestione del componente aggiuntivo

Quando installi o aggiorni il componente aggiuntivo Amazon CloudWatch Observability EKS, se noti un errore causato da un `Health Issue` di tipo `ConfigurationConflict` con una descrizione che inizia con`Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, è probabile che l' CloudWatch agente e i relativi componenti associati, come il ServiceAccount, il ClusterRole e il, siano ClusterRoleBinding installati nel cluster. Quando il componente aggiuntivo tenta di installare l' CloudWatch agente e i componenti associati, se rileva modifiche nei contenuti, per impostazione predefinita fallisce l'installazione o l'aggiornamento per evitare di sovrascrivere lo stato delle risorse sul cluster.

Se stai tentando di eseguire l'onboarding del componente aggiuntivo Amazon CloudWatch Observability EKS e riscontri questo errore, ti consigliamo di eliminare una configurazione di CloudWatch agente esistente che avevi precedentemente installato sul cluster e quindi di installare il componente aggiuntivo EKS. Assicurati di eseguire il backup di tutte le personalizzazioni che potresti aver apportato alla configurazione originale dell' CloudWatch agente, ad esempio una configurazione

personalizzata dell'agente, e di fornirle al componente aggiuntivo Amazon CloudWatch Ob servability EKS alla prossima installazione o aggiornamento. Se in precedenza avevi installato l' CloudWatch agente per l'onboarding su Container Insights, consulta per ulteriori informazioni. Eliminazione dell' CloudWatch agente e di Fluent Bit for Container Insights

In alternativa, il componente aggiuntivo supporta un'opzione di configurazione per la risoluzione dei conflitti che può specificare `OVERWRITE`. È possibile utilizzare questa opzione per procedere con l'installazione o l'aggiornamento del componente aggiuntivo sovrascrivendo i conflitti nel cluster. Se utilizzi la console Amazon EKS, trovi il metodo di risoluzione dei conflitti selezionando le impostazioni di configurazione facoltative quando crei o aggiorni il componente aggiuntivo. Se utilizzi il AWS CLI, puoi fornire il comando `--resolve-conflicts OVERWRITE` al tuo comando per creare o aggiornare il componente aggiuntivo.

Metriche raccolte dall'agente CloudWatch

Puoi raccogliere i parametri dai server installando l' CloudWatch agente sul server. Puoi installare l'agente nelle istanze Amazon EC2, nei computer locali e nei server che eseguono Linux, Windows Server o macOS. Se installi l'agente in un'istanza Amazon EC2, i parametri raccolti vengono aggiunti a quelli abilitati per impostazione predefinita nelle istanze Amazon EC2.

Per informazioni sull'installazione dell' CloudWatch agente su un'istanza, consultaRaccogli metriche, log e tracce con l'agente CloudWatch .

Tutte le metriche discusse in questa sezione vengono raccolte direttamente dall' CloudWatch agente.

Metriche raccolte dall' CloudWatchagente sulle istanze di Windows Server

Su un server che esegue Windows Server, l'installazione dell' CloudWatch agente consente di raccogliere le metriche associate ai contatori in Windows Performance Monitor. I nomi delle CloudWatch metriche per questi contatori vengono creati inserendo uno spazio tra il nome dell'oggetto e il nome del contatore. Ad esempio, al `% Interrupt Time` contatore dell'`Processor`oggetto viene assegnato il nome della metrica. `Processor % Interrupt Time` CloudWatch Per ulteriori informazioni sui contatori di Windows Performance Monitor, consulta la documentazione di Microsoft Windows Server.

Lo spazio dei nomi predefinito per le metriche raccolte dall' CloudWatch agente è`CWAgent`, sebbene sia possibile specificare uno spazio dei nomi diverso quando si configura l'agente.

Metriche raccolte dall' CloudWatchagente sulle istanze Linux e macOS

La tabella seguente elenca le metriche che è possibile raccogliere con l' CloudWatch agente su server Linux e computer macOS.

Parametro	Descrizione
`cpu_time_active`	Il periodo di tempo durante il quale la CPU è attiva su qualsiasi capacità. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_time_guest`	Il periodo di tempo durante il quale la CPU è in esecuzione su una CPU virtuale per un sistema operativo guest. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_time_guest_nice`	Il periodo di tempo durante il quale la CPU è in esecuzione su una CPU virtuale per un sistema operativo guest di bassa priorità e in cui può essere interrotta da altri processi. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_time_idle`	Il periodo di tempo durante il quale la CPU è inattiva. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_time_iowait`	Il periodo di tempo durante il quale la CPU è in attesa di operazioni di I/O da completare. Questo parametro è misurato in centesimi di secondo. Unità: nessuna

Parametro	Descrizione
`cpu_time_irq`	Il periodo di tempo durante il quale la CPU lavora sulle interruzioni. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_time_nice`	Il periodo di tempo durante il quale la CPU è in modalità utente con processi di bassa priorità che possono semplicemente essere interrotti da processi di priorità superiore. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_time_softirq`	Il periodo di tempo durante il quale la CPU lavora sulle interruzioni del software. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_time_steal`	Il periodo di tempo durante il quale la CPU si trova nello stato di stolen time (tempo rubato), ovvero il tempo impiegato su altri sistemi operativi in un ambiente virtualizzato. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_time_system`	Il periodo di tempo durante il quale la CPU si trova in modalità di sistema. Questo parametro è misurato in centesimi di secondo. Unità: nessuna

Parametro	Descrizione
`cpu_time_user`	Il periodo di tempo durante il quale la CPU si trova in modalità utente. Questo parametro è misurato in centesimi di secondo. Unità: nessuna
`cpu_usage_active`	La percentuale di tempo durante il quale la CPU è attiva su qualsiasi capacità. Unità: percentuale
`cpu_usage_guest`	La percentuale di tempo durante la quale la CPU è in esecuzione su una CPU virtuale per un sistema operativo guest. Unità: percentuale
`cpu_usage_guest_nice`	La percentuale di tempo durante la quale la CPU è in esecuzione su una CPU virtuale per un sistema operativo guest di bassa priorità e che può essere interrotta da altri processi. Unità: percentuale
`cpu_usage_idle`	La percentuale di tempo in cui la CPU è inattiva. Unità: percentuale
`cpu_usage_iowait`	La percentuale di tempo durante la quale la CPU è in attesa di operazioni di I/O da completare. Unità: percentuale
`cpu_usage_irq`	La percentuale di tempo durante la quale la CPU lavora sulle interruzioni. Unità: percentuale

Parametro	Descrizione
`cpu_usage_nice`	La percentuale di tempo durante il quale la CPU è in modalità utente con processi di bassa priorità che possono essere interrotti con facilità da processi di priorità superiore. Unità: percentuale
`cpu_usage_softirq`	La percentuale di tempo durante la quale la CPU lavora sulle interruzioni del software. Unità: percentuale
`cpu_usage_steal`	La percentuale di tempo durante il quale la CPU si trova nello stato di stolen time, ovvero il tempo impiegato su altri sistemi operativi in un ambiente virtualizzato. Unità: percentuale
`cpu_usage_system`	La percentuale di tempo durante la quale la CPU si trova in modalità di sistema. Unità: percentuale
`cpu_usage_user`	La percentuale di tempo durante la quale la CPU si trova in modalità utente. Unità: percentuale
`disk_free`	Spazio libero sui dischi. Unità: byte
`disk_inodes_free`	Il numero di nodi dell'indice disponibili sul disco. Unità: numero

Parametro	Descrizione
`disk_inodes_total`	Il numero totale di nodi dell'indice prenotati sul disco. Unità: numero
`disk_inodes_used`	Il numero di nodi dell'indice usati sul disco. Unità: numero
`disk_total`	Spazio totale dei dischi, inclusi quelli usati e quelli gratuiti. Unità: byte
`disk_used`	Spazio usato sui dischi. Unità: byte
`disk_used_percent`	La percentuale di spazio totale del disco usata. Unità: percentuale
`diskio_iops_in_progress`	Il numero di richieste di I/O indirizzate al driver del dispositivo, ma che non sono ancora state completate. Unità: numero
`diskio_io_time`	La quantità di tempo impiegata dal disco per le richieste di I/O in coda. Unità: millisecondi L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.

Parametro	Descrizione
`diskio_reads`	Il numero di operazioni di lettura del disco. Unità: numero L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.
`diskio_read_bytes`	Il numero di byte letti dai dischi. Unità: byte L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.
`diskio_read_time`	Il periodo di tempo che le richieste di lettura hanno aspettato su dischi. Varie richieste di lettura in attesa contemporaneamente aumentano il numero. Ad esempio, se tutte e 5 le richieste sono in attesa per una media di 100 millisecondi, ne vengono segnalati 500. Unità: millisecondi L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.
`diskio_writes`	Il numero di operazioni di scrittura del disco. Unità: numero L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.
`diskio_write_bytes`	Il numero di byte scritti sui dischi. Unità: byte L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.

Parametro	Descrizione
`diskio_write_time`	Il periodo di tempo che le richieste di scrittura hanno aspettato sui dischi. Varie richieste di scrittura in attesa contemporaneamente aumentano il numero. Ad esempio, se tutte e 8 le richieste sono in attesa per una media di 1000 millisecondi, ne vengono segnalati 8000. Unità: millisecondi L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.
`ethtool_bw_in_allowance_exceeded`	Il numero di pacchetti accordati e/o rilasciati perché la larghezza di banda aggregata in ingresso ha superato il valore massimo per l'istanza. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione `metrics_collected` dell' CloudWatch agente. Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna
`ethtool_bw_out_allowance_exceeded`	Il numero di pacchetti accodati e/o rilasciati perché la larghezza di banda aggregata in uscita ha superato il valore massimo per l'istanza. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione `metrics_collected` dell' CloudWatch agente. Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna

Parametro	Descrizione
`ethtool_conntrack_allowance_exceeded`	Il numero di pacchetti accodati o rilasciati perché il rilevamento delle connessioni ha superato il valore massimo per l'istanza e non è stato possibile stabilire nuove connessioni. Ciò può comportare la perdita di pacchetti per il traffico da o verso l'istanza. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione `metrics_collected` dell' CloudWatch agente. Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna
`ethtool_linklocal_allowance_exceeded`	Il numero di pacchetti accodati o rilasciati perché il PPS del traffico verso i servizi proxy locali ha superato il valore massimo per l'interfaccia di rete. Ciò influisce sul traffico verso il servizio DNS, il servizio di metadati dell'istanza e il servizio Amazon Time Sync. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione `metrics_collected` dell' CloudWatch agente. Per ulteriori informazioni, consulta la pagina Raccolta di parametri sulle prestazioni di rete Unità: nessuna

Parametro	Descrizione
`ethtool_pps_allowance_exceeded`	Il numero di pacchetti accodati e/o rilasciati perché il PPS bidirezionale ha superato il valore massimo per l'istanza. Questa metrica viene raccolta solo se è stata elencata nella `ethtool` sottosezione della sezione del file di configurazione `metrics_collected` dell' CloudWatch agente. Per ulteriori informazioni, consulta Raccolta di parametri sulle prestazioni di rete. Unità: nessuna
`mem_active`	La quantità di memoria utilizzata in un modo qualsiasi durante l'ultimo periodo di campionamento. Unità: byte
`mem_available`	La quantità di memoria che è disponibile e che può essere immediatamente determinata durante i processi. Unità: byte
`mem_available_percent`	La percentuale di memoria che è disponibile e che può essere immediatamente determinata durante i processi. Unità: percentuale
`mem_buffered`	La quantità di memoria che viene utilizzata per i buffer. Unità: byte

Parametro	Descrizione
`mem_cached`	La quantità di memoria che viene utilizzata per le cache dei file. Unità: byte
`mem_free`	La quantità di memoria che non viene utilizzata. Unità: byte
`mem_inactive`	La quantità di memoria non utilizzata in alcun modo durante l'ultimo periodo di campionamento. Unità: byte
`mem_total`	La quantità totale di memoria. Unità: byte
`mem_used`	La quantità di memoria attualmente in uso. Unità: byte
`mem_used_percent`	La percentuale di memoria attualmente in uso. Unità: percentuale
`net_bytes_recv`	Il numero di byte ricevuti dall'interfaccia di rete. Unità: byte L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.
`net_bytes_sent`	Il numero di byte inviati dall'interfaccia di rete. Unità: byte L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.

Parametro	Descrizione
net_drop_in	Il numero di pacchetti ricevuti da questa interfaccia di rete che sono stati interrotti. Unità: numero L'unica statistica da utilizzare per questo parametro è Sum. Non usare Average.
net_drop_out	Il numero di pacchetti trasmessi da questa interfaccia di rete che sono stati interrotti. Unità: numero L'unica statistica da utilizzare per questo parametro è Sum. Non usare Average.
net_err_in	Il numero di errori ricevuti rilevati da questa interfacc ia di rete. Unità: numero L'unica statistica da utilizzare per questo parametro è Sum. Non usare Average.
net_err_out	Il numero di errori trasmessi rilevati da questa interfaccia di rete. Unità: numero L'unica statistica da utilizzare per questo parametro è Sum. Non usare Average.
net_packets_sent	Il numero di pacchetti inviati da questa interfaccia di rete. Unità: numero L'unica statistica da utilizzare per questo parametro è Sum. Non usare Average.

Parametro	Descrizione
`net_packets_recv`	Il numero di pacchetti ricevuti da questa interfaccia di rete. Unità: numero L'unica statistica da utilizzare per questo parametro è `Sum`. Non usare `Average`.
`netstat_tcp_close`	Il numero di connessioni TCP senza stato. Unità: numero
`netstat_tcp_close_wait`	Il numero di connessioni TCP in attesa di una richiesta di terminazione dal cliente. Unità: numero
`netstat_tcp_closing`	Il numero di connessioni TCP in attesa di una richiesta di terminazione con conferma dal cliente. Unità: numero
`netstat_tcp_established`	Il numero di connessioni TCP stabilite. Unità: numero
`netstat_tcp_fin_wait1`	Il numero di connessioni TCP nello stato `FIN_WAIT1` durante la chiusura di una connessione. Unità: numero
`netstat_tcp_fin_wait2`	Il numero di connessioni TCP nello stato `FIN_WAIT2` durante la chiusura di una connessione. Unità: numero

Parametro	Descrizione
`netstat_tcp_last_ack`	Il numero di connessioni TCP in attesa dell'invio da parte del cliente della conferma del messaggio di terminazione della connessione. Questo è l'ultimo stato prima della chiusura della connessione. Unità: numero
`netstat_tcp_listen`	Il numero di porte TCP attualmente in ascolto di una richiesta di connessione. Unità: numero
`netstat_tcp_none`	Il numero di connessioni TCP con clienti inattivi. Unità: numero
`netstat_tcp_syn_sent`	Il numero di connessioni TCP in attesa di una richiesta di connessione corrispondente dopo aver inviato una richiesta di connessione. Unità: numero
`netstat_tcp_syn_recv`	Il numero di connessioni TCP in attesa di una conferma di richiesta di connessione dopo aver inviato e ricevuto una richiesta di connessione. Unità: numero
`netstat_tcp_time_wait`	Il numero di connessioni TCP attualmente in attesa che servono per assicurare al cliente la ricezione della conferma della sua richiesta di terminazione della connessione. Unità: numero
`netstat_udp_socket`	Il numero delle attuali connessioni UDP. Unità: numero

Parametro	Descrizione
`processes_blocked`	Il numero di processi che sono bloccati. Unità: numero
`processes_dead`	Il numero di processi "dead", indicati dal codice di stato `X` su Linux. Questo parametro non viene raccolto sui computer macOS. Unità: numero
`processes_idle`	Il numero di processi che sono inattivi (che sono in stato di sospensione per più di 20 secondi). Disponibile solo sulle istanze di FreeBSD. Unità: numero
`processes_paging`	Il numero di processi in fase di paging, indicati dal codice di stato `W` su Linux. Questo parametro non viene raccolto sui computer macOS. Unità: numero
`processes_running`	Il numero di processi in elaborazione, indicati dal codice di stato `R`. Unità: numero
`processes_sleeping`	Il numero di processi in fase di sospensione, indicati dal codice di stato `S`. Unità: numero

Parametro	Descrizione
`processes_stopped`	Il numero di processi arrestati, indicati dal codice di stato `T`. Unità: numero
`processes_total`	Il numero totale di processi sull'istanza. Unità: numero
`processes_total_threads`	Il numero totale di thread che costituiscono i processi. Questo parametro è disponibile solo per le istanze su Linux. Questo parametro non viene raccolto sui computer macOS. Unità: numero
`processes_wait`	Il numero di processi in fase di paging, indicati dal codice di stato `W` nelle istanze FreeBSD. Questo parametro è disponibile solo sulle istanze di FreeBSD e non è disponibile su Linux, Windows Server o macOS. Unità: numero
`processes_zombies`	Il numero di processi zombie, indicati dal codice di stato `Z`. Unità: numero
`swap_free`	La quantità di spazio di swapping che non viene usata. Unità: byte
`swap_used`	La quantità di spazio di swapping attualmente in uso. Unità: byte

Parametro	Descrizione
`swap_used_percent`	La percentuale di spazio di swapping attualmente in uso. Unità: percentuale

Definizioni delle metriche di memoria raccolte dall'agente CloudWatch

Quando l' CloudWatch agente raccoglie i parametri di memoria, la fonte è il sottosistema di gestione della memoria dell'host. Ad esempio, il kernel Linux espone i dati gestiti dal sistema operativo in `/proc`. Per quanto riguarda la memoria, i dati si trovano in `/proc/meminfo`.

Ogni sistema operativo e architettura diversi prevede calcoli diversi delle risorse utilizzate dai processi. Per ulteriori informazioni, consultare le sezioni indicate di seguito.

Durante ogni intervallo di raccolta, l' CloudWatch agente di ogni istanza raccoglie le risorse dell'istanza e calcola le risorse utilizzate da tutti i processi in esecuzione in quell'istanza. Queste informazioni vengono riportate alle metriche. CloudWatch È possibile configurare la lunghezza dell'intervallo di raccolta nel file di configurazione dell' CloudWatch agente. Per ulteriori informazioni, consulta CloudWatch file di configurazione dell'agente: sezione Agente.

L'elenco seguente spiega come vengono definite le metriche di memoria raccolte dall' CloudWatch agente.

- Memoria attiva: la memoria utilizzata da un processo. In altre parole, la memoria utilizzata dalle app attualmente in esecuzione.
- Memoria disponibile: la memoria che può essere assegnata istantaneamente ai processi senza che il sistema vada in swap (nota anche come memoria virtuale).
- Memoria buffer: l'area dati condivisa da dispositivi hardware o processi di programma che operano a velocità e priorità diverse.
- Memoria cache: archivia le istruzioni e i dati del programma che vengono utilizzati ripetutamente nel funzionamento dei programmi di cui probabilmente la CPU avrà bisogno successivamente.
- Memoria libera: memoria che non viene utilizzata affatto ed è prontamente disponibile. Il sistema può essere utilizzato in modo completamente gratuito quando necessario.
- Memoria inattiva: pagine a cui non è stato effettuato l'accesso "di recente".

- Memoria totale: la dimensione della RAM di memoria fisica effettiva.
- Memoria usata: la memoria attualmente utilizzata da programmi e processi.

Argomenti

Linux: parametri raccolti e calcoli utilizzati

Parametri raccolti e unità:

- Attivo (byte)
- Disponibile (byte)
- Percentuale disponibile (percentuale)
- Memorizzato nel buffer (byte)
- Memorizzato nella cache (byte)
- Gratuito (byte)
- Attivo (byte)
- Totale (byte)
- Utilizzato (byte)
- Percentuale utilizzata (percentuale)

Memoria utilizzata = Memoria totale - Memoria libera - Memoria cache - Memoria buffer

Memoria totale = Memoria utilizzata + Memoria libera + Memoria cache + Memoria buffer

macOS: parametri raccolti e calcoli utilizzati

Parametri raccolti e unità:

- Attivo (byte)
- Disponibile (byte)

- Percentuale disponibile (percentuale)
- Gratuito (byte)
- Attivo (byte)
- Totale (byte)
- Utilizzato (byte)
- Percentuale utilizzata (percentuale)

Memoria disponibile = Memoria libera + Memoria inattiva

Memoria utilizzata = Memoria totale - Memoria disponibile

Memoria totale = Memoria disponibile + Memoria utilizzata

Windows: parametri raccolti

I parametri raccolti sugli host Windows sono riportati di seguito. Tutti questi parametri hanno `None` per `Unit`.

- Byte disponibili
- Errori di cache/sec
- Errori di pagina/sec
- Pagine/sec

Non vengono utilizzati calcoli per le metriche di Windows perché l' CloudWatch agente analizza gli eventi dai contatori delle prestazioni.

Esempio: calcolo dei parametri di memoria su Linux

Ad esempio, supponiamo che l'immissione del comando **`cat /proc/meminfo`** su un host Linux mostri i seguenti risultati:

```
MemTotal:       3824388 kB
MemFree:         462704 kB
MemAvailable:   2157328 kB
Buffers:         126268 kB
Cached:         1560520 kB
SReclaimable:    289080 kB>
```

In questo esempio, l' CloudWatch agente raccoglierà i seguenti valori. Tutti i valori che l' CloudWatch agente raccoglie e riporta sono espressi in byte.

- `mem_total`: 3916173312 byte
- `mem_available`: 2209103872 byte (+ cache) MemFree
- `mem_free`: 473808896 byte
- `mem_cached`: 1893990400 byte (`cached` + `SReclaimable`)
- `mem_used`: 1419075584 byte (`MemTotal` – (`MemFree` + `Buffers` + (`Cached` + `SReclaimable`)))
- `mem_buffered`: 129667072 byte
- `mem_available_percent`: 56,41%
- `mem_used_percent`: 36,24% (`mem_used` / `mem_total`) * 100

Scenari comuni CloudWatch con l'agente

Le sezioni seguenti descrivono come completare le attività di configurazione e personalizzazione comuni per l'agente. CloudWatch

Argomenti

- Esecuzione dell' CloudWatch agente come utente diverso
- In che modo l' CloudWatch agente gestisce i file di registro sparsi
- Aggiungere dimensioni personalizzate alle metriche raccolte dall'agente CloudWatch
- File di configurazione di più CloudWatch agenti
- Aggregazione o aggregazione delle metriche raccolte dall'agente CloudWatch
- Raccolta di metriche ad alta risoluzione con l'agente CloudWatch
- Invio di parametri, log e tracce a un altro account
- Differenze nel timestamp tra l'agente unificato e il precedente CloudWatch agente Logs CloudWatch

Esecuzione dell' CloudWatch agente come utente diverso

Sui server Linux, CloudWatch viene eseguito come utente root per impostazione predefinita. Per fare in modo che l'agente venga eseguito come utente diverso, utilizzate il `run_as_user` parametro nella

`agent` sezione del file di configurazione dell' CloudWatch agente. Questa opzione è disponibile solo su server Linux.

Se stai già eseguendo l'agente con l'utente root e desideri cambiare per utilizzare un utente diverso, utilizza una delle procedure seguenti.

Per eseguire l' CloudWatch agente come utente diverso su un'istanza EC2 che esegue Linux

1. Scarica e installa un nuovo pacchetto di CloudWatch agenti. Per ulteriori informazioni, consulta Scarica il pacchetto dell' CloudWatch agente.
2. Creare un nuovo utente Linux o utilizzare l'utente predefinito denominato `cwagent` creato dal file RPM o DEB.
3. Fornire le credenziali per questo utente in uno dei seguenti modi:
   - Se il file `.aws/credentials` esiste nella home directory dell'utente root, è necessario creare un file di credenziali per l'utente che verrà utilizzato per eseguire l' CloudWatch agente. Questo file di credenziali sarà `/home/`*`username`*`/.aws/credentials`. Quindi impostare il valore del parametro `shared_credential_file` in `common-config.toml` sul nome percorso del file delle credenziali. Per ulteriori informazioni, consulta la pagina (Opzionale) Modifica della configurazione comune delle informazioni relative al proxy o alla regione.
   - Se il file `.aws/credentials` non esiste nella directory home dell'utente root, puoi procedere in uno dei seguenti modi:
     - Crea un file di credenziali per l'utente che utilizzerai per eseguire l' CloudWatch agente. Questo file di credenziali sarà `/home/`*`username`*`/.aws/credentials`. Quindi impostare il valore del parametro `shared_credential_file` in `common-config.toml` sul nome percorso del file delle credenziali. Per ulteriori informazioni, consulta la pagina (Opzionale) Modifica della configurazione comune delle informazioni relative al proxy o alla regione.
     - Anziché creare un file di credenziali, collega un ruolo IAM all'istanza. L'agente usa questo ruolo come provider di credenziali.
4. Nel file di configurazione CloudWatch dell'agente, aggiungi la seguente riga nella `agent` sezione:

```
"run_as_user": "username"
```

Apportare altre modifiche al file di configurazione in base alle esigenze. Per ulteriori informazioni, consulta la pagina Creare il file di configurazione CloudWatch dell'agente

5. Concedi all'utente le autorizzazioni richieste. L'utente deve disporre delle autorizzazioni Read (r) per i file di log da raccogliere e deve disporre dell'autorizzazione Execute (x) per ogni directory nel percorso dei file di log.
6. Avviare l'agente con il file di configurazione modificato.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Per eseguire l' CloudWatch agente come utente diverso su un server locale che esegue Linux

1. Scarica e installa un nuovo pacchetto di CloudWatch agenti. Per ulteriori informazioni, consulta Scarica il pacchetto dell' CloudWatch agente.
2. Creare un nuovo utente Linux o utilizzare l'utente predefinito denominato `cwagent` creato dal file RPM o DEB.
3. Archiviare le credenziali di questo utente in un percorso a cui l'utente può accedere, ad esempio `/home/username/.aws/credentials`.
4. Impostare il valore del parametro `shared_credential_file` in `common-config.toml` sul nome percorso del file delle credenziali. Per ulteriori informazioni, consulta (Opzionale) Modifica della configurazione comune delle informazioni relative al proxy o alla regione.
5. Nel file di configurazione CloudWatch dell'agente, aggiungi la seguente riga nella `agent` sezione:

```
"run_as_user": "username"
```

Apportare altre modifiche al file di configurazione in base alle esigenze. Per ulteriori informazioni, consulta la pagina Creare il file di configurazione CloudWatch dell'agente

6. Concedi all'utente le autorizzazioni richieste. L'utente deve disporre delle autorizzazioni Read (r) per i file di log da raccogliere e deve disporre dell'autorizzazione Execute (x) per ogni directory nel percorso dei file di log.
7. Avviare l'agente con il file di configurazione modificato.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

In che modo l' CloudWatch agente gestisce i file di registro sparsi

I file sparse sono file con blocchi vuoti e contenuti reali. Un file sparse utilizza lo spazio su disco in modo più efficiente scrivendo brevi informazioni che rappresentano i blocchi vuoti sul disco al posto dei byte nulli effettivi che costituiscono il blocco. Ciò rende la dimensione effettiva di un file sparse in genere molto più piccola della sua dimensione apparente.

Tuttavia, l' CloudWatch agente non tratta i file sparsi in modo diverso rispetto ai file normali. Quando l'agente legge un file sparse, i blocchi vuoti vengono trattati come blocchi "reali" con i byte nulli. Per questo motivo, l' CloudWatch agente pubblica tanti byte quanti sono le dimensioni apparenti di un file sparso. CloudWatch

La configurazione dell' CloudWatch agente per la pubblicazione di un file sparso può comportare CloudWatch costi superiori al previsto, pertanto si consiglia di non farlo. Ad esempio, `/var/logs/lastlog` in Linux di solito si tratta di un file molto scarso e si consiglia di non pubblicarlo su. CloudWatch

Aggiungere dimensioni personalizzate alle metriche raccolte dall'agente CloudWatch

Per aggiungere dimensioni personalizzate, ad esempio tag ai parametri raccolti dall'agente, aggiungi il campo `append_dimensions` nella sezione del file di configurazione dell'agente che elenca i parametri.

Ad esempio, la seguente sezione di esempio del file di configurazione aggiunge una dimensione personalizzata denominata `stackName` con un valore di `Prod` ai parametri `cpu` e `disk` raccolti dall'agente.

```
"cpu":{
  "resources":[
    "*"
  ],
  "measurement":[
    "cpu_usage_guest",
    "cpu_usage_nice",
```

```
    "cpu_usage_idle"
  ],
  "totalcpu":false,
  "append_dimensions":{
    "stackName":"Prod"
  }
},
"disk":{
  "resources":[
    "/",
    "/tmp"
  ],
  "measurement":[
    "total",
    "used"
  ],
  "append_dimensions":{
    "stackName":"Prod"
  }
}
```

Ogni volta che modifichi il file di configurazione dell'agente, dovrai riavviare l'agente per implementare le modifiche.

File di configurazione di più CloudWatch agenti

Sia sui server Linux che sui server Windows, è possibile configurare l' CloudWatch agente per utilizzare più file di configurazione. Ad esempio, puoi utilizzare un file di configurazione comune che raccoglie un set di parametri e i log che desideri raccogliere da tutti i server dell'infrastruttura. Potrai quindi possibile utilizzare altri file di configurazione che raccolgono i parametri da determinate applicazioni o in determinate situazioni.

Per effettuare tale configurazione, dovrai innanzitutto creare i file di configurazione da utilizzare. I file di configurazione che verranno utilizzati insieme nello stesso server devono avere nomi diversi. È possibile archiviare i file di configurazione nei server o in Parameter Store.

Avviate l' CloudWatch agente utilizzando l'`fetch-config`opzione e specificate il primo file di configurazione. Per aggiungere il secondo file di configurazione all'agente in esecuzione, utilizzare lo stesso comando, ma con l'opzione `append-config`. Verranno raccolti tutti i parametri, i log e le tracce elencati in entrambi i file di configurazione. I seguenti comandi di esempio illustrano questo scenario utilizzando gli archivi di configurazione come file. La prima riga avvia l'agente utilizzando il

file di configurazione `infrastructure.json`, mentre la seconda aggiunge il file di configurazione `app.json`.

I seguenti comandi di esempio sono per Linux.

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2
 -s -c file:/tmp/infrastructure.json
```

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a append-config -m
 ec2 -s -c file:/tmp/app.json
```

I comandi di esempio seguenti sono per Windows Server.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\infrastructure.json"
```

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a append-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\app.json"
```

I seguenti file di configurazione di esempio mostrano l'utilizzo di questa funzione. Il primo file di configurazione viene utilizzato per tutti i server dell'infrastruttura, mentre il secondo raccoglie solo i log da una determinata applicazione e viene aggiunto ai server che eseguono tale applicazione.

infrastructure.json

```
{
  "metrics": {
    "metrics_collected": {
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true
      },
      "mem": {
         "measurement": [
```

```
            "used_percent"
        ]
      }
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
            "log_group_name": "amazon-cloudwatch-agent.log"
          },
          {
            "file_path": "/var/log/messages",
            "log_group_name": "/var/log/messages"
          }
        ]
      }
    }
  }
}
```

app.json

```
{
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/app/app.log*",
                        "log_group_name": "/app/app.log"
                    }
                ]
            }
        }
    }
}
```

Tutti i file di configurazione aggiunti alla configurazione devono avere nomi diversi fra loro e rispetto al file di configurazione iniziale. Se usi `append-config` con un file di configurazione con lo stesso

nome di quello già in uso dall'agente, il comando di aggiunta sovrascriverà le informazioni del primo file di configurazione, anziché aggiungerle. Ciò vale anche se i due file di configurazione con lo stesso nome si trovano in percorsi diversi.

L'esempio precedente mostra l'uso di due file di configurazione, ma non vi è alcun limite al numero di file di configurazione che puoi aggiungere all'agente di configurazione. Puoi anche combinare l'uso di file di configurazione che si trovano nei server e le configurazioni situate in Parameter Store.

Aggregazione o aggregazione delle metriche raccolte dall'agente CloudWatch

Per aggregare o eseguire il rollup dei parametri raccolti dall'agente, aggiungi un campo `aggregation_dimensions` alla sezione relativa a tale parametro nel file di configurazione dell'agente.

Ad esempio, il seguente frammento del file di configurazione esegue il rollup dei parametri sulla dimensione `AutoScalingGroupName`. Vengono aggregati i parametri da tutte le istanze in ogni gruppo Auto Scaling e possono essere visualizzati complessivamente.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"]]
}
```

Per eseguire il rollup in base alla combinazione di ogni dimensione `InstanceId` e `InstanceType` oltre al rollup nel nome del gruppo Auto Scaling, aggiungi quanto segue.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId", "InstanceType"]]
}
```

Per eseguire il rollup dei parametri in un'unica raccolta, invece, utilizza `[]`.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
```

```
  "aggregation_dimensions" : [[]]
}
```

Ogni volta che modifichi il file di configurazione dell'agente, dovrai riavviare l'agente per implementare le modifiche.

Raccolta di metriche ad alta risoluzione con l'agente CloudWatch

Il campo `metrics_collection_interval` specifica l'intervallo di tempo per i parametri raccolti, in secondi. Specificando un valore inferiore a 60 per questo campo, i parametri vengono raccolti come i parametri ad alta risoluzione.

Ad esempio, se i parametri devono essere tutti ad alta risoluzione e raccolti ogni 10 secondi, specifica 10 come valore di `metrics_collection_interval` nella sezione `agent` come intervallo di raccolta dei parametri globale.

```
"agent": {
  "metrics_collection_interval": 10
}
```

In alternativa, il seguente esempio imposta i parametri `cpu` in modo che siano raccolti ogni secondo, mentre tutti gli altri parametri vengono raccolti ogni minuto.

```
"agent":{
  "metrics_collection_interval": 60
},
"metrics":{
  "metrics_collected":{
    "cpu":{
      "resources":[
        "*"
      ],
      "measurement":[
        "cpu_usage_guest"
      ],
      "totalcpu":false,
      "metrics_collection_interval": 1
    },
    "disk":{
      "resources":[
        "/",
```

```
        "/tmp"
      ],
      "measurement":[
        "total",
        "used"
      ]
    }
  }
}
```

Ogni volta che modifichi il file di configurazione dell'agente, dovrai riavviare l'agente per implementare le modifiche.

Invio di parametri, log e tracce a un altro account

Per fare in modo che l' CloudWatch agente invii le metriche, i log o le tracce a un account diverso, specifica un `role_arn` parametro nel file di configurazione dell'agente sul server di invio. Il valore `role_arn` specifica un ruolo IAM nell'account di destinazione che l'agente usa durante l'invio di dati a tale account. Questo ruolo consente all'account di invio di assumere un ruolo corrispondente nell'account di destinazione quando si distribuiscono i parametri o i log all'account di destinazione.

È anche possibile specificare stringhe `role_arn` separate nel file di configurazione dell'agente: uno da utilizzare quando si inviano i parametri e uno per l'invio delle tracce.

L'esempio seguente di parte della sezione `agent` del file di configurazione imposta l'agente in modo da utilizzare `CrossAccountAgentRole` per l'invio di dati a un altro account.

```
{
  "agent": {
    "credentials": {
      "role_arn": "arn:aws:iam::123456789012:role/CrossAccountAgentRole"
    }
  },
  .....
}
```

In alternativa, l'esempio seguente imposta ruoli diversi per l'account di invio da utilizzare per l'invio di parametri, log e tracce:

```
"metrics": {
    "credentials": {
     "role_arn": "RoleToSendMetrics"
    },
    "metrics_collected": {....
```

```
"logs": {
    "credentials": {
    "role_arn": "RoleToSendLogs"
    },
    ....
```

Policy richieste

Quando si specifica un `role_arn` nel file di configurazione dell'agente, è anche necessario accertarsi che i ruoli IAM degli account di invio e di destinazione abbiano determinate policy. I ruoli in entrambi gli account di invio e di destinazione devono avere `CloudWatchAgentServerPolicy`. Per ulteriori informazioni sull'assegnazione di questa policy a un ruolo, consulta Crea ruoli IAM da utilizzare con l' CloudWatch agente sulle istanze Amazon EC2.

Il ruolo nell'account di invio, inoltre, deve includere la seguente policy. Aggiungere questa policy alla scheda Permissions (Autorizzazioni) nella console IAM quando si modifica il ruolo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::target-account-ID:role/agent-role-in-target-account"
            ]
        }
    ]
}
```

Il ruolo nell'account di destinazione deve includere la seguente policy, in modo che riconosca il ruolo IAM utilizzato dall'account di invio. Aggiungi questa policy alla scheda Trust relationships (Relazioni

di trust) nella console IAM quando modifichi il ruolo. Il ruolo nell'account di destinazione in cui si aggiunge questa policy è il ruolo creato in Crea ruoli e utenti IAM da utilizzare con l' CloudWatch agente. Questo ruolo è il ruolo specificato in *agent-role-in-target-account* nella policy utilizzata dall'account di invio.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::sending-account-ID:role/role-in-sender-account"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Differenze nel timestamp tra l'agente unificato e il precedente CloudWatch agente Logs CloudWatch

L' CloudWatch agente supporta un set diverso di simboli per i formati di timestamp rispetto al precedente agente Logs. CloudWatch Tali differenze sono mostrate nella seguente tabella.

Simboli supportati da entrambi gli agenti	Simboli supportati solo dall'agente unificato CloudWatch	Simboli supportati solo dall'agente CloudWatch Logs precedente
%A, %a, %b, %B, %d, %f, %H, %I, %m, %M, %p, %S, %y, %Y, %Z, %z	%-d, %-I, %-m, %-M, %-S	%c, %j, %U, %W, %w

Per ulteriori informazioni sul significato dei simboli supportati dal nuovo CloudWatch agente, consulta la sezione CloudWatch Agent Configuration File: Logs nella Amazon CloudWatch User Guide. Per informazioni sui simboli supportati dall'agente CloudWatch Logs, consulta il file di configurazione dell'agente nella Amazon CloudWatch Logs User Guide.

Risoluzione dei problemi relativi all'agente CloudWatch

Utilizza le seguenti informazioni per risolvere i problemi relativi all' CloudWatch agente.

Argomenti

CloudWatch parametri della riga di comando dell'agente

Per visualizzare l'elenco completo dei parametri supportati dall' CloudWatch agente, inserisci quanto segue nella riga di comando del computer in cui è installato:

```
amazon-cloudwatch-agent-ctl -help
```

L'installazione dell' CloudWatch agente tramite Run Command non riesce

Per installare l' CloudWatch agente utilizzando Systems Manager Run Command, l'agente SSM sul server di destinazione deve essere la versione 2.2.93.0 o successiva. Se la versione di SSM Agent non è corretta, potrebbero venire visualizzati errori che includono i seguenti messaggi:

```
no latest version found for package AmazonCloudWatchAgent on platform linux
```

```
failed to download installation package reliably
```

Per informazioni sull'installazione o sull'aggiornamento di SSM Agent consulta la pagina relativa all'installazione e alla configurazione di SSM Agent nella Guida per l'utente AWS Systems Manager .

L'agente non si avvia CloudWatch

Se l' CloudWatch agente non si avvia, potrebbe esserci un problema nella configurazione. Le informazioni sulla configurazione sono registrate nel file `configuration-validation.log`. Il percorso del file è `/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` nei server Linux e `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log` nei server che eseguono Windows Server.

Verifica che l' CloudWatch agente sia in esecuzione

Puoi interrogare l' CloudWatch agente per scoprire se è in esecuzione o è fermo. Per eseguire questa operazione in remoto, puoi utilizzare AWS Systems Manager . Puoi inoltre utilizzare la riga di comando, ma solo per controllare il server locale.

Per interrogare lo stato dell' CloudWatch agente utilizzando Run Command

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione seleziona Run Command.

   oppure

   Se la AWS Systems Manager home page si apre, scorri verso il basso e scegli Esplora Run Command.
3. Seleziona Run command (Esegui comando).
4. Nell'elenco dei documenti Command, scegli il pulsante accanto a AmazonCloudWatch-ManageAgent.
5. Nell'elenco Action (Operazione), seleziona status (stato).
6. Per Origine configurazione facoltativa scegli il valore predefinito e non compilare il campo del percorso della configurazione facoltativa.

7. Nell'area Target (Destinazione), seleziona l'istanza da controllare.
8. Seleziona Esegui.

Se l'agente è in esecuzione, l'output sarà analogo al seguente.

```
{
       "status": "running",
       "starttime": "2017-12-12T18:41:18",
       "version": "1.73.4"
}
```

Se l'agente viene arrestato, nel campo `"status"` viene visualizzato `"stopped"`.

Per interrogare lo stato dell' CloudWatch agente localmente utilizzando la riga di comando

- In un server Linux, immetti quanto segue:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   status
  ```

  Su un server che esegue Windows Server, inserisci quanto segue PowerShell come amministratore:

  ```
  & $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
   ec2 -a status
  ```

L' CloudWatch agente non si avvia e l'errore indica una regione Amazon EC2

Se l'agente non viene avviato e il messaggio di errore fa riferimento all'endpoint della regione Amazon EC2, potresti aver configurato l'agente in modo da dover accedere all'endpoint Amazon EC2, senza avere concesso le autorizzazioni di accesso.

Ad esempio, se specifichi un valore del parametro `append_dimensions` nel file di configurazione dell'agente che dipende da metadati Amazon EC2 e utilizzi proxy, devi assicurarti che il server possa accedere all'endpoint per Amazon EC2. Per ulteriori informazioni su questi endpoint, consulta Amazon Elastic Compute Cloud (Amazon EC2) nella Riferimenti generali di Amazon Web Services.

L' CloudWatch agente non si avvierà su Windows Server

In Windows Server, potresti visualizzare il seguente errore:

```
Start-Service : Service 'Amazon CloudWatch Agent (AmazonCloudWatchAgent)' cannot be
 started due to the following
error: Cannot start service AmazonCloudWatchAgent on computer '.'.
At C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1:113
 char:12
+     $svc | Start-Service
+            ~~~~~~~~~~~~~
    + CategoryInfo          : OpenError:
 (System.ServiceProcess.ServiceController:ServiceController) [Start-Service],
   ServiceCommandException
    + FullyQualifiedErrorId :
 CouldNotStartService,Microsoft.PowerShell.Commands.StartServiceCommand
```

Per risolvere questo problema, assicurati innanzitutto che il servizio server sia in esecuzione. Questo errore può essere visualizzato se l'agente tenta di avviarsi quando il servizio server non è in esecuzione.

Se il servizio server è già in esecuzione, potrebbe essere il problema seguente. In alcune installazioni di Windows Server, l'avvio dell' CloudWatch agente impiega più di 30 secondi. Poiché Windows Server, per impostazione predefinita, consente solo 30 secondi per l'avvio dei servizi, questo causa l'errore dell'agente simile al seguente:

Per risolvere questo problema, aumenta il valore del timeout per il servizio. Per ulteriori informazioni, consulta Un servizio non viene avviato e gli eventi 7000 e 7011 vengono registrati nel log eventi di Windows.

Dove sono i parametri?

Se l' CloudWatch agente è in esecuzione ma non riesci a trovare le metriche da esso raccolte nel AWS Management Console o nel AWS CLI, conferma che stai utilizzando lo spazio dei nomi corretto. Per impostazione predefinita, lo spazio dei nomi dei parametri raccolti dall'agente è `CWAgent`. Puoi personalizzare questo spazio dei nomi utilizzando il campo `namespace` nella sezione `metrics` del file di configurazione dell'agente. Se non vengono visualizzati i parametri previsti, controlla il file di configurazione per verificare lo spazio dei nomi in uso.

Quando scarichi per la prima volta il pacchetto dell' CloudWatch agente, il file di configurazione dell'agente è. `amazon-cloudwatch-agent.json` Il file è ubicato nella directory in cui hai eseguito

la procedura guidata di configurazione oppure potrebbe trovarsi in un'altra directory. Se utilizzi la procedura guidata di configurazione, l'output del file di configurazione dell'agente è denominato `config.json`. Per ulteriori informazioni sul file di configurazione, incluso il campo `namespace`, consulta CloudWatch file di configurazione dell'agente: sezione Metrics.

L' CloudWatch agente impiega molto tempo per essere eseguito in un contenitore o registra un errore di limite di hop

Quando esegui l' CloudWatch agente come servizio container e desideri aggiungere le dimensioni dei parametri di Amazon EC2 a tutti i parametri raccolti dall'agente, potresti visualizzare i seguenti errori nella versione v1.247354.0 dell'agente:

```
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Unable to retrieve Instance
 Metadata Tags. This plugin must only be used on an EC2 instance.
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Please increase hop limit
 to 2 by following this document https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/
configuring-instance-metadata-options.html#configuring-IMDS-existing-instances.
2022-06-07T03:36:11Z E! [telegraf] Error running agent: could not initialize processor
 ec2tagger: EC2MetadataRequestError: failed to get EC2 instance identity document
caused by: EC2MetadataError: failed to make EC2Metadata request
        status code: 401, request id:
caused by:
```

È possibile che venga visualizzato questo errore se l'agente tenta di ottenere metadati da IMDSv2 all'interno di un container senza un limite di hop appropriato. Nelle versioni dell'agente precedenti alla v1.247354.0, è possibile riscontrare questo problema senza visualizzare il messaggio di log.

Per risolvere questo problema, aumentare il limite di hop a 2 seguendo le istruzioni riportate in Configurazione delle opzioni dei metadati dell'istanza.

Ho aggiornato la configurazione del mio agente ma non vedo le nuove metriche o i nuovi log nella console CloudWatch

Se aggiorni il file di configurazione dell' CloudWatch agente, al successivo avvio dell'agente, devi utilizzare l'**`fetch-config`**opzione. Ad esempio, se il file aggiornato è stato memorizzato nel computer locale, immetti il comando seguente:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
s -m ec2 -c file:configuration-file-path
```

CloudWatch file e posizioni degli agenti

La tabella seguente elenca i file installati e utilizzati con l' CloudWatch agente, insieme alle relative posizioni sui server che eseguono Linux o Windows Server.

File	Ubicazione in Linux	Ubicazione in Windows Server
Lo script di controllo che controlla l'avvio, l'arresto e il riavvio dell'agente.	`/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl` o `/usr/bin/amazon-cloudwatch-agent-ctl`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1`
Il file di log su cui scrive l'agente. Potrebbe essere necessario allegarlo quando si contatta AWS Support.	`/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` o `/var/log/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log`
File di convalida della configurazione dell'agente.	`/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` o `/var/log/amazon/amazon-cloudwatch-agent/configuration-validation.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log`
File JSON utilizzato per configurare l'agente, immediatamente dopo la sua creazione tramite la procedura guidata. Per ulteriori informazi	`/opt/aws/amazon-cloudwatch-agent/bin/config.json`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\config.json`

File	Ubicazione in Linux	Ubicazione in Windows Server
oni, consulta la pagina Creare il file di configurazione CloudWatch dell'agente.		
File JSON utilizzato per configurare l'agente, se questo file di configurazione è stato scaricato da Parameter Store.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` o `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json`
Il file TOML utilizzato per specificare le informazioni sulla Regione e sulle credenziali che l'agente deve utilizzare, sostituendo le impostazioni predefinite di sistema.	`/opt/aws/amazon-cloudwatch-agent/etc/common-config.toml` o `/etc/amazon/amazon-cloudwatch-agent/common-config.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\common-config.toml`
File TOML che contiene il contenuto convertito del file di configurazione JSON. Lo script `amazon-cloudwatch-agent-ctl` genera questo file. Gli utenti non devono modificare direttamente questo file. Può essere utile per verificare che la traduzione da JSON a TOML abbia avuto successo.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml` o `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.toml`

File	Ubicazione in Linux	Ubicazione in Windows Server
Il file YAML che contiene il contenuto convertito del file di configurazione JSON. Lo script `amazon-cloudwatch-agent-ctl` genera questo file. Questo file non deve essere modificato direttamente. Può essere utile per verificare che la traduzione da JSON a YAML abbia avuto successo.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.yaml or /etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.yaml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.yaml`

Ricerca di informazioni sulle versioni degli CloudWatch agenti

Per trovare il numero di versione dell' CloudWatch agente su un server Linux, immettete il seguente comando:

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status
```

Per trovare il numero di versione dell' CloudWatch agente su Windows Server, inserisci il seguente comando:

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m ec2
 -a status
```

> **Note**
>
> L'utilizzo di questo comando è il modo corretto per trovare la versione dell' CloudWatch agente. Se utilizzi Programs and Features (Programmi e funzionalità) nel Pannello di controllo, verrà visualizzato un numero di versione errato.

Puoi anche scaricare un file README relativo alle modifiche più recenti all'agente e un file che indica il numero di versione attualmente disponibile per il download. Questi file si trovano nelle seguenti posizioni:

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` o `https://amazoncloudwatch-agent-region.s3.region.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` o `https://amazoncloudwatch-agent-region.s3.region.amazonaws.com/info/latest/CWAGENT_VERSION`

Registri generati dall'agente CloudWatch

L'agente genera un log durante la sua esecuzione. Questo log include le informazioni relative alla risoluzione dei problemi. Questo log è il file `amazon-cloudwatch-agent.log`. Il percorso del file è `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` nei server Linux e `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log` nei server che eseguono Windows Server.

Puoi configurare l'agente per registrare dettagli aggiuntivi nel file `amazon-cloudwatch-agent.log`. Nel file di configurazione dell'agente, nella `agent` sezione, imposta il `debug` campo su`true`, quindi riconfigura e riavvia l' CloudWatch agente. Per disattivare la registrazione di queste informazioni supplementari, imposta il campo `debug` su `false`. Quindi, riconfigura e riavvia l'agente. Per ulteriori informazioni, consulta Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.

Nelle versioni 1.247350.0 e successive dell' CloudWatch agente, puoi facoltativamente impostare il `aws_sdk_log_level` campo nella `agent` sezione del file di configurazione dell'agente su una o più delle seguenti opzioni. Separare più opzioni con il carattere `|`.

- `LogDebug`
- `LogDebugWithSigning`
- `LogDebugWithHTTPBody`
- `LogDebugRequestRetries`
- `LogDebugWithEventStreamBody`

Per ulteriori informazioni su queste opzioni, vedere. LogLevelType

Arresto e riavvio dell'agente CloudWatch

È possibile arrestare manualmente l' CloudWatch agente utilizzando una delle due AWS Systems Manager o la riga di comando.

Per arrestare l' CloudWatch agente, utilizzare Run Command

1. Aprire la console Systems Manager all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel riquadro di navigazione seleziona Run Command.

   oppure

   Se la AWS Systems Manager home page si apre, scorri verso il basso e scegli Esplora Run Command.
3. Seleziona Run command (Esegui comando).
4. Nell'elenco dei documenti Command, scegli AmazonCloudWatch- ManageAgent.
5. Nell'area Target, scegliete l'istanza in cui avete installato l' CloudWatch agente.
6. Nell'elenco Action (Operazione), seleziona stop (arresta).
7. Non compilare i campi Optional Configuration Source (Origine configurazione opzionale) e Optional Configuration Location (Percorso configurazione opzionale).
8. Seleziona Esegui.

Per arrestare l' CloudWatch agente localmente utilizzando la riga di comando

- In un server Linux, immetti quanto segue:

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   stop
  ```

  Su un server che esegue Windows Server, inserisci quanto segue PowerShell come amministratore:

  ```
  & $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
   ec2 -a stop
  ```

Per riavviare l'agente, segui le istruzioni descritte in Avvia l'agente CloudWatch .

Incorporamento dei parametri nei log

Il formato metrico CloudWatch incorporato consente di generare metriche personalizzate in modo asincrono sotto forma di log scritti in Logs. CloudWatch Puoi incorporare metriche personalizzate insieme a dati dettagliati sugli eventi di registro ed estrarre CloudWatch automaticamente le metriche personalizzate in modo da poterle visualizzare e generare allarmi in base a esse, per il rilevamento degli incidenti in tempo reale. Inoltre, gli eventi di registro dettagliati associati alle metriche estratte possono essere interrogati utilizzando CloudWatch Logs Insights per fornire informazioni approfondite sulle cause principali degli eventi operativi.

Il formato dei parametri incorporati consente di generare parametri personalizzati concreti a partire da risorse effimere ad esempio le funzioni Lambda e i container. Utilizzando Embedded Metric Format per inviare log da queste risorse effimere, potrai ora creare facilmente parametri personalizzati senza doverti dotare di strumenti o mantenere codice separato, ottenendo allo stesso tempo potenti funzionalità analitiche sui dati di log.

Non è richiesta alcuna configurazione per utilizzare il formato dei parametri incorporati. Strutturate i log seguendo le specifiche del formato metrico Embedded oppure generateli utilizzando le nostre librerie client e inviateli a CloudWatch Logs utilizzando l'API o l'agente. PutLogEvents CloudWatch

Per l'acquisizione e l'archiviazione di log viene addebitato un costo e vengono generati parametri personalizzati. Per ulteriori informazioni, consulta Prezzi di Amazon CloudWatch.

> **Note**
>
> Fai attenzione durante la configurazione dell'estrazione dei parametri poiché influenza l'utilizzo di parametri personalizzati e l'addebito corrispondente. Se crei involontariamente parametri basati su dimensioni ad alta cardinalità (ad esempio `requestId`), Embedded Metric Format creerà per impostazione predefinita un parametro personalizzato corrispondente a ogni combinazione di dimensione univoca. Per ulteriori informazioni, consulta Dimensioni.

Argomenti

- Impostazione degli allarmi sui parametri creati con il formato dei parametri incorporati

Pubblicazione di log con il formato dei parametri incorporati

Puoi generare log in Embedded Metric Format con i seguenti metodi:

- Genera e invia i log utilizzando le librerie client open-source.
- Genera manualmente i log utilizzando la specifica del formato metrico incorporato, quindi utilizza l'CloudWatch agente o l'PutLogEvents API per inviare i log.

Argomenti

- Creazione di log in formato dei parametri incorporati utilizzando le librerie client
- Specifica: Embedded Metric Format
- Utilizzo dell' PutLogEventsAPI per inviare log in formato metrico incorporato creati manualmente
- Utilizzo dell' CloudWatch agente per inviare log in formato metrico incorporato
- Utilizzo del formato metrico incorporato con AWS Distro per OpenTelemetry

Creazione di log in formato dei parametri incorporati utilizzando le librerie client

Amazon fornisce librerie client open-source che puoi utilizzare per creare log in Embedded Metric Format. Attualmente tali librerie sono disponibili per i linguaggi dell'elenco seguente. Esempi completi per diverse configurazioni sono disponibili nelle nostre librerie client in /examples.

Le librerie e le istruzioni su come usarle si trovano su Github. Utilizza i seguenti link.

- Node.Js

> **Note**
>
> Per Node.js, per l'uso con il formato di log JSON Lambda sono richieste le versioni 4.1.1 e successive, 3.0.2 e successive, 2.0.7 e successive. L'utilizzo di versioni precedenti in tali ambienti Lambda comporterà una perdita di parametri.
> Per ulteriori informazioni, consulta Accedere ai CloudWatch log di Amazon per AWS Lambda.

- Python
- Java
- C#

Le librerie client sono pensate per funzionare immediatamente con l' CloudWatch agente. I log generati in formato metrico incorporato vengono inviati all' CloudWatch agente, che quindi li aggrega CloudWatch e li pubblica in Logs for you.

> **Note**
>
> Quando si utilizza Lambda, non è necessario alcun agente a cui inviare i log. CloudWatch Tutto ciò che viene registrato su STDOUT viene inviato ai CloudWatch registri tramite il Lambda Logging Agent.

Specifica: Embedded Metric Format

Il formato metrico CloudWatch incorporato è una specifica JSON utilizzata per indicare ai CloudWatch log di estrarre automaticamente i valori metrici incorporati negli eventi di registro strutturati. È possibile utilizzarlo CloudWatch per rappresentare graficamente e creare allarmi sui valori metrici estratti.

Convenzioni specifiche di Embedded Metric Format

Le parole chiave “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY” e “OPTIONAL” in questa specifica di formato devono essere interpretate come descritto in Key Words RFC2119.

I termini «JSON», «testo JSON», «valore JSON», «member», «element», «object», «array», «number», «string», «boolean», «true», «false» e «null» in questa specifica di formato devono essere interpretati come definito in JavaScript Object Notation RFC8259.

> **Note**
>
> Se prevedi di creare allarmi su parametri creati utilizzando il formato dei parametri incorporati, consulta Impostazione degli allarmi sui parametri creati con il formato dei parametri incorporati per ottenere dei suggerimenti.

Struttura del documento in Embedded Metric Format

In questa sezione viene descritta la struttura di un documento in formato del parametro integrato. I documenti in formato metrico incorporato sono definiti in JavaScript Object Notation RFC8259.

Salvo ove diversamente specificato, gli oggetti definiti da questa specifica NON DEVONO contenere alcun membro aggiuntivo. I membri non riconosciuti da questa specifica DEVONO essere ignorati. I membri definiti in questa specifica rispettano la distinzione tra maiuscole e minuscole.

Il formato metrico incorporato è soggetto agli stessi limiti degli eventi CloudWatch Logs standard e ha una dimensione massima di 256 KB.

Con la specifica Embedded Metric Format, puoi tenere traccia dell'elaborazione dei registri EMF in base ai parametri pubblicati nello spazio dei nomi `AWS/Logs` del tuo account. Questi possono essere utilizzati per tenere traccia della generazione di parametri con esito negativo per EMF e per verificare se gli errori sono dovuti all'analisi o alla convalida. Per maggiori dettagli, consulta Monitoraggio con metriche CloudWatch .

Nodo radice

Il LogEvent messaggio DEVE essere un oggetto JSON valido senza dati aggiuntivi all'inizio o alla fine della stringa del LogEvent messaggio. Per ulteriori informazioni sulla LogEvent struttura, vedere InputLogEvent.

I documenti in Embedded Metric Format DEVONO contenere il seguente membro di primo livello sul nodo principale. Questo è un oggetto Oggetto metadati.

```
{
 "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
}
```

Il nodo principale DEVE contenere tutti i membri Membri di destinazione definiti dai riferimenti nel MetricDirective oggetto.

Il nodo principale PUÒ contenere tutti gli altri membri che non sono inclusi nei requisiti precedenti. I valori di questi membri DEVONO essere tipi JSON validi.

Oggetto metadati

Il `_aws` membro può essere utilizzato per rappresentare i metadati relativi al payload che indicano ai servizi a valle come devono elaborare il. LogEvent Il valore DEVE essere un oggetto e DEVE contenere i seguenti membri:

- CloudWatchMetrics— Una matrice MetricDirective oggetto utilizzata per indicare di estrarre CloudWatch le metriche dal nodo radice di. LogEvent

```
{
  "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
}
```

- Time stamp Un numero che rappresenta la marca temporale usata per i parametri estratti dall'evento. I valori DEVONO essere espressi come il numero di millisecondi dopo il 1 gennaio 1970 00:00:00 UTC.

```
{
  "_aws": {
    "Timestamp": 1559748430481
  }
}
```

MetricDirective oggetto

L' MetricDirective oggetto indica ai servizi a valle che LogEvent contengono le metriche su cui verranno estratte e pubblicate. CloudWatch MetricDirectives DEVE contenere i seguenti membri:

- Namespace: una stringa che rappresenta lo spazio dei CloudWatch nomi per la metrica.
- Dimensioni Un DimensionSet matrice.
- Parametri Una matrice di oggetti MetricDefinition. Questo array NON DEVE contenere più di 100 oggetti. MetricDefinition

DimensionSet matrice

A DimensionSet è una matrice di stringhe contenente le chiavi di dimensione che verranno applicate a tutte le metriche del documento. I valori all'interno di questa matrice DEVONO anche essere membri del nodo principale, definiti come Membri di destinazione

A NON DimensionSet DEVE contenere più di 30 chiavi dimensionali. A DimensionSet PUÒ essere vuoto.

Il membro di destinazione DEVE avere un valore stringa. Questo valore NON DEVE contenere più di 1024 caratteri. Il membro di destinazione definisce una dimensione che verrà pubblicata nell'identità del parametro. Ogni DimensionSet elemento utilizzato crea una nuova metrica in CloudWatch. Per ulteriori informazioni sulle dimensioni, consulta Dimensione e Dimensioni.

```
{
 "_aws": {
   "CloudWatchMetrics": [
     {
       "Dimensions": [ [ "functionVersion" ] ],
       ...
     }
   ]
 },
 "functionVersion": "$LATEST"
}
```

> **Note**
>
> Fai attenzione durante la configurazione dell'estrazione dei parametri poiché influenza l'utilizzo di parametri personalizzati e l'addebito corrispondente. Se crei involontariamente parametri basati su dimensioni ad alta cardinalità (ad esempio `requestId`), Embedded Metric Format creerà per impostazione predefinita un parametro personalizzato corrispondente a ogni combinazione di dimensione univoca. Per ulteriori informazioni, consulta Dimensioni.

MetricDefinition oggetto

A MetricDefinition è un oggetto che DEVE contenere il seguente membro:

- Nome Una stringa Valori di riferimento in un parametro Membri di destinazione. Le destinazioni parametro DEVONO essere un valore numerico o una matrice di valori numerici.

Un MetricDefinition oggetto PUÒ contenere i seguenti membri:

- Unità Un valore stringa OPZIONALE che rappresenta l'unità di misura per il parametro corrispondente. I valori DEVONO essere unità CloudWatch metriche valide. Per informazioni sulle unità valide, vedere MetricDatum. Se non viene fornito un valore, viene assunto il valore predefinito NONE.
- StorageResolution— Un valore intero OPZIONALE che rappresenta la risoluzione di archiviazione per la metrica corrispondente. Se si imposta questo valore su 1, questa metrica viene specificata come metrica ad alta risoluzione, in modo che la metrica con una risoluzione inferiore al minuto CloudWatch venga memorizzata fino a un secondo. L'impostazione di questo valore su 60 specifica questa metrica come risoluzione standard, che viene memorizzata a una risoluzione di 1 minuto. CloudWatch I valori DEVONO essere risoluzioni CloudWatch supportate valide, 1 o 60. Se non viene fornito un valore, viene assunto il valore predefinito 60.

  Per ulteriori informazioni sui parametri ad alta risoluzione, consulta Parametri ad alta risoluzione.

> **Note**
>
> Se prevedi di creare allarmi su parametri creati utilizzando il formato dei parametri incorporati, consulta Impostazione degli allarmi sui parametri creati con il formato dei parametri incorporati per ottenere dei suggerimenti.

```
{
  "_aws": {
    "CloudWatchMetrics": [
      {
        "Metrics": [
          {
            "Name": "Time",
            "Unit": "Milliseconds",
            "StorageResolution": 60
          }
        ],
        ...
```

```
      }
    ]
  },
  "Time": 1
}
```

Valori di riferimento

I valori di riferimento sono valori stringa che fanno riferimento ai membri Membri di destinazione del nodo principale. Questi riferimenti NON devono essere confusi con i puntatori JSON descritti in RFC6901. I valori di destinazione non possono essere nidificati.

Membri di destinazione

Destinazioni valide DEVONO essere membri del nodo principale e non possono essere oggetti nidificati. Ad esempio, a _reference_ value di `"A.a"` DEVE corrispondere al seguente membro:

```
{ "A.a" }
```

NON DEVE corrispondere al membro nidificato:

```
{ "A": { "a" } }
```

I valori validi dei membri di destinazione dipendono da ciò che vi fa riferimento. Una destinazione parametro DEVE essere un valore numerico o una matrice di valori numerici. Le destinazioni di parametro di matrice numerica NON DEVONO avere più di 100 membri. Una destinazione di dimensione DEVE avere un valore stringa.

Esempio di Embedded Metric Format e schema JSON

Di seguito è riportato un esempio valido di Embedded Metric Format.

```
{
  "_aws": {
    "Timestamp": 1574109732004,
    "CloudWatchMetrics": [
      {
        "Namespace": "lambda-function-metrics",
        "Dimensions": [["functionVersion"]],
        "Metrics": [
          {
```

```
            "Name": "time",
            "Unit": "Milliseconds",
            "StorageResolution": 60
          }
        ]
      }
    ]
  },
  "functionVersion": "$LATEST",
  "time": 100,
  "requestId": "989ffbf8-9ace-4817-a57c-e4dd734019ee"
}
```

Puoi utilizzare lo schema seguente per convalidare i documenti in Embedded Metric Format.

```
{
    "type": "object",
    "title": "Root Node",
    "required": [
        "_aws"
    ],
    "properties": {
        "_aws": {
            "$id": "#/properties/_aws",
            "type": "object",
            "title": "Metadata",
            "required": [
                "Timestamp",
                "CloudWatchMetrics"
            ],
            "properties": {
                "Timestamp": {
                    "$id": "#/properties/_aws/properties/Timestamp",
                    "type": "integer",
                    "title": "The Timestamp Schema",
                    "examples": [
                        1565375354953
                    ]
                },
                "CloudWatchMetrics": {
                    "$id": "#/properties/_aws/properties/CloudWatchMetrics",
                    "type": "array",
                    "title": "MetricDirectives",
```

```
                    "items": {
                        "$id": "#/properties/_aws/properties/CloudWatchMetrics/items",
                        "type": "object",
                        "title": "MetricDirective",
                        "required": [
                            "Namespace",
                            "Dimensions",
                            "Metrics"
                        ],
                        "properties": {
                            "Namespace": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Namespace",
                                "type": "string",
                                "title": "CloudWatch Metrics Namespace",
                                "examples": [
                                    "MyApp"
                                ],
                                "pattern": "^(.*)$",
                                "minLength": 1,
                                "maxLength": 1024
                            },
                            "Dimensions": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Dimensions",
                                "type": "array",
                                "title": "The Dimensions Schema",
                                "minItems": 1,
                                "items": {
                                    "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items",
                                    "type": "array",
                                    "title": "DimensionSet",
                                    "minItems": 0,
                                    "maxItems": 30,
                                    "items": {
                                        "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items/items",
                                        "type": "string",
                                        "title": "DimensionReference",
                                        "examples": [
                                            "Operation"
                                        ],
                                        "pattern": "^(.*)$",
```

```
                                        "minLength": 1,
                                        "maxLength": 250
}
                                }
                            },
                            "Metrics": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Metrics",
                                "type": "array",
                                "title": "MetricDefinitions",
                                "items": {
                                    "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items",
                                    "type": "object",
                                    "title": "MetricDefinition",
                                    "required": [
                                        "Name"
                                    ],
                                    "properties": {
                                        "Name": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Name",
                                            "type": "string",
                                            "title": "MetricName",
                                            "examples": [
                                                "ProcessingLatency"
                                            ],
                                            "pattern": "^(.*)$",
                                            "minLength": 1,
                                            "maxLength": 1024
                                        },
                                        "Unit": {
                                            "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Unit",
                                            "type": "string",
                                            "title": "MetricUnit",
                                            "examples": [
                                                "Milliseconds"
                                            ],
                                            "pattern": "^(Seconds|Microseconds|
Milliseconds|Bytes|Kilobytes|Megabytes|Gigabytes|Terabytes|Bits|Kilobits|Megabits|
Gigabits|Terabits|Percent|Count|Bytes\\/Second|Kilobytes\\/Second|Megabytes\\/Second|
Gigabytes\\/Second|Terabytes\\/Second|Bits\\/Second|Kilobits\\/Second|Megabits\\/
Second|Gigabits\\/Second|Terabits\\/Second|Count\\/Second|None)$"
```

```
                                        },
                                        "StorageResolution": {
                                           "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/StorageResolution",
                                           "type": "integer",
                                           "title": "StorageResolution",
                                           "examples": [
                                               60
                                           ]
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Utilizzo dell' PutLogEventsAPI per inviare log in formato metrico incorporato creati manualmente

È possibile inviare log in formato metrico incorporato a Logs utilizzando l'API CloudWatch Logs. CloudWatch PutLogEvents Quando chiami PutLogEvents, puoi facoltativamente includere la seguente intestazione HTTP per indicare a CloudWatch Logs che le metriche devono essere estratte, ma non è più necessario.

```
x-amzn-logs-format: json/emf
```

Di seguito è riportato un esempio completo di utilizzo dell' AWS SDK for Java 2.x:

```
package org.example.basicapp;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatchlogs.CloudWatchLogsClient;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsRequest;
import software.amazon.awssdk.services.cloudwatchlogs.model.DescribeLogStreamsResponse;
import software.amazon.awssdk.services.cloudwatchlogs.model.InputLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.PutLogEventsRequest;
```

```
import java.util.Collections;

public class EmbeddedMetricsExample {
        public static void main(String[] args) {

                final String usage = "To run this example, supply a Region code (eg.
 us-east-1), log group, and stream name as command line arguments"
                                + "Ex: PutLogEvents <region-id> <log-group-name>
 <stream-name>";

                if (args.length != 3) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String regionId = args[0];
                String logGroupName = args[1];
                String logStreamName = args[2];

                CloudWatchLogsClient logsClient =
 CloudWatchLogsClient.builder().region(Region.of(regionId)).build();

                // Build a JSON log using the EmbeddedMetricFormat.
                long timestamp = System.currentTimeMillis();
                String message = "{" +
                                "  \"_aws\": {" +
                                "    \"Timestamp\": " + timestamp  + "," +
                                "    \"CloudWatchMetrics\": [" +
                                "      {" +
                                "        \"Namespace\": \"MyApp\"," +
                                "        \"Dimensions\": [[\"Operation\"], [\"Operation
\", \"Cell\"]]," +
                                "        \"Metrics\": [{ \"Name\": \"ProcessingLatency
\", \"Unit\": \"Milliseconds\", \"StorageResolution\": 60 }]" +
                                "      }" +
                                "    ]" +
                                "  }," +
                                "  \"Operation\": \"Aggregator\"," +
                                "  \"Cell\": \"001\"," +
                                "  \"ProcessingLatency\": 100" +
                                "}";
                InputLogEvent inputLogEvent = InputLogEvent.builder()
                        .message(message)
```

```
                        .timestamp(timestamp)
                        .build();

                // Specify the request parameters.
                PutLogEventsRequest putLogEventsRequest = PutLogEventsRequest.builder()
                        .logEvents(Collections.singletonList(inputLogEvent))
                        .logGroupName(logGroupName)
                        .logStreamName(logStreamName)
                        .build();

                logsClient.putLogEvents(putLogEventsRequest);

                System.out.println("Successfully put CloudWatch log event");
        }

}
```

> **Note**
>
> Con la specifica Embedded Metric Format, puoi tenere traccia dell'elaborazione dei registri EMF in base ai parametri pubblicati nello spazio dei nomi `AWS/Logs` del tuo account. Questi possono essere utilizzati per tenere traccia della generazione di parametri con esito negativo per EMF e per verificare se gli errori sono dovuti all'analisi o alla convalida. Per maggiori dettagli, consulta Monitoraggio con metriche CloudWatch .

Utilizzo dell' CloudWatch agente per inviare log in formato metrico incorporato

Per utilizzare questo metodo, installa prima l' CloudWatch agente per i servizi da cui desideri inviare i log in formato metrico incorporato, quindi puoi iniziare a inviare gli eventi.

L' CloudWatch agente deve avere la versione 1.230621.0 o successiva.

> **Note**
>
> Non è necessario installare l' CloudWatch agente per inviare i log dalle funzioni Lambda.

I timeout della funzione Lambda non vengono gestiti automaticamente. Ciò significa che se la funzione scade prima che i parametri vengano scaricati, i parametri per tale chiamata non verranno acquisiti.

Installazione dell'agente CloudWatch

Installa l' CloudWatch agente per ogni servizio che deve inviare log in formato metrico incorporato.

Installazione dell' CloudWatchagente su EC2

Innanzitutto, installa l' CloudWatch agente sull'istanza. Per ulteriori informazioni, consulta Installazione dell'agente CloudWatch .

Dopo aver installato l'agente, configura l'agente per l'ascolto su una porta UDP o TCP dei in log Embedded Metric Format. Di seguito è riportato un esempio di questa configurazione che rimane in ascolto sul socket predefinito `tcp:25888`. Per ulteriori informazioni sulla configurazione dell'agente, consulta Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Installazione dell' CloudWatchagente su Amazon ECS

Il modo più semplice per distribuire l' CloudWatch agente su Amazon ECS è eseguirlo come sidecar, definendolo nella stessa definizione di attività dell'applicazione.

Creazione del file di configurazione dell'agente

Crea il file di configurazione CloudWatch dell'agente localmente. In questo esempio, il percorso file relativo sarà `amazon-cloudwatch-agent.json`.

Per ulteriori informazioni sulla configurazione dell'agente, consulta Crea o modifica manualmente il file di configurazione dell' CloudWatch agente.

```
{
```

```
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Configurazione push in SSM Parameter Store

Immettere il seguente comando per inviare il file di configurazione dell' CloudWatch agente all'archivio dei parametri di AWS Systems Manager (SSM).

```
aws ssm put-parameter \
    --name "cwagentconfig" \
    --type "String" \
    --value "`cat amazon-cloudwatch-agent.json`" \
    --region "{{region}}"
```

Configurazione della definizione di attività

Configura la definizione dell'attività per utilizzare l' CloudWatch agente ed esporre la porta TCP o UDP. La definizione di attività di esempio da utilizzare dipende dalla modalità di rete.

Tieni presente che `webapp` specifica la variabile di ambiente `AWS_EMF_AGENT_ENDPOINT`. Questa viene utilizzata dalla libreria e deve mostrare l'endpoint su cui l'agente è in ascolto. Inoltre, `cwagent` specifica `CW_CONFIG_CONTENT` come parametro “valueFrom” che punta alla configurazione SSM creata nella fase precedente.

Questa sezione contiene un esempio per la modalità bridge e un esempio per la modalità host o awsvpc. Per altri esempi di come configurare l' CloudWatch agente su Amazon ECS, consulta il repository di esempi Github

Di seguito è riportato un esempio di modalità bridge. Quando è abilitata la modalità di rete bridge, l'agente deve essere collegato all'applicazione utilizzando il parametro `links` e deve essere indirizzato utilizzando il nome del container.

```
{
  "containerDefinitions": [
          {
              "name": "webapp",
              "links": [ "cwagent" ],
```

```
            "image": "my-org/web-app:latest",
            "memory": 256,
            "cpu": 256,
            "environment": [{
              "name": "AWS_EMF_AGENT_ENDPOINT",
              "value": "tcp://cwagent:25888"
            }],
        },
        {
            "name": "cwagent",
            "mountPoints": [],
            "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
            "memory": 256,
            "cpu": 256,
            "portMappings": [{
                "protocol": "tcp",
                "containerPort": 25888
            }],
            "environment": [{
              "name": "CW_CONFIG_CONTENT",
              "valueFrom": "cwagentconfig"
            }],
        }
    ],
}
```

Di seguito è riportato un esempio per la modalità host o la modalità awsvpc. Durante l'esecuzione su tali modalità di rete, l'agente può essere indirizzato su `localhost`.

```
{
  "containerDefinitions": [
        {
            "name": "webapp",
            "image": "my-org/web-app:latest",
            "memory": 256,
            "cpu": 256,
            "environment": [{
              "name": "AWS_EMF_AGENT_ENDPOINT",
              "value": "tcp://127.0.0.1:25888"
            }],
        },
        {
            "name": "cwagent",
```

```
            "mountPoints": [],
            "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
            "memory": 256,
            "cpu": 256,
            "portMappings": [{
                "protocol": "tcp",
                "containerPort": 25888
            }],
            "environment": [{
              "name": "CW_CONFIG_CONTENT",
              "valueFrom": "cwagentconfig"
            }],
        }
    ],
}
```

> **Note**
>
> In modalità awsvpc, è necessario fornire un indirizzo IP pubblico al VPC (solo Fargate), configurare un gateway NAT o configurare un endpoint VPC Logs. CloudWatch Per ulteriori informazioni sulla configurazione di un NAT, consulta Gateway NAT. Per ulteriori informazioni sulla configurazione di un endpoint VPC CloudWatch Logs, consulta CloudWatch Using Logs with Interface VPC Endpoints.
> Di seguito è riportato un esempio di come assegnare un indirizzo IP pubblico a un'attività che utilizza il tipo di lancio Fargate.
>
> ```
> aws ecs run-task \
> --cluster {{cluster-name}} \
> --task-definition cwagent-fargate \
> --region {{region}} \
> --launch-type FARGATE \
> --network-configuration
>  "awsvpcConfiguration={subnets=[{{subnetId}}],securityGroups=[{{sgId}}],assignPublicIp=ENA
> ```

Garantire le autorizzazioni

Assicurati che il ruolo IAM che esegue le attività disponga dell'autorizzazione per leggere dall'Archivio parametri SSM. Puoi aggiungere questa autorizzazione allegando la policy AmazonSSM. ReadOnlyAccess A questo scopo, immetti il comando seguente.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonSSMReadOnlyAccess
 \
--role-name CWAgentECSExecutionRole
```

Installazione dell' CloudWatchagente su Amazon EKS

Alcune parti di questo processo possono essere ignorate se hai già installato CloudWatch Container Insights su questo cluster.

Autorizzazioni

Se Container Insights non è già stato installato, assicurati innanzitutto che i nodi Amazon EKS dispongano delle autorizzazioni IAM appropriate. Dovrebbero avere il file CloudWatchAgentServerPolicyallegato. Per ulteriori informazioni, consulta Verifica dei prerequisiti di .

Crea ConfigMap

Crea un ConfigMap file per l'agente. Indica ConfigMap inoltre all'agente di ascoltare su una porta TCP o UDP. Usa quanto segue. ConfigMap

```
# cwagent-emf-configmap.yaml
apiVersion: v1
data:
  # Any changes here must not break the JSON format
  cwagentconfig.json: |
    {
      "agent": {
        "omit_hostname": true
      },
      "logs": {
        "metrics_collected": {
          "emf": { }
        }
      }
    }
kind: ConfigMap
metadata:
  name: cwagentemfconfig
  namespace: default
```

Se hai già installato Container Insights, aggiungi la `"emf": { }` riga seguente a quella esistente ConfigMap.

Applica il ConfigMap

Immettere il comando seguente per applicare il ConfigMap.

```
kubectl apply -f cwagent-emf-configmap.yaml
```

Distribuzione dell'agente

Per distribuire l' CloudWatch agente come sidecar, aggiungete l'agente alla definizione del contenitore, come illustrato nell'esempio seguente.

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
  namespace: default
spec:
  containers:
    # Your container definitions go here
    - name: web-app
      image: my-org/web-app:latest
    # CloudWatch Agent configuration
    - name: cloudwatch-agent
      image: public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest
      imagePullPolicy: Always
      resources:
        limits:
          cpu: 200m
          memory: 100Mi
        requests:
          cpu: 200m
          memory: 100Mi
      volumeMounts:
        - name: cwagentconfig
          mountPath: /etc/cwagentconfig
      ports:
  # this should match the port configured in the ConfigMap
        - protocol: TCP
          hostPort: 25888
          containerPort: 25888
  volumes:
    - name: cwagentconfig
      configMap:
```

```
        name: cwagentemfconfig
```

Utilizzo dell' CloudWatch agente per inviare log in formato metrico incorporato

Una volta installato e CloudWatch funzionante l'agente, è possibile inviare i log in formato metrico incorporato tramite TCP o UDP. Durante l'invio di log sull'agente è necessario rispettare due requisiti:

- I log devono contenere una chiave `LogGroupName` che indica all'agente quale gruppo di log utilizzare.
- Ogni evento di log deve trovarsi su una singola riga. In altre parole, un evento di log non può contenere il carattere di nuova riga (n).

Gli eventi di log devono inoltre seguire le specifiche Embedded Metric Format. Per ulteriori informazioni, consulta Specifica: Embedded Metric Format .

Se prevedi di creare allarmi su parametri creati utilizzando il formato dei parametri incorporati, consulta Impostazione degli allarmi sui parametri creati con il formato dei parametri incorporati per ottenere dei suggerimenti.

Di seguito è riportato un esempio di invio manuale di eventi di log da una shell bash Linux. Puoi invece utilizzare le interfacce socket UDP fornite dal linguaggio di programmazione preferito.

```
echo '{"_aws":{"Timestamp":1574109732004,"LogGroupName":"Foo","CloudWatchMetrics":
[{"Namespace":"MyApp","Dimensions":[["Operation"]],"Metrics":
[{"Name":"ProcessingLatency","Unit":"Milliseconds","StorageResolution":60}]}]},"Operation":"Agg
 \
> /dev/udp/0.0.0.0/25888
```

> **Note**
>
> Con la specifica Embedded Metric Format, puoi tenere traccia dell'elaborazione dei registri EMF in base ai parametri pubblicati nello spazio dei nomi `AWS/Logs` del tuo account. Questi possono essere utilizzati per tenere traccia della generazione di parametri con esito negativo per EMF e per verificare se gli errori sono dovuti all'analisi o alla convalida. Per maggiori dettagli, consulta Monitoraggio con metriche. CloudWatch

Utilizzo del formato metrico incorporato con AWS Distro per OpenTelemetry

È possibile utilizzare il formato metrico incorporato come parte del progetto. OpenTelemetry OpenTelemetry è un'iniziativa open source che rimuove i confini e le restrizioni tra i formati specifici del fornitore per la tracciabilità, i log e le metriche offrendo un unico set di specifiche e API. Per ulteriori informazioni, consulta. OpenTelemetry

L'utilizzo del formato metrico incorporato OpenTelemetry richiede due componenti: un'origine dati OpenTelemetry conforme e AWS Distro for OpenTelemetry Collector abilitato per l'uso con i log in formato metrico incorporato. CloudWatch

Abbiamo ridistribuzioni preconfigurate dei OpenTelemetry componenti, gestite da, per rendere l'onboarding il più semplice possibile. AWSPer ulteriori informazioni sull'utilizzo OpenTelemetry con il formato metrico incorporato, oltre ad altri AWS servizi, consulta Distro for.AWS OpenTelemetry

Per ulteriori informazioni sul supporto linguistico e sull'utilizzo, consulta Osservabilità di AWS su Github.

Visualizzazione dei parametri e dei log nella console

Dopo aver generato i log in formato metrico incorporato che estraggono le metriche, puoi utilizzare la CloudWatch console per visualizzare le metriche. Le dimensioni dei parametri incorporati sono quelle specificate al momento della generazione dei log. Inoltre, le dimensioni predefinite dei parametri incorporati generati utilizzando le librerie client sono le seguenti:

- ServiceType
- ServiceName
- LogGroup

Per visualizzare i parametri generati dai log in Embedded Metric Format

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nel riquadro di navigazione, seleziona Parametri.
3. Seleziona uno spazio dei nomi specificato per i parametri incorporati quando sono stati generati. Se hai utilizzato le librerie client per generare le metriche e non hai specificato uno spazio dei nomi, seleziona. aws-embedded-metrics Questo è lo spazio dei nomi predefinito per i parametri incorporati generati utilizzando le librerie client.

4. Seleziona una dimensione metrica (ad esempio,). ServiceName
5. La scheda All metrics (Tutti i parametri) visualizza tutti i parametri per tale dimensione nello spazio dei nomi. Puoi eseguire le operazioni indicate di seguito:

    a. Per ordinare la tabella, utilizza l'intestazione della colonna.

    b. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro. Per selezionare tutte i parametri, seleziona la casella di controllo nella riga dell'intestazione della tabella.

    c. Per filtrare per risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).

    d. Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).

Interrogazione dei log utilizzando Logs Insights CloudWatch

È possibile interrogare gli eventi di registro dettagliati associati alle metriche estratte utilizzando CloudWatch Logs Insights per fornire informazioni approfondite sulle cause principali degli eventi operativi. Uno dei vantaggi dell'estrazione di parametri dai log è la possibilità di filtrare i log in un secondo momento in base al parametro univoco (nome parametro più set di dimensioni univoco) e ai valori dei parametri, per ottenere il contesto sugli eventi che hanno contribuito al valore del parametro aggregato

Ad esempio, per ottenere un ID di richiesta o un ID di traccia a raggi x interessato, è possibile eseguire la seguente query in Logs Insights. CloudWatch

```
filter Latency > 1000 and Operation = "Aggregator"
| fields RequestId, TraceId
```

Puoi inoltre eseguire l'aggregazione al momento dell'esecuzione delle query su chiavi ad alta cardinalità, ad esempio individuando i clienti influenzati da un evento. Nell'esempio seguente viene descritto quanto segue.

```
filter Latency > 1000 and Operation = "Aggregator"
| stats count() by CustomerId
```

Per ulteriori informazioni, consulta Analisi dei dati di registro con Logs Insights CloudWatch

Impostazione degli allarmi sui parametri creati con il formato dei parametri incorporati

In generale, la creazione di allarmi su parametri generati dal formato dei parametri incorporati segue lo stesso schema della creazione di allarmi su qualsiasi altro parametro. Per ulteriori informazioni, consulta Utilizzo degli CloudWatch allarmi Amazon.

La generazione di metriche in formato metrico incorporato dipende dal flusso di pubblicazione dei log, poiché i log devono essere elaborati da CloudWatch Logs per essere trasformati in metriche. È quindi importante pubblicare i log in modo tempestivo, in modo che i punti di dati dei parametri vengano creati entro il periodo di tempo in cui vengono valutati gli allarmi.

Se prevedi di utilizzare il formato metrico incorporato per inviare metriche ad alta risoluzione e creare allarmi in base a tali metriche, ti consigliamo di scaricare i log nei CloudWatch registri a intervalli di 5 secondi o meno per evitare di introdurre ulteriori ritardi che possono causare allarmi su dati parziali o mancanti. Se si utilizza l' CloudWatch agente, è possibile regolare l'intervallo di flush impostando il parametro nel file di configurazione dell'agente. `force_flush_interval` CloudWatch Il valore predefinito di questo valore è 5 secondi.

Se utilizzi Lambda su altre piattaforme in cui non puoi controllare l'intervallo di caricamento dei log, prendi in considerazione l'utilizzo degli allarmi "M di N" per controllare il numero di punti di dati utilizzati per creare un allarme. Per ulteriori informazioni, consulta Valutazione di un allarme.

AWS servizi che pubblicano CloudWatch metriche

I seguenti AWS servizi pubblicano le metriche su. CloudWatch Per informazioni sui parametri e le dimensioni, consulta la documentazione indicata.

Servizio	Spazio dei nomi	Documentazione
AWS Amplify	`AWS/AmplifyHosting`	Monitoraggio
Amazon API Gateway	`AWS/ApiGateway`	Monitora l'esecuzione delle API con Amazon CloudWatch
Amazon AppFlow	`AWS/AppFlow`	Monitoraggio di Amazon AppFlow con Amazon CloudWatch
AWS Servizio di migrazione delle applicazioni	`AWS/MGN`	Servizio di monitoraggio della migrazione delle applicazioni con Amazon CloudWatch
AWS App Runner	`AWS/AppRunner`	Visualizzazione delle metriche del servizio App Runner riportate a CloudWatch
AppStream 2.0	`AWS/AppStream`	Monitoraggio delle risorse Amazon AppStream 2.0
AWS AppSync	`AWS/AppSync`	CloudWatch Metriche
Amazon Athena	`AWS/Athena`	Monitoraggio delle query Athena con metriche CloudWatch
Amazon Aurora	`AWS/RDS`	Parametri di Amazon Aurora
AWS Backup	`AWS/Backup`	Monitoraggio delle metriche di AWS Backup con CloudWatch
Amazon Bedrock	`AWS/Bedrock`	Monitoraggio di Amazon Bedrock con Amazon CloudWatch

Servizio	Spazio dei nomi	Documentazione
AWS Billing and Cost Management	`AWS/Billing`	Monitoraggio dei costi con avvisi e notifiche
Amazon Braket	`AWS/Braket/By Device`	Monitoraggio di Amazon Braket con Amazon CloudWatch
AWS Certificate Manager	`AWS/CertificateManager`	Metriche supportate CloudWatch
CA privata AWS	`AWS/ACMPrivateCA`	Metriche supportate CloudWatch
AWS Chatbot	`AWS/Chatbot`	Monitoraggio AWS Chatbot con Amazon CloudWatch
Amazon Chime	`AWS/ChimeVoiceConnector`	Monitoraggio Amazon Chime con Amazon CloudWatch
SDK Amazon Chime	`AWS/ChimeSDK`	Parametri del servizio
AWS Client VPN	`AWS/ClientVPN`	Monitoraggio con Amazon CloudWatch
Amazon CloudFront	`AWS/CloudFront`	Monitoraggio CloudFront dell'attività utilizzando CloudWatch
AWS CloudHSM	`AWS/CloudHSM`	Ottenere CloudWatch metriche
Amazon CloudSearch	`AWS/CloudSearch`	Monitoraggio di un CloudSearch dominio Amazon con Amazon CloudWatch
AWS CloudTrail	`AWS/CloudTrail`	CloudWatch Metriche supportate

Servizio	Spazio dei nomi	Documentazione
CloudWatch agente	`CWAgent` o uno spazio dei nomi personalizzato	Metriche raccolte dall'agente CloudWatch
CloudWatch flussi metrici	`AWS/CloudWatch/MetricStreams`	Monitoraggio dei flussi metrici con le metriche CloudWatch
CloudWatch RUM	`AWS/RUM`	CloudWatch metriche che puoi raccogliere con RUM CloudWatch
CloudWatch Synthetics	`CloudWatchSynthetics`	CloudWatch metriche pubblicate da canaries
CloudWatch Registri Amazon	`AWS/Logs`	Monitoraggio dell'utilizzo con metriche CloudWatch
AWS CodeBuild	`AWS/CodeBuild`	Monitoraggio AWS CodeBuild
CodeGuru Revisore Amazon		Monitoraggio dei CodeGuru revisori con Amazon CloudWatch
Amazon Kendra		Monitoraggio di Amazon Kendra con Amazon CloudWatch
Amazon CodeWhisperer	`AWS/CodeWhisperer`	Monitoraggio Amazon CodeWhisperer con Amazon CloudWatch
Amazon Cognito	`AWS/Cognito`	Monitoraggio Amazon Cognito
Amazon Comprehend	`AWS/Comprehend`	Monitoraggio degli Amazon Comprehend endpoint
AWS Config	`AWS/Config`	AWS Config Metriche di utilizzo e successo

Servizio	Spazio dei nomi	Documentazione
Amazon Connect	`AWS/Connect`	Monitoraggio di Amazon Connect in Amazon CloudWatch Metrics
Amazon Data Lifecycle Manager	`AWS/DataLifecycleManager`	Monitora le tue politiche con Amazon CloudWatch
AWS DataSync	`AWS/DataSync`	Monitoraggio dell'attività
Amazon DataZone		Monitoraggio di Amazon DataZone con Amazon CloudWatch
Amazon DevOps Guru	`AWS/DevOps-Guru`	Monitoraggio Amazon DevOps Guru con Amazon CloudWatch
AWS Database Migration Service	`AWS/DMS`	AWS DMS Attività di monitoraggio
AWS Direct Connect	`AWS/DX`	Monitoraggio con Amazon CloudWatch
AWS Directory Service	`AWS/DirectoryService`	Usa i CloudWatch parametri di Amazon per determinare quando aggiungere controller di dominio
Amazon DocumentDB	`AWS/DocDB`	Parametri di Amazon DocumentDB
Amazon DynamoDB	`AWS/DynamoDB`	Parametri e dimensioni di DynamoDB
DynamoDB Accelerator (DAX)	`AWS/DAX`	Visualizzazione di parametri e dimensioni DAX

Servizio	Spazio dei nomi	Documentazione
Amazon EC2	`AWS/EC2`	Monitoraggio delle istanze tramite CloudWatch
Amazon EC2 Elastic Graphics	`AWS/ElasticGPUs`	Utilizzo delle CloudWatch metriche per monitorare Elastic Graphics
Serie di istanze Spot Amazon EC2	`AWS/EC2Spot`	CloudWatch Metriche per Spot Fleet
Dimensionamento automatico Amazon EC2	`AWS/AutoScaling`	Monitoraggio dei gruppi e delle istanze di Auto Scaling utilizzando CloudWatch
AWS Elastic Beanstalk	`AWS/ElasticBeanstalk`	Pubblicazione di parametri Amazon CloudWatch Custom per un ambiente
Amazon Elastic Block Store	`AWS/EBS`	Amazon CloudWatch Metrics per Amazon EBS
Amazon Elastic Container Registry	`AWS/ECR`	Parametri del repository Amazon ECR
Amazon Elastic Container Service	`AWS/ECS`	Metriche di Amazon ECS CloudWatch
Amazon ECS tramite CloudWatch Container Insights	`ECS/ContainerInsights`	Parametri di Container Insights per Amazon ECS

Servizio	Spazio dei nomi	Documentazione
Auto Scaling del cluster Amazon ECS	`AWS/ECS/ManagedScaling`	Auto Scaling del cluster Amazon ECS
AWS Elastic Disaster Recovery		CloudWatch Metriche per DRS
Amazon Elastic File System	`AWS/EFS`	Monitoraggio con CloudWatch
Amazon Elastic Inference	`AWS/ElasticInference`	Utilizzo di CloudWatch metriche per monitorare Amazon Elastic Inference
Amazon EKS tramite CloudWatch Container Insights	`ContainerInsights`	Parametri di Container Insights per Amazon EKS e Kubernetes
Sistema di bilanciamento del carico elastico	`AWS/ApplicationELB`	CloudWatch Metriche per il tuo Application Load Balancer
Sistema di bilanciamento del carico elastico	`AWS/NetworkELB`	CloudWatch Metriche per il tuo Network Load Balancer
Sistema di bilanciamento del carico elastico	`AWS/GatewayELB`	CloudWatch Metriche per il tuo Gateway Load Balancer

Servizio	Spazio dei nomi	Documentazione
Sistema di bilanciamento del carico elastico	`AWS/ELB`	CloudWatch Metriche per il tuo Classic Load Balancer
Amazon Elastic Transcoder	`AWS/ElasticTranscoder`	Monitoraggio con Amazon CloudWatch
Amazon ElastiCache per Memcached	`AWS/ElastiCache`	Monitoraggio dell'utilizzo con metriche CloudWatch
Amazon ElastiCache per Redis	`AWS/ElastiCache`	Monitoraggio dell'utilizzo con metriche CloudWatch
OpenSearch Servizio Amazon	`AWS/ES`	Monitoraggio delle metriche dei OpenSearch cluster con Amazon CloudWatch
Amazon EMR	`AWS/ElasticMapReduce`	Monitora le metriche con CloudWatch
AWS Elemental MediaConnect	`AWS/MediaConnect`	Monitoraggio MediaConnect con Amazon CloudWatch
AWS Elemental MediaConvert	`AWS/MediaConvert`	Utilizzo CloudWatch delle metriche per visualizzare le metriche relative alle risorse AWS Elemental MediaConvert
AWS Elemental MediaLive	`AWS/MediaLive`	Monitoraggio dell'attività utilizzando i CloudWatch parametri di Amazon
AWS Elemental MediaPackage	`AWS/MediaPackage`	Monitoraggio AWS Elemental MediaPackage con Amazon CloudWatch Metrics

Servizio	Spazio dei nomi	Documentazione
AWS Elemental MediaStore	`AWS/MediaStore`	Monitoraggio AWS Elemental MediaStore con Amazon CloudWatch Metrics
AWS Elemental MediaTailor	`AWS/MediaTailor`	Monitoraggio AWS Elemental MediaTailor con Amazon CloudWatch
Amazon EventBridge	`AWS/Events`	Monitoraggio di Amazon EventBridge
Amazon FinSpace		Registrazione e monitoraggio
Amazon Forecast		CloudWatch Metriche per Amazon Forecast
Amazon Fraud Detector		Monitoraggio di Amazon Fraud Detector con Amazon CloudWatch
Amazon FSx per Lustre	`AWS/FSx`	Monitoraggio di Amazon FSx for Lustre
Amazon FSx per OpenZFS	`AWS/FSx`	Monitoraggio con Amazon CloudWatch
Amazon FSx per Windows File Server	`AWS/FSx`	Monitoraggio di Amazon FSx per Windows File Server
Amazon FSx per ONTAP NetApp	`AWS/FSx`	Monitoraggio con Amazon CloudWatch
Amazon FSx per OpenZFS	`AWS/FSx`	Monitoraggio con Amazon CloudWatch
Amazon GameLift	`AWS/GameLift`	Monitora Amazon GameLift con CloudWatch

Servizio	Spazio dei nomi	Documentazione
AWS Global Accelerator	`AWS/GlobalAccelerator`	Usare Amazon CloudWatch con AWS Global Accelerator
AWS Glue	`Glue`	Monitoraggio AWS Glue tramite CloudWatch metriche
AWS Ground Station	`AWS/GroundStation`	Metriche con Amazon CloudWatch
AWS HealthLake	`AWS/HealthLake`	Monitoraggio HealthLake con CloudWatch
Amazon Inspector	`AWS/Inspector`	Monitoraggio tramite Amazon Inspector CloudWatch
Amazon Interactive Video Service	`AWS/IVS`	Monitoraggio di Amazon IVS con Amazon CloudWatch
Amazon Interactive Video Service Chat	`AWS/IVSChat`	Monitoraggio di Amazon IVS con Amazon CloudWatch
AWS IoT	`AWS/IoT`	AWS IoT Parametri e dimensioni
AWS IoT Analytics	`AWS/IoTAnalytics`	Spazio dei nomi, parametri e dimensioni
AWS IoT FleetWise	`AWS/IoTFleetWise`	Monitoraggio AWS dell'IoT FleetWise con Amazon CloudWatch
AWS IoT SiteWise	`AWS/IoTSiteWise`	Monitoraggio AWS IoT SiteWise con i CloudWatch parametri di Amazon
AWS IoT TwinMaker	`AWS/IoTTwinMaker`	Monitoraggio AWS IoT TwinMaker con i CloudWatch parametri di Amazon
AWS IoT 1 clic		Monitoraggio AWS IoT 1-Click con Amazon CloudWatch

Servizio	Spazio dei nomi	Documentazione
AWS Key Management Service	`AWS/KMS`	Monitoraggio con CloudWatch
Amazon Keyspaces (per Apache Cassandra)	`AWS/Cassandra`	Parametri e dimensioni di Amazon Keyspaces
Amazon Kendra		Monitoraggio di Amazon Kendra con Amazon CloudWatch
Servizio gestito da Amazon per Apache Flink	`AWS/KinesisAnalytics`	Servizio gestito per applicazioni Apache Flink per SQL: monitoraggio con CloudWatch Servizio gestito per Apache Flink: visualizzazione dei parametri e delle dimensioni del servizio gestito da Amazon per Apache Flink
Amazon Data Firehose	`AWS/Firehose`	Monitoraggio di Firehose mediante metriche CloudWatch
Flusso di dati Amazon Kinesis	`AWS/Kinesis`	Monitoraggio di Amazon Kinesis Data Streams con Amazon CloudWatch
Flusso di video Amazon Kinesis	`AWS/KinesisVideo`	Monitoraggio delle metriche di Kinesis Video Streams con CloudWatch
AWS Lambda	`AWS/Lambda`	AWS Lambda Metriche
Amazon Lex	`AWS/Lex`	Monitoraggio di Amazon Lex con Amazon CloudWatch

Servizio	Spazio dei nomi	Documentazione

AWS License Manager	`AWSLicenseManager/licenseUsage`
	
	`AWS/LicenseManager/LinuxSubscriptions`
		Monitoraggio dell'utilizzo delle licenze con Amazon CloudWatch
		
		Metriche di utilizzo e CloudWatch allarmi Amazon per gli abbonamenti Linux

Servizio di posizione Amazon	`AWS/Location`	Metriche di Amazon Location Service esportate su Amazon CloudWatch
Amazon Lookout per le apparecchiature	`AWS/lookoutequipment`	Monitoraggio di Lookout for Equipment con Amazon CloudWatch
Amazon Lookout per le metriche	`AWS/LookoutMetrics`	Monitoraggio di Lookout for Metrics con Amazon CloudWatch
Amazon Lookout per Vision	`AWS/LookoutVision`	Monitoraggio di Lookout for Vision con Amazon CloudWatch
AWS Modernizzazione del mainframe		Monitoraggio della modernizzazione AWS del mainframe con Amazon CloudWatch
Amazon Machine Learning	`AWS/ML`	Monitoraggio di Amazon ML con CloudWatch metriche
Blockchain gestita da Amazon	`AWS/managedblockchain`	Utilizzo dei parametri dei nodi peer di Hyperledger Fabric su Blockchain gestita da Amazon

Servizio	Spazio dei nomi	Documentazione
Amazon Managed Service per Prometheus	`AWS/Prometheus`	CloudWatch Metriche Amazon
Amazon Managed Streaming per Apache Kafka	`AWS/Kafka`	Monitoraggio di Amazon MSK con Amazon CloudWatch
Amazon Managed Streaming per Apache Kafka	`AWS/KafkaConnect`	Monitoraggio di MSK Connect
Amazon Managed Workflows for Apache Airflow	`AWS/MWAA`	Parametri di container, code e database per Amazon MWAA
Amazon MemoryDB for Redis	`AWS/MemoryDB`	Parametri di monitoraggio CloudWatch
Amazon MQ	`AWS/AmazonMQ`	Monitoraggio dei broker Amazon MQ tramite Amazon CloudWatch
Amazon Neptune	`AWS/Neptune`	Monitoraggio di Neptune con CloudWatch
AWS Network Firewall	`AWS/NetworkFirewall`	AWS Network Firewall metriche in Amazon CloudWatch
AWS Gestore di rete	`AWS/NetworkManager`	CloudWatch metriche per le risorse locali

Servizio	Spazio dei nomi	Documentazione
Amazon Nimble Studio	`AWS/NimbleStudio`	Monitoraggio di Nimble Studio con Amazon CloudWatch
AWS HealthOmics	`AWS/Omics`	Monitoraggio AWS HealthOmics con Amazon CloudWatch
AWS OpsWorks	`AWS/OpsWorks`	Monitoraggio degli stack tramite Amazon CloudWatch
AWS Outposts	`AWS/Outposts`	CloudWatch metriche per AWS Outposts
AWS Panorama	`AWS/PanoramaDeviceMetrics`	Monitoraggio di dispositivi e applicazioni con Amazon CloudWatch
Amazon Personalize	`AWS/Personalize`	CloudWatch metriche per Amazon Personalize
Amazon Pinpoint	`AWS/Pinpoint`	Visualizza le metriche Amazon Pinpoint in CloudWatch
Amazon Polly	`AWS/Polly`	Integrazione CloudWatch con Amazon Polly
AWS PrivateLink	`AWS/PrivateLinkEndpoints`	CloudWatch metriche per AWS PrivateLink
AWS PrivateLink	`AWS/PrivateLinkServices`	CloudWatch metriche per AWS PrivateLink
AWS 5G privato	`AWS/Private5G`	CloudWatch Metriche Amazon
Amazon QLDB	`AWS/QLDB`	Monitoraggio dei dati in Amazon QuickSight

Servizio	Spazio dei nomi	Documentazione
Amazon QuickSight	`AWS/QuickSight`	Monitoraggio con Amazon CloudWatch
Amazon Redshift	`AWS/Redshift`	Dati di prestazioni di Amazon Redshift
Amazon Relational Database Service	`AWS/RDS`	Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch
Amazon Rekognition	`AWS/Rekognition`	Monitoraggio di Rekognition con Amazon CloudWatch
AWS re:Post Privato	`AWS/rePostPrivate`	Monitoraggio AWS re:Post privato con Amazon CloudWatch
AWS RoboMaker	`AWS/Robomaker`	Monitoraggio AWS RoboMaker con Amazon CloudWatch
Amazon Route 53	`AWS/Route53`	Monitoraggio di Amazon Route 53
Route 53 Application Recovery Controller	`AWS/Route53RecoveryReadiness`	Utilizzo di Amazon CloudWatch con Application Recovery Controller
Amazon SageMaker	`AWS/SageMaker`	Monitoraggio SageMaker con CloudWatch
Amazon SageMaker Model Building Pipeline	`AWS/SageMaker/ModelBuildingPipeline`	SageMaker Metriche delle pipeline

Servizio	Spazio dei nomi	Documentazione
AWS Secrets Manager	`AWS/SecretsManager`	Monitoraggio di Secrets Manager con Amazon CloudWatch
Amazon Security Lake	`AWS/SecurityLake`	CloudWatch metriche per Amazon Security Lake
Catalogo dei servizi	`AWS/ServiceCatalog`	CloudWatch Metriche del Service Catalog
AWS Shield Advanced	`AWS/DDoSProtection`	Monitoraggio con CloudWatch
Amazon Simple Email Service	`AWS/SES`	Recupero dei dati degli eventi di Amazon SES da CloudWatch
AWS SimSpace Weaver	`AWS/simspaceweaver`	Monitoraggio AWS SimSpace Weaver con Amazon CloudWatch
Amazon Simple Notification Service	`AWS/SNS`	Monitoraggio di Amazon SNS con CloudWatch
Amazon Simple Queue Service	`AWS/SQS`	Monitoraggio delle code Amazon SQS tramite CloudWatch
Amazon S3	`AWS/S3`	Monitoraggio delle metriche con Amazon CloudWatch
S3 Storage Lens	`AWS/S3/Storage-Lens`	Monitora i parametri di S3 Storage Lens in CloudWatch
Amazon Simple Workflow Service	`AWS/SWF`	Metriche di Amazon SWF per CloudWatch
AWS Step Functions	`AWS/States`	Monitoraggio di Step Functions tramite CloudWatch

Servizio	Spazio dei nomi	Documentazione
AWS Storage Gateway	`AWS/StorageGateway`	Utilizzo dei CloudWatch parametri di Amazon
AWS Systems Manager Esegui comando	`AWS/SSM-RunCommand`	Monitoraggio Run Command Metrics utilizzando CloudWatch
Amazon Textract	`AWS/Textract`	CloudWatch Metriche per Amazon Textract
Amazon Timestream	`AWS/Timestream`	Parametri e dimensioni di Timestream
AWS Transfer for SFTP	`AWS/Transfer`	AWS SFTP CloudWatch Metriche
Amazon Transcribe	`AWS/Transcribe`	Monitoraggio Amazon Transcribe con Amazon CloudWatch
Amazon Translate	`AWS/Translate`	CloudWatch Metriche e dimensioni per Amazon Translate
AWS Trusted Advisor	`AWS/TrustedAdvisor`	Creazione di allarmi Trusted Advisor utilizzando CloudWatch
Amazon VPC	`AWS/NATGateway`	Monitoraggio del gateway NAT con CloudWatch
Amazon VPC	`AWS/TransitGateway`	CloudWatch Metriche per i tuoi gateway di transito
Amazon VPC	`AWS/VPN`	Monitoraggio con CloudWatch
Amazon VPC IP Address Manager	`AWS/IPAM`	Crea allarmi con Amazon CloudWatch

Servizio	Spazio dei nomi	Documentazione
AWS WAF	`AWS/WAFV2` per risorse AWS WAF `WAF`per risorse AWS WAF classiche	Monitoraggio con CloudWatch
Amazon WorkMail	`AWS/WorkMail`	Monitoraggio Amazon WorkMail con Amazon CloudWatch
Amazon WorkSpaces	`AWS/WorkSpaces`	Monitora le tue CloudWatch metriche di WorkSpaces utilizzo
Amazon WorkSpaces Web	`AWS/WorkSpacesWeb`	Monitoraggio di Amazon WorkSpaces Web con Amazon CloudWatch

AWS metriche di utilizzo

CloudWatch raccoglie metriche che tracciano l'utilizzo di alcune AWS risorse e API. Tali parametri sono pubblicati nello spazio dei nomi `AWS/Usage`. Le metriche di utilizzo CloudWatch consentono di gestire in modo proattivo l'utilizzo visualizzando le metriche nella CloudWatch console, creando dashboard personalizzate, rilevando le variazioni di attività con il rilevamento delle CloudWatch anomalie e configurando allarmi che avvisano l'utente quando l'utilizzo si avvicina a una soglia.

Alcuni AWS servizi integrano queste metriche di utilizzo con Service Quotas. Per questi servizi, puoi utilizzarli CloudWatch per gestire l'utilizzo delle quote di servizio da parte del tuo account. Per ulteriori informazioni, consulta Visualizzazione delle quote di servizio e impostazione degli allarmi.

Argomenti

Visualizzazione delle quote di servizio e impostazione degli allarmi

Per alcuni AWS servizi, puoi utilizzare le metriche di utilizzo per visualizzare l'utilizzo corrente del servizio su grafici e dashboard. CloudWatch È possibile utilizzare una funzione matematica CloudWatch metrica per visualizzare le quote di servizio per tali risorse sui grafici. È possibile, inoltre, configurare gli allarmi che avvisano quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni sulle quote di servizio, consulta la sezione che descrive cosa solo le quote di servizio nella Guida per l'utente di Service Quotas.

Se hai effettuato l'accesso a un account configurato come account di monitoraggio in modalità osservabile CloudWatch tra più account, puoi utilizzare quell'account di monitoraggio per visualizzare le quote di servizio e impostare allarmi per le metriche negli account di origine collegati a quell'account di monitoraggio. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

Attualmente, i seguenti servizi integrano i parametri di utilizzo con Service Quotas:

- AWS CloudHSM
- SDK Amazon Chime
- Amazon CloudWatch

- CloudWatch Registri Amazon
- Amazon DynamoDB
- Amazon EC2
- Amazon Elastic Container Registry
- Sistema di bilanciamento del carico elastico
- AWS Fargate
- AWS Fault Injection Service
- AWS Servizio video interattivo
- AWS Key Management Service
- Amazon Data Firehose
- Servizio di posizione Amazon
- Interrogazione su Amazon Managed Blockchain (AMB)
- AWS RoboMaker
- Amazon SageMaker

Per visualizzare una quota di servizio e impostare facoltativamente un allarme

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nella scheda Tutte le metriche, scegli Utilizzo, quindi scegli Per AWS risorsa.

   Viene visualizzato l'elenco dei parametri di utilizzo delle quote di servizio.
4. Seleziona la casella di controllo accanto a uno dei parametri.

   Il grafico mostra l'utilizzo corrente di quella AWS risorsa.
5. Per aggiungere la quota di servizio al grafico, procedere come indicato di seguito:

   a. Seleziona la scheda Graphed metrics (Parametri nel grafico).

   b. Scegli Math expression (Espressione matematica), Start with an empty expression (Inizia con un'espressione vuota). Quindi nella nuova riga, in Details (Dettagli), inserisci **`SERVICE_QUOTA(m1)`**.

      Una nuova riga viene aggiunta al grafico, visualizzando la quota di servizio per la risorsa rappresentata nel parametro.

6. Per visualizzare l'utilizzo corrente come una percentuale della quota, aggiungere una nuova espressione o modificare l'espressione SERVICE_QUOTA corrente. La nuova espressione da usare è **`m1/SERVICE_QUOTA(m1)*100`**.
7. (Facoltativo) Per impostare un allarme che avvisa se ci si avvicina alla quota di servizio, procedere nel modo seguente:

   a. Nella riga con **`m1/SERVICE_QUOTA(m1)*100`**, in Actions (Operazioni), scegli l'icona di allarme. L'aspetto è simile quello di una campana.

      Viene visualizzata la pagina di creazione dell'allarme.

   b. In Conditions (Condizioni), assicurarti che Threshold type (Tipo di soglia) sia Static (Statico) e Whenever Expression1 is (Ogni volta che Expression1 è) sia impostato su Greater (Maggiore). In than (di), immetti **`80`**. Viene creato un allarme che passo nello stato ALARM quando l'utilizzo supera l'80% della quota.

   c. Seleziona Next (Successivo).

   d. Nella pagina successiva, seleziona un argomento Amazon SNS o creane uno nuovo, quindi seleziona Next (Avanti). L'argomento selezionato riceve una notifica quando l'allarme passa nello stato ALLARME.

   e. Nella pagina successiva, immetti un nome e una descrizione per l'allarme, quindi scegli Next (Avanti).

   f. Scegli Crea allarme.

AWS Metriche di utilizzo delle API

La maggior parte delle API che supportano la AWS CloudTrail registrazione riporta anche le metriche di utilizzo a. CloudWatch Le metriche di utilizzo delle API CloudWatch consentono di gestire in modo proattivo l'utilizzo delle API visualizzando le metriche nella CloudWatch console, creando dashboard personalizzate, rilevando le variazioni di attività con CloudWatch Anomaly Detection e configurando allarmi che avvisano quando l'utilizzo si avvicina a una soglia.

La tabella seguente elenca i servizi a cui segnalano le metriche di utilizzo delle API e il valore da utilizzare per la CloudWatch dimensione per visualizzare le metriche di utilizzo di quel servizio. `Service`

Servizio	Valore della dimensione **`Service`**
AWS Identity and Access Management Access Analyzer	`Access Analyzer`
AWS Account Management	`Account Management`
Alexa for Business	`A4B`
Amazon API Gateway	`API Gateway`
AWS App Mesh	`App Mesh`
AWS AppConfig	`AWS AppConfig`
Amazon AppFlow	`AppFlow`
Application Auto Scaling	`Application Auto Scaling`
Application Discovery Service	`Application Discovery Service`
Amazon AppStream	`AppStream`
AppStream 2.0 Image Builder	`Image Builder`
Amazon Athena	`Athena`
AWS Audit Manager	`Audit Manager`
AWS Backup	`Backup`
AWS Batch	`Batch`
Amazon Braket	`Braket`
AWS Bilanci	`Budgets`
AWS Certificate Manager	`Certificate Manager`
SDK Amazon Chime	`ChimeSDK`

Servizio	Valore della dimensione **`Service`**
Directory del cloud Amazon	`Cloud Directory`
AWS Cloud Map	`Cloud Map`
AWS CloudFormation	`CloudFormation`
AWS CloudHSM	`CloudHSM`
Amazon CloudSearch	`CloudSearch`
AWS CloudShell	`CloudShell`
AWS CloudTrail	`CloudTrail`
Amazon CloudWatch	`CloudWatch`
CloudWatch Registri Amazon	`Logs`
Informazioni approfondite sulle CloudWatch applicazioni Amazon	`CloudWatch Application Insights`
AWS CodeBuild	`CodeBuild`
AWS CodeCommit	`CodeCommit`
Amazon CodeGuru Profiler	`CodeGuru Profiler`
AWS CodePipeline	`CodePipeline`
AWS CodeStar	`CodeStar`
AWS CodeStar Notifiche	`CodeStar Notifications`
AWS CodeStar Connessioni	`CodeStar Connections`
Pool di identità di Amazon Cognito	`Cognito Identity Pools`
Amazon Cognito Sync	`Cognito Sync`
Amazon Comprehend	`Comprehend`

Servizio	Valore della dimensione **`Service`**
Amazon Comprehend Medical	`Comprehend Medical`
AWS Compute Optimizer	`ComputeOptimzier`
Amazon Connect	`Connect`
Customer Profiles Amazon Connect	`Customer Profiles`
AWS Rapporti sui costi e sull'utilizzo	`Cost and Usage Report`
AWS Cost Explorer	`Cost Explorer`
AWS Data Exchange	`Data Exchange`
AWS Gestore del ciclo di vita dei dati	`Data Lifecycle Manager`
AWS Database Migration Service	`Database Migration Service`
AWS DataSync	`DataSync`
AWS DeepLens	`AWS DeepLens`
Amazon Detective	`Detective`
Device Advisor	`Device Advisor`
AWS Direct Connect	`Direct Connect`
AWS Directory Service	`Directory Service`
DynamoDB Accelerator	`DynamoDBAccelerator`
Amazon EC2	`EC2`
Dimensionamento automatico di EC2	`EC2 Auto Scaling`
Amazon Elastic Container Registry	`ECR Public`
Amazon Elastic Container Service	`ECS`

Servizio	Valore della dimensione **`Service`**
Amazon Elastic File System	`EFS`
Amazon Elastic Kubernetes Service	`EKS`
AWS Elastic Beanstalk	`Elastic Beanstalk`
Amazon Elastic Inference	`Elastic Inference`
Sistema di bilanciamento del carico elastico	`Elastic Load Balancing`
Amazon EMR	`EMR Containers`
AWS Firewall Manager	`Firewall Manager`
Amazon FSx	`FSx`
Amazon GameLift	`GameLift`
AWS Glue DataBrew	`DataBrew`
Grafana gestito da Amazon	`Grafana`
AWS IoT Greengrass	`Greengrass`
AWS Ground Station	`Ground Station`
AWS Health API e notifiche	`AWS Health APIs And Notifications`
Amazon Interactive Video Service	`IVS`
AWS IoT Core	`IoT`
AWS IoT 1 clic	`IoT 1-Click`
AWS IoT Events	`IoT Events`
AWS IoT RoboRunner	`IoT RoboRunner`
AWS IoT SiteWise	`IoT Sitewise`

Servizio	Valore della dimensione **`Service`**
AWS IoT Wireless	`IoT Wireless`
Amazon Kendra	`Kendra`
Amazon Keyspaces (per Apache Cassandra)	`Keyspaces`
Servizio gestito da Amazon per Apache Flink	`Kinesis Analytics`
Amazon Data Firehose	`Firehose`
Kinesis Video Streams	`Kinesis Video Streams`
AWS Key Management Service	`KMS`
AWS Lambda	`Lambda`
AWS Launch Wizard	`Launch Wizard`
Amazon Lex	`Amazon Lex`
Amazon Lightsail	`Lightsail`
Servizio di posizione Amazon	`Location`
Amazon Lookout per Vision	`Lookout for Vision`
Amazon Machine Learning	`Amazon Machine Learning`
Amazon Macie	`Macie`
Interrogazione su Amazon Managed Blockchain (AMB)	`Amazon Managed Blockchain Query`
AWS Managed Services	`AWS Managed Services`
AWS Marketplace Commerce Analytics	`Marketplace Analytics Service`
AWS Elemental MediaConnect	`MediaConnect`
AWS Elemental MediaConvert	`MediaConvert`

Servizio	Valore della dimensione **`Service`**
AWS Elemental MediaLive	`MediaLive`
AWS Elemental MediaStore	`Mediastore`
AWS Elemental MediaTailor	`MediaTailor`
AWS Mobile Hub	`Mobile Hub`
AWS Network Firewall	`Network Firewall`
AWS OpsWorks	`OpsWorks`
AWS OpsWorks per la gestione della configurazione	`OPsWorks CM`
AWS Outposts	`Outposts`
AWS Organizations	`Organizations`
Performance Insights di Amazon RDS	`Performance Insights`
Amazon Pinpoint	`Pinpoint`
AWS Private Certificate Authority	`Private Certificate Authority`
Amazon Managed Service per Prometheus	`Prometheus`
AWS Proton	`Proton`
Database Amazon Quantum Ledger (Amazon QLDB)	`QLDB`
Amazon RDS	`RDS`
Amazon Redshift	`Redshift Data API`
Amazon Rekognition	`Rekognition`
AWS Resource Access Manager	`Resource Access Manager`

Servizio	Valore della dimensione **`Service`**
AWS Resource Groups	`Resource Groups`
AWS Resource Groups Tagging API	`Resource Groups Tagging API`
AWS RoboMaker	`RoboMaker`
Domini Amazon Route 53	`Route 53 Domains`
Amazon Route 53 Resolver	`Route 53 Resolver`
Amazon S3	`S3`
Amazon S3 Glacier	`Amazon S3 Glacier`
SageMaker Runtime di Amazon	`Sagemaker`
Savings Plans	`Savings Plans`
AWS Secrets Manager	`Secrets Manager`
AWS Security Hub	`Security Hub`
AWS Server Migration Service	`AWS Server Migration Service`
AWS Service Catalog AppRegistry	`Service Catalog AppRegistry`
Service Quotas (Quote di Servizio)	`Service Quotas`
AWS Shield	`Shield`
AWS Firmatario	`Signer`
Amazon Simple Notification Service	`SNS`
Amazon Simple Email Service	`SES`
Amazon Simple Queue Service	`SQS`
Archivio identità	`Identity Store`

Servizio	Valore della dimensione **`Service`**
Storage Gateway	`Storage Gateway`
AWS Support	`Support`
Amazon Simple Workflow Service	`SWF`
Amazon Textract	`Textract`
AWS IoT Things Graph	`ThingsGraph`
Amazon Timestream	`Timestream`
Amazon Transcribe	`Transcribe`
Amazon Translate	`Translate`
Trascrizione in streaming con Amazon Transcribe	`Transcribe Streaming`
AWS Transfer Family	`Transfer`
AWS WAF	`WAF`
Amazon WorkDocs	`Amazon WorkDocs`
Amazon WorkLink	`WorkLink`
Amazon WorkMail	`Amazon WorkMail`
Amazon WorkSpaces	`Workspaces`
AWS X-Ray	`X-Ray`

Alcuni servizi segnalano anche i parametri di utilizzo per API aggiuntive. Per vedere se un'API riporta le metriche di utilizzo a CloudWatch, usa la CloudWatch console per visualizzare le metriche riportate da quel servizio nel `AWS/Usage` namespace.

Per visualizzare l'elenco delle API di un servizio che segnalano le metriche di utilizzo a CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Nella scheda Tutte le metriche, scegli Utilizzo, quindi scegli Per AWS risorsa.
4. Nella casella di ricerca accanto all'elenco dei parametri, inserisci il nome del servizio. I parametri vengono filtrati in base al servizio immesso.

CloudWatch metriche di utilizzo

CloudWatch raccoglie metriche che tracciano l'utilizzo di alcune risorse. AWS Queste metriche corrispondono alle quote di servizio. AWS Il monitoraggio di questi parametri consente di gestire in modo proattivo le tue quote. Per ulteriori informazioni, consulta Visualizzazione delle quote di servizio e impostazione degli allarmi.

I parametri di utilizzo delle quote di servizio si trovano nello spazio dei nomi `AWS/Usage` e vengono raccolti ogni minuto.

Attualmente, l'unico nome di metrica in questo spazio dei nomi che pubblica è. CloudWatch `CallCount` Questo parametro viene pubblicato con le dimensioni `Resource`, `Service` e `Type`. La dimensione `Resource` specifica il nome dell'operazione API monitorata. Ad esempio, la `CallCount` metrica con le dimensioni `"Service": "CloudWatch"` `"Type": "API"` e `"Resource": "PutMetricData"` indica il numero di volte in cui l'operazione CloudWatch `PutMetricData` API è stata chiamata nel tuo account.

Il parametro `CallCount` non ha un'unità specificata. La statistica più utile per il parametro è `SUM`, che rappresenta il conteggio totale delle operazioni per il periodo di 1 minuto.

Metriche

Parametro	Descrizione
`CallCount`	Il numero di operazioni specificate eseguite nel tuo account.

Dimensioni

Dimensione	Descrizione
`Service`	Il nome del AWS servizio che contiene la risorsa. Per le metriche di CloudWatch utilizzo, il valore per questa dimensione è`CloudWatch` .
`Class`	La classe di risorsa monitorata. CloudWatch Le metriche di utilizzo dell'API utilizzano questa dimensione con un valore di. `None`
`Type`	Il tipo di risorsa monitorata. Attualmente, quando la dimensione `Service` è `CloudWatch` , l'unico valore valido per `Type` è `API`.
`Resource`	Il nome dell'operazione API. I valori validi includono i seguenti: `DeleteAlarms` , `DeleteDashboards` , `DescribeAlarmHistory` , `DescribeAlarms` , `GetDashboard` , `GetMetricData` , `GetMetricStatistics` , `ListMetrics` , `PutDashboard` e `PutMetricData`

CloudWatch tutorial

I seguenti scenari illustrano gli usi di Amazon CloudWatch. Nel primo scenario, utilizzi la CloudWatch console per creare un allarme di fatturazione che monitora AWS l'utilizzo e ti avvisa quando hai superato una determinata soglia di spesa. Nel secondo scenario, più avanzato, si utilizza la AWS Command Line Interface (AWS CLI) per pubblicare una singola metrica per un'applicazione ipotetica denominata. GetStarted

Scenari

- Monitoraggio dei costi stimati
- Pubblicazione di parametri

Scenario: monitora le spese stimate utilizzando CloudWatch

In questo scenario, crei un CloudWatch allarme Amazon per monitorare i costi stimati. Quando abiliti il monitoraggio degli addebiti stimati per il tuo AWS account, gli addebiti stimati vengono calcolati e inviati più volte al giorno CloudWatch come dati metrici.

I dati dei parametri di fatturazione sono archiviati nella regione Stati Uniti orientali (Virginia settentrionale) e rappresentano i costi a livello mondiale. Questi dati includono i costi stimati per ogni servizio utilizzato, nonché il totale complessivo stimato dei AWS costi. AWS

Puoi scegli di ricevere avvisi via e-mail nel momento in cui i costi superano una determinata soglia. Questi avvisi vengono attivati CloudWatch e i messaggi vengono inviati tramite Amazon Simple Notification Service (Amazon SNS).

> **Note**
>
> Per informazioni sull'analisi degli CloudWatch addebiti che ti sono già stati fatturati, consulta. CloudWatch fatturazione e costi

Attività

- Fase 1: Attivazione degli avvisi di fatturazione
- Fase 2: Creazione di un allarme di fatturazione

Fase 1: Attivazione degli avvisi di fatturazione

Prima di poter creare un allarme per gli addebiti stimati, devi abilitare gli avvisi di fatturazione, in modo da poter monitorare gli AWS addebiti stimati e creare un allarme utilizzando i dati metrici di fatturazione. Dopo aver attivato gli avvisi di fatturazione, non è possibile disabilitare la raccolta di dati, ma è possibile eliminare qualsiasi allarme di fatturazione creato.

Dopo aver attivato gli avvisi di fatturazione per la prima volta, sono necessari circa 15 minuti prima di visualizzare i dati di fatturazione e di impostare gli allarmi di fatturazione.

Requisiti

- È necessario aver effettuato l'accesso utilizzando le credenziali dell'utente root dell'account o come utente a cui è stata concessa l'autorizzazione per visualizzare le informazioni di fatturazione.
- Per gli account di fatturazione consolidata, puoi individuare i dati di fatturazione di ciascun account collegato accedendo come account di pagamento. Puoi visualizzare i dati di fatturazione per i costi stimati totali e i costi stimati per servizio per ogni account collegato aggiunto all'account consolidato.
- In un conto di fatturazione consolidato, le metriche degli account collegati ai membri vengono acquisite solo se il conto pagatore abilita la preferenza Ricevi avvisi di fatturazione. Se si modifica il conto gestione/pagante, è necessario abilitare gli avvisi di fatturazione nel nuovo conto gestione/pagante.
- L'account non deve far parte dell'Amazon Partner Network (APN) perché le metriche di fatturazione non vengono pubblicate sugli account APN. CloudWatch Per ulteriori informazioni, consulta la pagina Partner Network AWS.

Attivazione del monitoraggio dei costi stimati

1. Apri la console all'indirizzo https://console.aws.amazon.com/billing/ AWS Billing .
2. Nel riquadro di navigazione, scegli Billing preferences (Preferenze di fatturazione).
3. Per Preferenze di avviso, scegli Modifica.

4. Scegli Ricevi avvisi di CloudWatch fatturazione.
5. Scegli Save preferences (Salva preferenze).

Fase 2: Creazione di un allarme di fatturazione

> **Important**
>
> Prima di creare un allarme di fatturazione, imposta la regione su Stati Uniti orientali (Virginia settentrionale). I dati dei parametri di fatturazione sono archiviati in questa regione e rappresentano i costi a livello mondiale. Devi inoltre abilitare gli avvisi di fatturazione per il tuo l'account o per l'account gestione/pagante se utilizzi la fatturazione consolidata. Per ulteriori informazioni, consulta la sezione Fase 1: Attivazione degli avvisi di fatturazione.

In questa procedura, crei un allarme che invia una notifica quando gli addebiti stimati AWS superano una soglia definita.

Per creare un allarme di fatturazione utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), quindi Create alarm (Tutti gli allarmi).
3. Scegli Crea allarme.
4. Scegli Select Metric (Seleziona parametro). In Browse (Sfoglia), scegli Billing (Fatturazione), quindi seleziona Total Estimated Charge (Addebito totale stimato).

   > **Note**
   >
   > Se non visualizzi il parametro Fatturazione/Addebito totale stimato, abilita gli avvisi di fatturazione e passa alla Regione Stati Uniti orientali (Virginia settentrionale). Per ulteriori informazioni, consulta Attivazione di avvisi di fatturazione.

5. Seleziona la casella per la EstimatedChargesmetrica, quindi scegli Seleziona metrica.
6. Per Statistic (Statistica), scegli Maximum (Massima).
7. Per Period (Periodo), scegli 6 hours (6 ore).
8. For Threshold type (Tipo di soglia), scegli Static (Statica).
9. Per EstimatedCharges Whenever is.. , scegli Maggiore.

10. Per than . . ., definisci il valore che desideri attivi l'allarme. Per un esempio, **`200`** USD.

    I valori delle EstimatedChargesmetriche sono solo in dollari USA (USD) e la conversione di valuta è fornita da Amazon Services LLC. Per ulteriori informazioni, consulta Cos'è AWS Billing? .

    > **Note**
    >
    > Dopo aver definito un valore di soglia, il grafico di anteprima mostra i costi stimati per il mese corrente.

11. Scegli Configurazione aggiuntiva e completa le seguenti operazioni:

    - In Datapoints to alarm (Data point per allarme), specifica 1 out of 1 (1 su 1).
    - In Missing data treatment (Trattamento dei dati mancanti), scegli Treat missing data as missing (Tratta i dati mancanti come mancanti).

12. Seleziona Successivo.
13. In Notifica, assicurati che sia selezionata l'opzione In allarme. Quindi, specifica un argomento Amazon SNS per segnalare quando l'allarme si trova nello stato `ALARM`. L'argomento Amazon SNS può includere il tuo indirizzo e-mail in modo da ricevere un messaggio e-mail quando l'importo della fatturazione supera la soglia specificata.

    Puoi selezionare un argomento Amazon SNS esistente, crearne uno nuovo o utilizzare un ARN dell'argomento per inviare una notifica a un altro account. Se vuoi che l'allarme invii più notifiche per lo stesso stato di allarme o per stati di allarme diversi, scegli Add notification (Aggiungi notifica).
14. Seleziona Successivo.
15. In Name and description (Nome e descrizione), immetti un nome per l'allarme.

    - (Facoltativo) Immetti una descrizione per l'allarme.

16. Seleziona Successivo.
17. In Preview and create (Anteprima e creazione), verifica che la configurazione sia corretta, quindi seleziona Create alarm (Crea allarme).

Fase 3: Controllo dello stato dell'allarme

Ora, verifica lo stato dell'allarme di fatturazione appena creato.

Controllo dello stato dell'allarme

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Se necessario modifica la regione in Stati Uniti orientali (Virginia settentrionale). I dati dei parametri di fatturazione sono archiviati in questa regione e riflettono i costi mondiali.
3. Nel pannello di navigazione, seleziona Alarms (Allarmi).
4. Seleziona la casella di controllo accanto all'allarme. Finché l'abbonamento non è confermato, viene visualizzato come "In attesa di conferma". Dopo avere confermato la sottoscrizione, aggiorna la console per visualizzare lo stato aggiornato.

Fase 4: Modifica di un allarme di fatturazione

Ad esempio, potresti voler aumentare la quantità di denaro che spendi AWS ogni mese da 200 a 400$. Puoi modificare l'allarme di fatturazione esistente e aumentare l'importo monetario che deve essere superato per l'attivazione dell'allarme.

Modifica di un allarme di fatturazione

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Se necessario modifica la regione in Stati Uniti orientali (Virginia settentrionale). I dati dei parametri di fatturazione sono archiviati in questa regione e riflettono i costi mondiali.
3. Nel pannello di navigazione, seleziona Alarms (Allarmi).
4. Seleziona la casella di controllo accanto all'allarme e scegli Actions (Operazioni), Modify (Modifica).
5. Per Ogni volta che i miei AWS addebiti totali mensili superano, specifica il nuovo importo da superare per far scattare l'allarme e invia una notifica via e-mail.
6. Seleziona Salva modifiche.

Fase 5: Eliminazione di un allarme di fatturazione

Se l'allarme di fatturazione non è più necessario, puoi eliminarlo.

Per eliminare un allarme di fatturazione

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Se necessario modifica la regione in Stati Uniti orientali (Virginia settentrionale). I dati dei parametri di fatturazione sono archiviati in questa regione e riflettono i costi mondiali.
3. Nel pannello di navigazione, seleziona Alarms (Allarmi).
4. Seleziona la casella di controllo accanto all'allarme e scegli Actions (Operazioni), Delete (Elimina).
5. Quando viene richiesta la conferma, seleziona Yes, Delete (Sì, elimina).

Scenario: pubblica le metriche su CloudWatch

In questo scenario, si utilizza la AWS Command Line Interface (AWS CLI) per pubblicare una singola metrica per un'applicazione ipotetica denominata. GetStarted Se non avete ancora installato e configurato il AWS CLI, consultate Getting Set Up with the AWS Command Line Interface nella Guida per l'utente.AWS Command Line Interface

Attività

- Fase 1: Definizione della configurazione dei dati
- Passaggio 2: aggiungere metriche a CloudWatch
- Passaggio 3: Ottieni statistiche da CloudWatch
- Fase 4: Visualizzazione di grafici con la console

Fase 1: Definizione della configurazione dei dati

In questo scenario, pubblichi punti dati in grado di monitorare la latenza delle richieste per l'applicazione. Scegli dei nomi per il parametro e per lo spazio dei nomi per te significativi. Per questo esempio, assegnate un nome alla metrica RequestLatencye inserite tutti i punti dati nel GetStartednamespace.

Pubblica vari punti dati che rappresentano collettivamente tre ore di dati di latenza. I dati non elaborati comprendono 15 letture di latenza di richieste distribuite in tre ore. Ogni lettura è in millisecondi:

- Ora uno: 87, 51, 125, 235
- Ora due: 121, 113, 189, 65, 89
- Ora tre: 100, 47, 133, 98, 100, 328

È possibile pubblicare i dati CloudWatch come punti dati singoli o come set aggregato di punti dati denominato set di statistiche. Puoi aggregare i parametri con una granularità fino a un minuto. È possibile pubblicare i punti dati aggregati CloudWatch come set di statistiche con quattro chiavi predefinite:`Sum`,, `Minimum` e. `Maximum` `SampleCount`

Pubblica i punti dati provenienti dalla prima ora come singoli punti dati. Per i dati provenienti dalla seconda e terza ora, aggrega i punti dati e pubblica un set di statistiche per ogni ora. I valori della chiave sono visualizzati nella tabella seguente.

Ora	Dati non elaborati	Somma	Minimo	Massimo	SampleCount
1	87				
1	51				
1	125				
1	235				
2	121, 113, 189, 65, 89	577	65	189	5
3	100, 47, 133, 98, 100, 328	806	47	328	6

Passaggio 2: aggiungere metriche a CloudWatch

Dopo aver definito la configurazione dei dati, sei pronto per aggiungere dati.

Per pubblicare punti dati su CloudWatch

1. Al prompt dei comandi, put-metric-dataesegui i comandi seguenti per aggiungere dati per la prima ora. Sostituisci il timestamp di esempio con un timestamp riferito a due ore nel passato, in formato UTC (Universal Coordinated Time).

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 87 --unit Milliseconds
```

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 51 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 125 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 235 --unit Milliseconds
```

2. Aggiungi i dati alla seconda ora utilizzando un timestamp riferito a un'ora in avanti rispetto alla prima ora.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T21:30:00Z --statistic-values
 Sum=577,Minimum=65,Maximum=189,SampleCount=5 --unit Milliseconds
```

3. Aggiungi i dati alla terza ora, omettendo il timestamp al valore predefinito per l'ora corrente.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--statistic-values Sum=806,Minimum=47,Maximum=328,SampleCount=6 --unit Milliseconds
```

Passaggio 3: Ottieni statistiche da CloudWatch

Ora che hai pubblicato le metriche su CloudWatch, puoi recuperare le statistiche basate su tali metriche utilizzando il get-metric-statisticscomando seguente. Assicurati di specificare `--start-time` e `--end-time` in momenti sufficientemente distanti nel passato in modo da coprire il primo timestamp pubblicato.

```
aws cloudwatch get-metric-statistics --namespace GetStarted --metric-name
 RequestLatency --statistics Average \
--start-time 2016-10-14T00:00:00Z --end-time 2016-10-15T00:00:00Z --period 60
```

Di seguito è riportato un output di esempio:

```
{
 "Datapoints": [],
 "Label": "Request:Latency"
```

```
}
```

Fase 4: Visualizzazione di grafici con la console

Dopo aver pubblicato le metriche su CloudWatch, puoi utilizzare la CloudWatch console per visualizzare grafici statistici.

Visualizzazione dei grafici delle statistiche nella console

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello Navigation (Navigazione), seleziona Metrics (Parametri).
3. Nella scheda Tutte le metriche, nella casella di ricerca, digita RequestLatencye premi Invio.
4. Seleziona la casella di controllo per la RequestLatencymetrica. Verrà visualizzato un grafico dei dati del parametro nel riquadro superiore.

Per ulteriori informazioni, consulta Rappresentazione grafica dei parametri.

Utilizzo con un SDK CloudWatch AWS

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK for C++	AWS SDK for C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK for Go	AWS SDK for Go esempi di codice
AWS SDK for Java	AWS SDK for Java esempi di codice
AWS SDK for JavaScript	AWS SDK for JavaScript esempi di codice
SDK AWS for Kotlin	SDK AWS for Kotlin esempi di codice
AWS SDK for .NET	AWS SDK for .NET esempi di codice
AWS SDK for PHP	AWS SDK for PHP esempi di codice
AWS Tools for PowerShell	Strumenti per esempi di PowerShell codice
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) esempi di codice
AWS SDK for Ruby	AWS SDK for Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

Per esempi specifici CloudWatch, vedereEsempi di codice per l' CloudWatch utilizzo degli AWS SDK.

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

Esempi di codice per l' CloudWatch utilizzo degli AWS SDK

I seguenti esempi di codice mostrano come utilizzarlo CloudWatch con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Esempi cross-service: applicazioni di esempio che funzionano su più servizi Servizi AWS.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

Salve CloudWatch

I seguenti esempi di codice mostrano come iniziare a utilizzare CloudWatch.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
using Amazon.CloudWatch;
using Amazon.CloudWatch.Model;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;
```

```
namespace CloudWatchActions;

public static class HelloCloudWatch
{
    static async Task Main(string[] args)
    {
        // Use the AWS .NET Core Setup package to set up dependency injection for
 the Amazon CloudWatch service.
        // Use your AWS profile name, or leave it blank to use the default
 profile.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonCloudWatch>()
            ).Build();

        // Now the client is available for injection.
        var cloudWatchClient =
 host.Services.GetRequiredService<IAmazonCloudWatch>();

        // You can use await and any of the async methods to get a response.
        var metricNamespace = "AWS/Billing";
        var response = await cloudWatchClient.ListMetricsAsync(new
 ListMetricsRequest
        {
            Namespace = metricNamespace
        });
        Console.WriteLine($"Hello Amazon CloudWatch! Following are some metrics
 available in the {metricNamespace} namespace:");
        Console.WriteLine();
        foreach (var metric in response.Metrics.Take(5))
        {
            Console.WriteLine($"\tMetric: {metric.MetricName}");
            Console.WriteLine($"\tNamespace: {metric.Namespace}");
            Console.WriteLine($"\tDimensions: {string.Join(", ",
 metric.Dimensions.Select(m => $"{m.Name}:{m.Value}"))}");
            Console.WriteLine();
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListMetricssezione AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloService {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> System.out.println(" Retrieved metric is:
 " + metrics.metricName()));

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListMetricssezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:
https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
 */
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
           <namespace>
        Where:
           namespace - The namespace to filter against (for example, AWS/EC2).
    """

    if (args.size != 1) {
        println(usage)
        exitProcess(0)
    }

    val namespace = args[0]
    listAllMets(namespace)
}

suspend fun listAllMets(namespaceVal: String?) {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listMetricsPaginated(request)
            .transform { it.metrics?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.metricName}")
                println("Namespace is ${obj.namespace}")
            }
    }
}
```

- Per i dettagli sull'API, ListMetricsconsulta AWS SDK for Kotlin API reference.

Esempi di codice

Azioni per l'utilizzo degli SDK CloudWatch AWS

I seguenti esempi di codice mostrano come eseguire CloudWatch azioni individuali con gli AWS SDK. Questi estratti richiamano l' CloudWatch API e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. Ogni esempio include un collegamento a GitHub, dove è possibile trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta Amazon CloudWatch API Reference.

Esempi

- Utilizzo DeleteAlarms con un AWS SDK o una CLI
- Utilizzo DeleteAnomalyDetector con un AWS SDK o una CLI
- Utilizzo DeleteDashboards con un AWS SDK o una CLI
- Utilizzo DescribeAlarmHistory con un AWS SDK o una CLI
- Utilizzo DescribeAlarms con un AWS SDK o una CLI
- Utilizzo DescribeAlarmsForMetric con un AWS SDK o una CLI
- Utilizzo DescribeAnomalyDetectors con un AWS SDK o una CLI
- Utilizzo DisableAlarmActions con un AWS SDK o una CLI
- Utilizzo EnableAlarmActions con un AWS SDK o una CLI
- Utilizzo GetDashboard con un AWS SDK o una CLI
- Utilizzo GetMetricData con un AWS SDK o una CLI
- Utilizzo GetMetricStatistics con un AWS SDK o una CLI
- Utilizzo GetMetricWidgetImage con un AWS SDK o una CLI
- Utilizzo ListDashboards con un AWS SDK o una CLI
- Utilizzo ListMetrics con un AWS SDK o una CLI
- Utilizzo PutAnomalyDetector con un AWS SDK o una CLI
- Utilizzo PutDashboard con un AWS SDK o una CLI
- Utilizzo PutMetricAlarm con un AWS SDK o una CLI
- Utilizzo PutMetricData con un AWS SDK o una CLI

Utilizzo **`DeleteAlarms`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAlarms`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base sugli allarmi
- Inizia con parametri, pannelli di controllo e allarmi
- Gestione di parametri e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Delete a list of alarms from CloudWatch.
/// </summary>
/// <param name="alarmNames">A list of names of alarms to delete.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteAlarms(List<string> alarmNames)
{
    var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
        new DeleteAlarmsRequest()
        {
            AlarmNames = alarmNames
        });

    return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
}
```

- Per i dettagli sull'API, consulta la DeleteAlarmssezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DeleteAlarmsRequest.h>
#include <iostream>
```

Eliminare l'allarme.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DeleteAlarmsRequest request;
        request.AddAlarmNames(alarm_name);

        auto outcome = cw.DeleteAlarms(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to delete CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully deleted CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Per i dettagli sull'API, consulta la DeleteAlarmssezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per eliminare un allarme

L'esempio seguente utilizza il `delete-alarms` comando per eliminare l' CloudWatch allarme Amazon denominato «myalarm»:

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Output:

```
This command returns to the prompt if successful.
```

- Per i dettagli sull'API, consulta DeleteAlarms AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class DeleteAlarm {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to delete (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_2;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        deleteCWAlarm(cw, alarmName);
        cw.close();
    }

    public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
        try {
            DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.deleteAlarms(request);
            System.out.printf("Successfully deleted alarm %s", alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DeleteAlarmssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import { DeleteAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteAlarmsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Creare il client in un modulo separato ed esportarlo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DeleteAlarmssezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmNames: ["Web_Server_CPU_Utilization"],
};

cw.deleteAlarms(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DeleteAlarmssezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}
```

- Per i dettagli sull'API, DeleteAlarmsconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
```

```
        """
        self.cloudwatch_resource = cloudwatch_resource


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

- Per i dettagli sull'API, consulta DeleteAlarms AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
        lo_cwt->deletealarms(
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarms deleted.' TYPE 'I'.
      CATCH /aws1/cx_cwtresourcenotfound .
        MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeleteAlarmsconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteAnomalyDetector`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteAnomalyDetector`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete a single metric anomaly detector.
    /// </summary>
    /// <param name="anomalyDetector">The anomaly detector to delete.</param>
```

```
   /// <returns>True if successful.</returns>
   public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
   {
       var deleteAnomalyDetectorResponse = await
_amazonCloudWatch.DeleteAnomalyDetectorAsync(
           new DeleteAnomalyDetectorRequest()
           {
               SingleMetricAnomalyDetector = anomalyDetector
           });

       return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
   }
```

- Per i dettagli sull'API, consulta la DeleteAnomalyDetectorsezione AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
```

```
            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.deleteAnomalyDetector(request);
            System.out.println("Successfully deleted the Anomaly Detector.");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteAnomalyDetectorsezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
```

```
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}
```

- Per i dettagli sull'API, DeleteAnomalyDetectorconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteDashboards`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteDashboards`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
_amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteDashboardssezione AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
        try {
            DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                    .dashboardNames(dashboardName)
                    .build();
            cw.deleteDashboards(dashboardsRequest);
            System.out.println(dashboardName + " was successfully deleted.");

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteDashboardssezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}
```

- Per i dettagli sull'API, DeleteDashboardsconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: elimina la dashboard specificata, richiedendone conferma prima di procedere. Per ignorare la conferma, aggiungi l'interruttore -Force al comando.

```
Remove-CWDashboard -DashboardName Dashboard1
```

- Per i dettagli sull'API, vedere DeleteDashboardsin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeAlarmHistory`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeAlarmHistory`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
```

```
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }
```

- Per i dettagli sull'API, consulta la DescribeAlarmHistorysezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per recuperare la cronologia di un allarme

L'esempio seguente utilizza il `describe-alarm-history` comando per recuperare la cronologia dell'CloudWatch allarme Amazon denominato «myalarm»:

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type
 StateUpdate
```

Output:

```
{
    "AlarmHistoryItems": [
        {
            "Timestamp": "2014-04-09T18:59:06.442Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"ALARM\",\"stateReason\":\"testing purposes\"},\"newState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were not
 greater than the threshold (70.0). The most recent datapoints: [38.958,
 40.292].\",\"stateReasonData\":{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}}}",
            "HistorySummary": "Alarm updated from ALARM to OK"
        },
        {
            "Timestamp": "2014-04-09T18:59:05.805Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were
 not greater than the threshold (70.0). The most recent datapoints:
 [38.839999999999996, 39.714].\",\"stateReasonData\":{\"version\":
\"1.0\",\"queryDate\":\"2014-03-11T22:45:41.569+0000\",\"startDate\":
\"2014-03-11T22:30:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.839999999999996,39.714],\"threshold\":70.0}},\"newState
\":{\"stateValue\":\"ALARM\",\"stateReason\":\"testing purposes\"}}",
            "HistorySummary": "Alarm updated from OK to ALARM"
        }
    ]
}
```


- Per i dettagli sull'API, consulta AWS CLI Command DescribeAlarmHistoryReference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String alarmName = rootNode.findValue("exampleAlarmName").asText();

           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();
           DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                   .startDate(start)
                   .endDate(endDate)
                   .alarmName(alarmName)
                   .historyItemType(HistoryItemType.ACTION)
                   .build();

           DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
           List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
           if (historyItems.isEmpty()) {
               System.out.println("No alarm history data found for " + alarmName
+ ".");
           } else {
               for (AlarmHistoryItem item : historyItems) {
                   System.out.println("History summary: " +
item.historySummary());
                   System.out.println("Time stamp: " + item.timestamp());
               }
```

```
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta la DescribeAlarmHistorysezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
```

```
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}
```

- Per i dettagli sull'API, DescribeAlarmHistoryconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeAlarms`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeAlarms`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base sugli allarmi
- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
        var paginatedDescribeAlarms =
_amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }
```

- Per i dettagli sull'API, consulta la DescribeAlarmssezione AWS SDK for .NET API Reference.

CLI

AWS CLI

Per elencare le informazioni di un allarme

L'esempio seguente utilizza il comando `describe-alarms` per fornire informazioni sull'allarme denominato "myalarm":

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Output:

```
{
```

```
    "MetricAlarms": [
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:myalarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myalarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Per i dettagli sull'API, consulta DescribeAlarms AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta la DescribeAlarmssezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}
```

- Per i dettagli sull'API, DescribeAlarmsconsulta AWS SDK for Kotlin API reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require "aws-sdk-cloudwatch"

# Lists the names of available Amazon CloudWatch alarms.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   list_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def list_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms
  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts alarm.alarm_name
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end
```

- Per i dettagli sull'API, consulta la DescribeAlarmssezione AWS SDK for Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
        oo_result = lo_cwt->describealarms(                " oo_result is
 returned for testing purposes. "
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarms retrieved.' TYPE 'I'.
```

```
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DescribeAlarmsconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeAlarmsForMetric`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeAlarmsForMetric`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Inizia con parametri, pannelli di controllo e allarmi
- Gestione di parametri e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
```

```
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }
```

- Per i dettagli sull'API, consulta la DescribeAlarmsForMetricsezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DescribeAlarmsRequest.h>
#include <aws/monitoring/model/DescribeAlarmsResult.h>
#include <iomanip>
#include <iostream>
```

Descrive gli allarmi.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DescribeAlarmsRequest request;
```

```
        request.SetMaxRecords(1);

        bool done = false;
        bool header = false;
        while (!done)
        {
            auto outcome = cw.DescribeAlarms(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to describe CloudWatch alarms:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left <<
                    std::setw(32) << "Name" <<
                    std::setw(64) << "Arn" <<
                    std::setw(64) << "Description" <<
                    std::setw(20) << "LastUpdated" <<
                    std::endl;
                header = true;
            }

            const auto &alarms = outcome.GetResult().GetMetricAlarms();
            for (const auto &alarm : alarms)
            {
                std::cout << std::left <<
                    std::setw(32) << alarm.GetAlarmName() <<
                    std::setw(64) << alarm.GetAlarmArn() <<
                    std::setw(64) << alarm.GetAlarmDescription() <<
                    std::setw(20) <<
                    alarm.GetAlarmConfigurationUpdatedTimestamp().ToGmtString(
                        SIMPLE_DATE_FORMAT_STR) <<
                    std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Per i dettagli sull'API, consulta la DescribeAlarmsForMetricsezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per visualizzare informazioni sugli allarmi associati a un parametro

L'esempio seguente utilizza il comando `describe-alarms-for-metric` per visualizzare informazioni su eventuali allarmi associati al parametro CPUUtilization di Amazon EC2 e all'istanza con l'ID i-0c986c72.:

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

Output:

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 10,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm2",
            "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
            "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:NotifyMe"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2013-10-30T03:03:51.479+0000\",\"startDate\":\"2013-10-30T02:08:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm2",
            "Dimensions": [
```

```
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 10 datapoints were not
 greater than or equal to the threshold (70.0). The most recent datapoints:
 [40.760000000000005, 41.316].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        },
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
```

```
            "OKActions": [],
            "ActionsEnabled": false,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Per i dettagli sull'API, consulta DescribeAlarmsForMetric AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           boolean hasAlarm = false;
           int retries = 10;

           DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           while (!hasAlarm && retries > 0) {
```

```
                DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
                hasAlarm = response.hasMetricAlarms();
                retries--;
                Thread.sleep(20000);
                System.out.println(".");
            }
            if (!hasAlarm)
                System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
            else
                System.out.println("Alarm state found for " + customMetricName +
".");

        } catch (CloudWatchException | IOException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta la DescribeAlarmsForMetricsezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import { DescribeAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DescribeAlarmsCommand({
```

```
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Creare il client in un modulo separato ed esportarlo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DescribeAlarmsForMetricsezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.describeAlarms({ StateValue: "INSUFFICIENT_DATA" }, function (err, data) {
```

```
  if (err) {
    console.log("Error", err);
  } else {
    // List the names of all current alarms in the console
    data.MetricAlarms.forEach(function (item, index, array) {
      console.log(item.AlarmName);
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DescribeAlarmsForMetricsezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
```

```
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
            println(".")
        }
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}
```

- Per i dettagli sull'API, DescribeAlarmsForMetricconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
```

```
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
 metric_name)
        return alarm_iter
```

- Per i dettagli sull'API, consulta DescribeAlarmsForMetric AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   describe_metric_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def describe_metric_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms

  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts "-" * 16
      puts "Name:           " + alarm.alarm_name
      puts "State value:    " + alarm.state_value
      puts "State reason:   " + alarm.state_reason
      puts "Metric:         " + alarm.metric_name
      puts "Namespace:      " + alarm.namespace
      puts "Statistic:      " + alarm.statistic
```

```
      puts "Period:         " + alarm.period.to_s
      puts "Unit:           " + alarm.unit.to_s
      puts "Eval. periods:  " + alarm.evaluation_periods.to_s
      puts "Threshold:      " + alarm.threshold.to_s
      puts "Comp. operator: " + alarm.comparison_operator

      if alarm.key?(:ok_actions) && alarm.ok_actions.count.positive?
        puts "OK actions:"
        alarm.ok_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:alarm_actions) && alarm.alarm_actions.count.positive?
        puts "Alarm actions:"
        alarm.alarm_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:insufficient_data_actions) &&
          alarm.insufficient_data_actions.count.positive?
        puts "Insufficient data actions:"
        alarm.insufficient_data_actions.each do |a|
          puts "  " + a
        end
      end

      puts "Dimensions:"
      if alarm.key?(:dimensions) && alarm.dimensions.count.positive?
        alarm.dimensions.each do |d|
          puts "  Name: " + d.name + ", Value: " + d.value
        end
      else
        puts "  None for this alarm."
      end
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end
```

```
# Example usage:
def run_me
  region = ""

  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
    puts "Usage:   ruby cw-ruby-example-show-alarms.rb REGION"
    puts "Example: ruby cw-ruby-example-show-alarms.rb us-east-1"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = "us-east-1"
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  puts "Available alarms:"
  describe_metric_alarms(cloudwatch_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Per i dettagli sull'API, consulta la DescribeAlarmsForMetricsezione AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo con un SDK CloudWatch AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeAnomalyDetectors`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeAnomalyDetectors`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Describe anomaly detectors for a metric and namespace.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The metric of the anomaly detectors.</param>
    /// <returns>The list of detectors.</returns>
    public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
    {
        List<AnomalyDetector> detectors = new List<AnomalyDetector>();
        var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
            new DescribeAnomalyDetectorsRequest()
            {
                MetricName = metricName,
                Namespace = metricNamespace
            });

        await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
        {
            detectors.Add(data);
        }

        return detectors;
    }
```

- Per i dettagli sull'API, consulta la DescribeAnomalyDetectorssezione AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                   .maxResults(10)
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
           List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
           for (AnomalyDetector detector : anomalyDetectorList) {
               System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
               System.out.println("State: " + detector.stateValue());
           }

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
```

```
        }
    }
```

- Per i dettagli sull'API, consulta la DescribeAnomalyDetectorssezione AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}
```

- Per i dettagli sull'API, DescribeAnomalyDetectorsconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DisableAlarmActions`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DisableAlarmActions`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base sugli allarmi
- Gestione di parametri e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });
```

```
        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DisableAlarmActionssezione AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DisableAlarmActionsRequest.h>
#include <iostream>
```

Disattiva le operazioni di allarme.

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::DisableAlarmActionsRequest
disableAlarmActionsRequest;
        disableAlarmActionsRequest.AddAlarmNames(alarm_name);

        auto disableAlarmActionsOutcome =
cw.DisableAlarmActions(disableAlarmActionsRequest);
        if (!disableAlarmActionsOutcome.IsSuccess())
        {
            std::cout << "Failed to disable actions for alarm " << alarm_name <<
                ": " << disableAlarmActionsOutcome.GetError().GetMessage() <<
                std::endl;
```

```
        }
        else
        {
            std::cout << "Successfully disabled actions for alarm " <<
                alarm_name << std::endl;
        }
```

- Per i dettagli sull'API, consulta la DisableAlarmActionssezione AWS SDK for C++ API Reference.

CLI

AWS CLI

Per disattivare le operazioni su un allarme

L'esempio seguente utilizza il comando `disable-alarm-actions` per disabilitare tutte le operazioni per l'allarme denominato myalarm.:

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

In caso di esito positivo, il comando torna al prompt.

- Per i dettagli sull'API, consulta DisableAlarmActions AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import
 software.amazon.awssdk.services.cloudwatch.model.DisableAlarmActionsRequest;
```

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DisableAlarmActions {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to disable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        disableActions(cw, alarmName);
        cw.close();
    }

    public static void disableActions(CloudWatchClient cw, String alarmName) {
        try {
            DisableAlarmActionsRequest request =
 DisableAlarmActionsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.disableAlarmActions(request);
```

```
            System.out.printf("Successfully disabled actions on alarm %s",
 alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, consulta la DisableAlarmActionssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import { DisableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DisableAlarmActionsCommand({
    AlarmNames: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};
```

```
export default run();
```

Creare il client in un modulo separato ed esportarlo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DisableAlarmActionssezione AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.disableAlarmActions(
  { AlarmNames: ["Web_Server_CPU_Utilization"] },
  function (err, data) {
    if (err) {
      console.log("Error", err);
    } else {
      console.log("Success", data);
    }
  }
);
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, consulta la DisableAlarmActionssezione AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun disableActions(alarmName: String) {

    val request = DisableAlarmActionsRequest {
        alarmNames = listOf(alarmName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.disableAlarmActions(request)
        println("Successfully disabled actions on alarm $alarmName")
    }
}
```

- Per i dettagli sull'API, DisableAlarmActionsconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Per i dettagli sull'API, consulta DisableAlarmActions AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Disables an alarm in Amazon CloudWatch.
#
# Prerequisites.
#
# - The alarm to disable.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm to disable.
# @return [Boolean] true if the alarm was disabled; otherwise, false.
# @example
#   exit 1 unless alarm_actions_disabled?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket'
#   )
def alarm_actions_disabled?(cloudwatch_client, alarm_name)
  cloudwatch_client.disable_alarm_actions(alarm_names: [alarm_name])
  return true
rescue StandardError => e
  puts "Error disabling alarm actions: #{e.message}"
  return false
end

# Example usage:
def run_me
  alarm_name = "ObjectsInBucket"
  alarm_description = "Objects exist in this bucket for more than 1 day."
  metric_name = "NumberOfObjects"
```

```
  # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
  # the alarm transitions to the ALARM state.
  alarm_actions = ["arn:aws:sns:us-
east-1:111111111111:Default_CloudWatch_Alarms_Topic"]
  namespace = "AWS/S3"
  statistic = "Average"
  dimensions = [
    {
      name: "BucketName",
      value: "doc-example-bucket"
    },
    {
      name: "StorageType",
      value: "AllStorageTypes"
    }
  ]
  period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
  unit = "Count"
  evaluation_periods = 1 # More than one day.
  threshold = 1 # One object.
  comparison_operator = "GreaterThanThreshold" # More than one object.
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  if alarm_created_or_updated?(
    cloudwatch_client,
    alarm_name,
    alarm_description,
    metric_name,
    alarm_actions,
    namespace,
    statistic,
    dimensions,
    period,
    unit,
    evaluation_periods,
    threshold,
    comparison_operator
  )
    puts "Alarm '#{alarm_name}' created or updated."
  else
    puts "Could not create or update alarm '#{alarm_name}'."
```

```
  end

  if alarm_actions_disabled?(cloudwatch_client, alarm_name)
    puts "Alarm '#{alarm_name}' disabled."
  else
    puts "Could not disable alarm '#{alarm_name}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- Per i dettagli sull'API, DisableAlarmActionsconsulta AWS SDK for Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    "Disables actions on the specified alarm. "
    TRY.
        lo_cwt->disablealarmactions(
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarm actions disabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DisableAlarmActionsconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`EnableAlarmActions`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`EnableAlarmActions`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Gestione di parametri e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, EnableAlarmActionsconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/EnableAlarmActionsRequest.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Attiva le operazioni di allarme.

```
    Aws::CloudWatch::CloudWatchClient cw;
    Aws::CloudWatch::Model::PutMetricAlarmRequest request;
    request.SetAlarmName(alarm_name);
    request.SetComparisonOperator(
        Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
    request.SetEvaluationPeriods(1);
    request.SetMetricName("CPUUtilization");
    request.SetNamespace("AWS/EC2");
    request.SetPeriod(60);
    request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
    request.SetThreshold(70.0);
    request.SetActionsEnabled(false);
    request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
    request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);
    request.AddAlarmActions(actionArn);

    Aws::CloudWatch::Model::Dimension dimension;
    dimension.SetName("InstanceId");
```

```
    dimension.SetValue(instanceId);
    request.AddDimensions(dimension);

    auto outcome = cw.PutMetricAlarm(request);
    if (!outcome.IsSuccess())
    {
        std::cout << "Failed to create CloudWatch alarm:" <<
            outcome.GetError().GetMessage() << std::endl;
        return;
    }

    Aws::CloudWatch::Model::EnableAlarmActionsRequest enable_request;
    enable_request.AddAlarmNames(alarm_name);

    auto enable_outcome = cw.EnableAlarmActions(enable_request);
    if (!enable_outcome.IsSuccess())
    {
        std::cout << "Failed to enable alarm actions:" <<
            enable_outcome.GetError().GetMessage() << std::endl;
        return;
    }

    std::cout << "Successfully created alarm " << alarm_name <<
        " and enabled actions on it." << std::endl;
```

- Per i dettagli sull'API, EnableAlarmActionsconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Per abilitare tutte le operazioni relative a un allarme

L'esempio seguente utilizza il comando `enable-alarm-actions` per abilitare tutte le operazioni per l'allarme denominato myalarm.:

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

In caso di esito positivo, il comando torna al prompt.

- Per i dettagli sull'API, consulta EnableAlarmActions AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import
 software.amazon.awssdk.services.cloudwatch.model.EnableAlarmActionsRequest;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class EnableAlarmActions {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to enable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarm = args[0];
```

```
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        enableActions(cw, alarm);
        cw.close();
    }

    public static void enableActions(CloudWatchClient cw, String alarm) {
        try {
            EnableAlarmActionsRequest request =
 EnableAlarmActionsRequest.builder()
                    .alarmNames(alarm)
                    .build();

            cw.enableAlarmActions(request);
            System.out.printf("Successfully enabled actions on alarm %s", alarm);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, EnableAlarmActionsconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import { EnableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
```

```
import { client } from "../libs/client.js";

const run = async () => {
  const command = new EnableAlarmActionsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Creare il client in un modulo separato ed esportarlo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, EnableAlarmActionsconsulta AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });
```

```
// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: true,
  AlarmActions: ["ACTION_ARN"],
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
};

cw.putMetricAlarm(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Alarm action added", data);
    var paramsEnableAlarmAction = {
      AlarmNames: [params.AlarmName],
    };
    cw.enableAlarmActions(paramsEnableAlarmAction, function (err, data) {
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Alarm action enabled", data);
      }
    });
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, EnableAlarmActionsconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun enableActions(alarm: String) {

    val request = EnableAlarmActionsRequest {
        alarmNames = listOf(alarm)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.enableAlarmActions(request)
        println("Successfully enabled actions on alarm $alarm")
    }
}
```

- Per i dettagli sull'API, EnableAlarmActionsconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class CloudWatchWrapper:
```

```
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
they
                       disabled.
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Per i dettagli sull'API, consulta EnableAlarmActions AWSSDK for Python (Boto3) API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    "Enable actions on the specified alarm."
    TRY.
        lo_cwt->enablealarmactions(
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarm actions enabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, EnableAlarmActionsconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetDashboard`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetDashboard`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Get information on a dashboard.
/// </summary>
/// <param name="dashboardName">The name of the dashboard.</param>
/// <returns>A JSON object with dashboard information.</returns>
public async Task<string> GetDashboard(string dashboardName)
{
    var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
        new GetDashboardRequest()
        {
            DashboardName = dashboardName
        });

    return dashboardResponse.DashboardBody;
}
```

- Per i dettagli sull'API, GetDashboardconsulta AWS SDK for .NET API Reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce l'arn al corpo del pannello di controllo specificato.

```
Get-CWDashboard -DashboardName Dashboard1
```

Output:

```
DashboardArn                                        DashboardBody
```

```
------------                                                  -------------
arn:aws:cloudwatch::123456789012:dashboard/Dashboard1 {...
```

- Per i dettagli sull'API, vedere GetDashboardin AWS Tools for PowerShell Cmdlet Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetMetricData`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetMetricData`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
```

```
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
 ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }
```

- Per i dettagli sull'API, GetMetricDataconsulta AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           // Set the date.
           Instant nowDate = Instant.now();

           long hours = 1;
           long minutes = 30;
           Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                   ChronoUnit.MINUTES);

           Metric met = Metric.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           MetricStat metStat = MetricStat.builder()
                   .stat("Maximum")
                   .period(1)
                   .metric(met)
                   .build();
```

```
            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, GetMetricDataconsulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }

    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }
```

```
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}
```

- Per i dettagli sull'API, GetMetricDataconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetMetricStatistics`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetMetricStatistics`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Inizia con parametri, pannelli di controllo e allarmi
- Gestione di parametri e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
```

```
   /// <returns>A collection of billing statistics.</returns>
   private static async Task<List<Datapoint>> SetupBillingStatistics()
   {
       // Make a request for EstimatedCharges with a period of one day for the
past seven days.
       var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
           "AWS/Billing",
           "EstimatedCharges",
           new List<string>() { "Maximum" },
           new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
           7,
           86400);

       billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();

       return billingStatistics;
   }

   /// <summary>
   /// Wrapper to get statistics for a specific CloudWatch metric.
   /// </summary>
   /// <param name="metricNamespace">The namespace of the metric.</param>
   /// <param name="metricName">The name of the metric.</param>
   /// <param name="statistics">The list of statistics to include.</param>
   /// <param name="dimensions">The list of dimensions to include.</param>
   /// <param name="days">The number of days in the past to include.</param>
   /// <param name="period">The period for the data.</param>
   /// <returns>A list of DataPoint objects for the statistics.</returns>
   public async Task<List<Datapoint>> GetMetricStatistics(string
metricNamespace,
       string metricName, List<string> statistics, List<Dimension> dimensions,
int days, int period)
   {
       var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
           new GetMetricStatisticsRequest()
           {
               Namespace = metricNamespace,
               MetricName = metricName,
               Dimensions = dimensions,
               Statistics = statistics,
               StartTimeUtc = DateTime.UtcNow.AddDays(-days),
               EndTimeUtc = DateTime.UtcNow,
               Period = period
```

```
            });

        return metricStatistics.Datapoints;
    }
```

- Per i dettagli sull'API, GetMetricStatisticsconsulta AWS SDK for .NET API Reference.

CLI

AWS CLI

Per ottenere 'utilizzo della CPU per un'istanza EC2

L'esempio seguente utilizza il comando `get-metric-statistics` per ottenere l'utilizzo della CPU per un'istanza EC2 con ID i-abcdef.

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time
 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --namespace
 AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

Output:

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-04-09T11:18:00Z",
            "Maximum": 44.79,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T20:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T19:18:00Z",
            "Maximum": 50.85,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T09:18:00Z",
```

```
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T03:18:00Z",
            "Maximum": 76.84,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T21:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T14:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T08:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T16:18:00Z",
            "Maximum": 45.55,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T06:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T13:18:00Z",
            "Maximum": 45.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T05:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
```

```
            "Timestamp": "2014-04-09T18:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T17:18:00Z",
            "Maximum": 52.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T07:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T02:18:00Z",
            "Maximum": 51.23,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T12:18:00Z",
            "Maximum": 47.67,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-08T23:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T10:18:00Z",
            "Maximum": 51.91,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T04:18:00Z",
            "Maximum": 47.13,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T15:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
```

```
        {
            "Timestamp": "2014-04-09T00:18:00Z",
            "Maximum": 48.16,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T01:18:00Z",
            "Maximum": 49.18,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

Specifica di più dimensioni

Nell'esempio seguente viene illustrato come specificare più dimensioni. Ogni dimensione è specificata come coppia nome/valore, con una virgola tra il nome e il valore. Più dimensioni sono separate da uno spazio. Se un unico parametro include più dimensioni, è necessario specificare un valore per ogni dimensione definita.

Per altri esempi di utilizzo del `get-metric-statistics` comando, consulta Get Statistics for a Metric nella Amazon CloudWatch Developer Guide.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --
namespace MyNameSpace --dimensions Name=InstanceID,Value=i-abcdef
 Name=InstanceType,Value=m1.small --start-time 2016-10-15T04:00:00Z --end-time
 2016-10-19T07:00:00Z --statistics Average --period 60
```

- Per i dettagli sull'API, consulta GetMetricStatistics AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
           String metricOption, String date, Dimension myDimension) {
       try {
           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();

           GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                   .endTime(endDate)
                   .startTime(start)
                   .dimensions(myDimension)
                   .metricName(metVal)
                   .namespace(nameSpace)
                   .period(86400)
                   .statistics(Statistic.fromValue(metricOption))
                   .build();

           GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
           List<Datapoint> data = response.datapoints();
           if (!data.isEmpty()) {
               for (Datapoint datapoint : data) {
                   System.out
                           .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
               }
           } else {
               System.out.println("The returned data list is empty");
           }

       } catch (CloudWatchException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

- Per i dettagli sull'API, GetMetricStatisticsconsulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}
```

- Per i dettagli sull'API, GetMetricStatisticsconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_statistics(self, namespace, name, start, end, period,
 stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
 grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
 period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
 must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
 value
                           or maximum value.
        :return: The retrieved statistics for the metric.
```

```
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats
```

- Per i dettagli sull'API, consulta GetMetricStatistics AWSSDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GetMetricWidgetImage`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`GetMetricWidgetImage`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };

        var metricImageWidgetString =
JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
```

```
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }
```

- Per i dettagli sull'API, GetMetricWidgetImageconsulta AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
       System.out.println("Getting Image data for custom metric.");
       try {
           String myJSON = "{\n" +
                   "  \"title\": \"Example Metric Graph\",\n" +
                   "  \"view\": \"timeSeries\",\n" +
                   "  \"stacked \": false,\n" +
                   "  \"period\": 10,\n" +
                   "  \"width\": 1400,\n" +
```

```
                    "  \"height\": 600,\n" +
                    "  \"metrics\": [\n" +
                    "    [\n" +
                    "      \"AWS/Billing\",\n" +
                    "      \"EstimatedCharges\",\n" +
                    "      \"Currency\",\n" +
                    "      \"USD\"\n" +
                    "    ]\n" +
                    "  ]\n" +
                    "}";

            GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                    .metricWidget(myJSON)
                    .build();

            GetMetricWidgetImageResponse response =
cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, GetMetricWidgetImageconsulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}
```

- Per i dettagli sull'API, GetMetricWidgetImageconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListDashboards`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListDashboards`.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Get a list of dashboards.
/// </summary>
/// <returns>A list of DashboardEntry objects.</returns>
public async Task<List<DashboardEntry>> ListDashboards()
{
    var results = new List<DashboardEntry>();
    var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
        new ListDashboardsRequest());
    // Get the entire list using the paginator.
    await foreach (var data in paginateDashboards.DashboardEntries)
    {
        results.Add(data);
    }

    return results;
}
```

- Per i dettagli sull'API, ListDashboardsconsulta AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, ListDashboardsconsulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
}
```

- Per i dettagli sull'API, ListDashboardsconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: restituisce la raccolta di dashboard per il tuo account.

```
Get-CWDashboardList
```

Output:

```
DashboardArn DashboardName LastModified        Size
------------ ------------- ------------        ----
arn:...      Dashboard1    7/6/2017 8:14:15 PM 252
```

Esempio 2: restituisce la raccolta di dashboard per il tuo account i cui nomi iniziano con il prefisso 'dev'.

```
Get-CWDashboardList -DashboardNamePrefix dev
```

- Per i dettagli sull'API, vedere ListDashboardsin Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ListMetrics`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ListMetrics`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Inizia con parametri, pannelli di controllo e allarmi
- Gestione di parametri e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// List metrics available, optionally within a namespace.
   /// </summary>
   /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
   /// <param name="filter">Optional dimension filter.</param>
   /// <param name="metricName">Optional metric name filter.</param>
   /// <returns>The list of metrics.</returns>
   public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
```

```
    {
        var results = new List<Metric>();
        var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
            new ListMetricsRequest
            {
                Namespace = metricNamespace,
                Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
                MetricName = metricName
            });
        // Get the entire list using the paginator.
        await foreach (var metric in paginateMetrics.Metrics)
        {
            results.Add(metric);
        }

        return results;
    }
```

- Per i dettagli sull'API, ListMetricsconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/ListMetricsRequest.h>
#include <aws/monitoring/model/ListMetricsResult.h>
#include <iomanip>
#include <iostream>
```

Elenca i parametri.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::ListMetricsRequest request;

        if (argc > 1)
        {
            request.SetMetricName(argv[1]);
        }

        if (argc > 2)
        {
            request.SetNamespace(argv[2]);
        }

        bool done = false;
        bool header = false;
        while (!done)
        {
            auto outcome = cw.ListMetrics(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to list CloudWatch metrics:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left << std::setw(48) << "MetricName" <<
                    std::setw(32) << "Namespace" << "DimensionNameValuePairs" <<
                    std::endl;
                header = true;
            }

            const auto &metrics = outcome.GetResult().GetMetrics();
            for (const auto &metric : metrics)
            {
                std::cout << std::left << std::setw(48) <<
                    metric.GetMetricName() << std::setw(32) <<
                    metric.GetNamespace();
                const auto &dimensions = metric.GetDimensions();
                for (auto iter = dimensions.cbegin();
                    iter != dimensions.cend(); ++iter)
```

```
                {
                    const auto &dimkv = *iter;
                    std::cout << dimkv.GetName() << " = " << dimkv.GetValue();
                    if (iter + 1 != dimensions.cend())
                    {
                        std::cout << ", ";
                    }
                }
                std::cout << std::endl;
            }

            const auto &next_token = outcome.GetResult().GetNextToken();
            request.SetNextToken(next_token);
            done = next_token.empty();
        }
```

- Per i dettagli sull'API, ListMetricsconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Per elencare i parametri per Amazon SNS

L'esempio `list-metrics` seguente mostra i parametri per Amazon SNS.

```
aws cloudwatch list-metrics \
    --namespace "AWS/SNS"
```

Output:

```
{
    "Metrics": [
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "PublishSize"
```

```
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
```

```
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        }
    ]
}
```

- Per i dettagli sull'API, consulta ListMetrics AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ListMetrics {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }
```

```
    public static void listMets(CloudWatchClient cw, String namespace) {
        boolean done = false;
        String nextToken = null;

        try {
            while (!done) {

                ListMetricsResponse response;
                if (nextToken == null) {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .build();

                    response = cw.listMetrics(request);
                } else {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .nextToken(nextToken)
                            .build();

                    response = cw.listMetrics(request);
                }

                for (Metric metric : response.metrics()) {
                    System.out.printf("Retrieved metric %s",
 metric.metricName());
                    System.out.println();
                }

                if (response.nextToken() == null) {
                    done = true;
                } else {
                    nextToken = response.nextToken();
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, ListMetricsconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import { ListMetricsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

export const main = () => {
  // Use the AWS console to see available namespaces and metric names. Custom
 metrics can also be created.
  // https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
viewing_metrics_with_cloudwatch.html
  const command = new ListMetricsCommand({
    Dimensions: [
      {
        Name: "LogGroupName",
      },
    ],
    MetricName: "IncomingLogEvents",
    Namespace: "AWS/Logs",
  });

  return client.send(command);
};
```

Creare il client in un modulo separato ed esportarlo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, ListMetricsconsulta AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  Dimensions: [
    {
      Name: "LogGroupName" /* required */,
    },
  ],
  MetricName: "IncomingLogEvents",
  Namespace: "AWS/Logs",
};

cw.listMetrics(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Metrics", JSON.stringify(data.Metrics));
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, ListMetricsconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}
```

- Per i dettagli sull'API, ListMetricsconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def list_metrics(self, namespace, name, recent=False):
        """
        Gets the metrics within a namespace that have the specified name.
        If the metric has no dimensions, a single metric is returned.
        Otherwise, metrics for all dimensions are returned.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param recent: When True, only metrics that have been active in the last
                       three hours are returned.
        :return: An iterator that yields the retrieved metrics.
        """
        try:
            kwargs = {"Namespace": namespace, "MetricName": name}
            if recent:
                kwargs["RecentlyActive"] = "PT3H"  # List past 3 hours only
            metric_iter = self.cloudwatch_resource.metrics.filter(**kwargs)
            logger.info("Got metrics for %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't get metrics for %s.%s.", namespace, name)
            raise
        else:
            return metric_iter
```

- Per i dettagli sull'API, consulta ListMetrics AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Lists available metrics for a metric namespace in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric.
# @example
#   list_metrics_for_namespace(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC'
#   )
def list_metrics_for_namespace(cloudwatch_client, metric_namespace)
  response = cloudwatch_client.list_metrics(namespace: metric_namespace)

  if response.metrics.count.positive?
    response.metrics.each do |metric|
      puts "  Metric name: #{metric.metric_name}"
      if metric.dimensions.count.positive?
        puts "    Dimensions:"
        metric.dimensions.each do |dimension|
          puts "      Name: #{dimension.name}, Value: #{dimension.value}"
        end
      else
        puts "No dimensions found."
      end
    end
  else
    puts "No metrics found for namespace '#{metric_namespace}'. " \
      "Note that it could take up to 15 minutes for recently-added metrics " \
      "to become available."
  end
end
```

```
# Example usage:
def run_me
  metric_namespace = "SITE/TRAFFIC"
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  # Add three datapoints.
  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisitors",
    "SiteName",
    "example.com",
    5_885.0,
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisits",
    "SiteName",
    "example.com",
    8_628.0,
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "PageViews",
    "PageURL",
    "example.html",
    18_057.0,
    "Count"
  )

  puts "Metrics for namespace '#{metric_namespace}':"
  list_metrics_for_namespace(cloudwatch_client, metric_namespace)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Per i dettagli sull'API, ListMetricsconsulta AWS SDK for Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   "The following list-metrics example displays the metrics for Amazon
CloudWatch."
   TRY.
       oo_result = lo_cwt->listmetrics(           " oo_result is returned for
testing purposes. "
         iv_namespace = iv_namespace
       ).
       DATA(lt_metrics) = oo_result->get_metrics( ).
       MESSAGE 'Metrics retrieved.' TYPE 'I'.
     CATCH /aws1/cx_cwtinvparamvalueex .
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
   ENDTRY.
```

- Per i dettagli sulle API, ListMetricsconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutAnomalyDetector`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutAnomalyDetector`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Add an anomaly detector for a single metric.
    /// </summary>
    /// <param name="anomalyDetector">A single metric anomaly detector.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
    {
        var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
            new PutAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, PutAnomalyDetectorconsulta AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.putAnomalyDetector(anomalyDetectorRequest);
           System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
```

```
    }
```

- Per i dettagli sull'API, PutAnomalyDetectorconsulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}
```

- Per i dettagli sull'API, PutAnomalyDetectorconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **PutDashboard** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutDashboard`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Inizia con parametri, pannelli di controllo e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
```

```
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
 newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
 dashboardName,
        string dashboardBody)
```

```
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }
```

- Per i dettagli sull'API, PutDashboardconsulta AWS SDK for .NET API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            PutDashboardResponse response = cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully created.");
            List<DashboardValidationMessage> messages =
response.dashboardValidationMessages();
            if (messages.isEmpty()) {
                System.out.println("There are no messages in the new Dashboard");
```

```
            } else {
                for (DashboardValidationMessage message : messages) {
                    System.out.println("Message is: " + message.message());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, PutDashboardconsulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
```

```
                }
            }
        }
    }
}
```

- Per i dettagli sull'API, PutDashboardconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: crea o aggiorna la dashboard denominata «Dashboard1» per includere due widget metrici affiancati.

```
$dashBody = @"
{
    "widgets":[
        {
             "type":"metric",
             "x":0,
             "y":0,
             "width":12,
             "height":6,
             "properties":{
                "metrics":[
                   [
                      "AWS/EC2",
                      "CPUUtilization",
                      "InstanceId",
                      "i-012345"
                   ]
                ],
                "period":300,
                "stat":"Average",
                "region":"us-east-1",
                "title":"EC2 Instance CPU"
             }
        },
        {
             "type":"metric",
             "x":12,
```

```
            "y":0,
            "width":12,
            "height":6,
            "properties":{
               "metrics":[
                  [
                     "AWS/S3",
                     "BucketSizeBytes",
                     "BucketName",
                     "MyBucketName"
                  ]
               ],
               "period":86400,
               "stat":"Maximum",
               "region":"us-east-1",
               "title":"MyBucketName bytes"
            }
         }
    ]
}
"@

Write-CWDashboard -DashboardName Dashboard1 -DashboardBody $dashBody
```

Esempio 2: crea o aggiorna il dashboard, inserendo il contenuto che descrive il dashboard nel cmdlet.

```
$dashBody = @"
{
...
}
"@

$dashBody | Write-CWDashboard -DashboardName Dashboard1
```

- Per i dettagli sull'API, vedere PutDashboardin Cmdlet Reference.AWS Tools for PowerShell

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **PutMetricAlarm** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutMetricAlarm`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Nozioni di base sugli allarmi
- Inizia con parametri, pannelli di controllo e allarmi
- Gestione di parametri e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Add a metric alarm to send an email when the metric passes a threshold.
    /// </summary>
    /// <param name="alarmDescription">A description of the alarm.</param>
    /// <param name="alarmName">The name for the alarm.</param>
    /// <param name="comparison">The type of comparison to use.</param>
    /// <param name="metricName">The name of the metric for the alarm.</param>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="threshold">The threshold value for the alarm.</param>
    /// <param name="alarmActions">Optional actions to execute when in an alarm
state.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
alarmName, ComparisonOperator comparison,
        string metricName, string metricNamespace, double threshold, List<string>
alarmActions = null!)
    {
        try
        {
```

```
            var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
                new PutMetricAlarmRequest()
                {
                    AlarmActions = alarmActions,
                    AlarmDescription = alarmDescription,
                    AlarmName = alarmName,
                    ComparisonOperator = comparison,
                    Threshold = threshold,
                    Namespace = metricNamespace,
                    MetricName = metricName,
                    EvaluationPeriods = 1,
                    Period = 10,
                    Statistic = new Statistic("Maximum"),
                    DatapointsToAlarm = 1,
                    TreatMissingData = "ignore"
                });
            return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (LimitExceededException lex)
        {
            _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
has already been reached.");
        }

        return false;
    }

    /// <summary>
    /// Add specific email actions to a list of action strings for a CloudWatch
alarm.
    /// </summary>
    /// <param name="accountId">The AccountId for the alarm.</param>
    /// <param name="region">The region for the alarm.</param>
    /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
topic for the alarm email.</param>
    /// <param name="alarmActions">Optional list of existing alarm actions to
append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
    {
        alarmActions ??= new List<string>();
```

```
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }
```

- Per i dettagli sull'API, PutMetricAlarmconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Crea l'allarme per guardare il parametro.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::PutMetricAlarmRequest request;
        request.SetAlarmName(alarm_name);
        request.SetComparisonOperator(
            Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
        request.SetEvaluationPeriods(1);
        request.SetMetricName("CPUUtilization");
        request.SetNamespace("AWS/EC2");
        request.SetPeriod(60);
        request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
        request.SetThreshold(70.0);
        request.SetActionsEnabled(false);
```

```
        request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
        request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("InstanceId");
        dimension.SetValue(instanceId);

        request.AddDimensions(dimension);

        auto outcome = cw.PutMetricAlarm(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Per i dettagli sull'API, PutMetricAlarmconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Per inviare un messaggio e-mail Amazon Simple Notification Service quando l'utilizzo della CPU supera il 70%

Nell'esempio seguente viene utilizzato il comando `put-metric-alarm` per inviare un messaggio e-mail Amazon Simple Notification Service quando l'utilizzo della CPU supera il 70%:

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/
EC2 --statistic Average --period 300 --threshold 70 --comparison-operator
 GreaterThanThreshold  --dimensions "Name=InstanceId,Value=i-12345678" --
evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic
 --unit Percent
```

In caso di esito positivo, il comando torna al prompt. Se esiste già un allarme con lo stesso nome, verrà sovrascritto dal nuovo allarme.

Per specificare più dimensioni

Nell'esempio seguente viene illustrato come specificare più dimensioni. Ogni dimensione è specificata come coppia nome/valore, con una virgola tra il nome e il valore. Più dimensioni sono separate da uno spazio:

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-
description "The default example alarm" --namespace "CW EXAMPLE METRICS" --
metric-name Default_Test --statistic Average --period 60 --evaluation-periods 3
 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --dimensions
 Name=key1,Value=value1 Name=key2,Value=value2
```

- Per i dettagli sull'API, consulta PutMetricAlarm AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static String createAlarm(CloudWatchClient cw, String fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           String alarmName = rootNode.findValue("exampleAlarmName").asText();
           String emailTopic = rootNode.findValue("emailTopic").asText();
```

```
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }
```

- Per i dettagli sull'API, PutMetricAlarmconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import { PutMetricAlarmCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  // This alarm triggers when CPUUtilization exceeds 70% for one minute.
  const command = new PutMetricAlarmCommand({
    AlarmName: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
    ComparisonOperator: "GreaterThanThreshold",
    EvaluationPeriods: 1,
    MetricName: "CPUUtilization",
    Namespace: "AWS/EC2",
    Period: 60,
    Statistic: "Average",
    Threshold: 70.0,
    ActionsEnabled: false,
    AlarmDescription: "Alarm when server CPU exceeds 70%",
    Dimensions: [
      {
        Name: "InstanceId",
        Value: process.env.EC2_INSTANCE_ID, // Set the value of EC_INSTANCE_ID to
 the Id of an existing Amazon EC2 instance.
      },
    ],
    Unit: "Percent",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
```

```
  }
};

export default run();
```

Creare il client in un modulo separato ed esportarlo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutMetricAlarmconsulta AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: false,
```

```
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
};

cw.putMetricAlarm(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutMetricAlarmconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun putMetricAlarm(alarmNameVal: String, instanceIdVal: String) {

    val dimensionOb = Dimension {
        name = "InstanceId"
        value = instanceIdVal
    }

    val request = PutMetricAlarmRequest {
        alarmName = alarmNameVal
```

```
        comparisonOperator = ComparisonOperator.GreaterThanThreshold
        evaluationPeriods = 1
        metricName = "CPUUtilization"
        namespace = "AWS/EC2"
        period = 60
        statistic = Statistic.fromValue("Average")
        threshold = 70.0
        actionsEnabled = false
        alarmDescription = "An Alarm created by the Kotlin SDK when server CPU
 utilization exceeds 70%"
        unit = StandardUnit.fromValue("Seconds")
        dimensions = listOf(dimensionOb)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(request)
        println("Successfully created an alarm with name $alarmNameVal")
    }
}
```

- Per i dettagli sull'API, PutMetricAlarmconsulta AWS SDK for Kotlin API reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource
```

```
   def create_metric_alarm(
       self,
       metric_namespace,
       metric_name,
       alarm_name,
       stat_type,
       period,
       eval_periods,
       threshold,
       comparison_op,
   ):
       """
       Creates an alarm that watches a metric.

       :param metric_namespace: The namespace of the metric.
       :param metric_name: The name of the metric.
       :param alarm_name: The name of the alarm.
       :param stat_type: The type of statistic the alarm watches.
       :param period: The period in which metric data are grouped to calculate
                      statistics.
       :param eval_periods: The number of periods that the metric must be over
the
                            alarm threshold before the alarm is set into an
alarmed
                            state.
       :param threshold: The threshold value to compare against the metric
statistic.
       :param comparison_op: The comparison operation used to compare the
threshold
                             against the metric.
       :return: The newly created alarm.
       """
       try:
           metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
           alarm = metric.put_alarm(
               AlarmName=alarm_name,
               Statistic=stat_type,
               Period=period,
               EvaluationPeriods=eval_periods,
               Threshold=threshold,
               ComparisonOperator=comparison_op,
           )
```

```
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm
```

- Per i dettagli sull'API, consulta PutMetricAlarm AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
```

```
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param theshold [Float] The value against which the specified statistic is
 compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
#       },
#       {
#         name: 'StorageType',
#         value: 'AllStorageTypes'
#       }
#     ],
#     86_400,
#     'Count',
#     1,
#     1,
#     'GreaterThanThreshold'
#   )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
```

```
  statistic,
  dimensions,
  period,
  unit,
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end
```

- Per i dettagli sull'API, PutMetricAlarmconsulta AWS SDK for Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
```

```
        lo_cwt->putmetricalarm(
          iv_alarmname                = iv_alarm_name
          iv_comparisonoperator       = iv_comparison_operator
          iv_evaluationperiods        = iv_evaluation_periods
          iv_metricname               = iv_metric_name
          iv_namespace                = iv_namespace
          iv_statistic                = iv_statistic
          iv_threshold                = iv_threshold
          iv_actionsenabled           = iv_actions_enabled
          iv_alarmdescription         = iv_alarm_description
          iv_unit                     = iv_unit
          iv_period                   = iv_period
          it_dimensions               = it_dimensions
        ).
        MESSAGE 'Alarm created.' TYPE 'I'.
      CATCH /aws1/cx_cwtlimitexceededfault.
        MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, PutMetricAlarmconsulta AWS SDK for SAP ABAP API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`PutMetricData`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`PutMetricData`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Inizia con parametri, pannelli di controllo e allarmi
- Gestione di parametri e allarmi

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Add some metric data using a call to a wrapper class.
   /// </summary>
   /// <param name="customMetricName">The metric name.</param>
   /// <param name="customMetricNamespace">The metric namespace.</param>
   /// <returns></returns>
   private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
       string customMetricNamespace)
   {
       List<MetricDatum> customData = new List<MetricDatum>();
       Random rnd = new Random();

       // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
       var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
       for (int i = 0; i < 10; i++)
       {
           var metricValue = rnd.Next(0, 100);
           customData.Add(
               new MetricDatum
               {
                   MetricName = customMetricName,
                   Value = metricValue,
                   TimestampUtc = utcNowMinus15.AddMinutes(i)
               }
           );
       }

       await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
```

```
        return customData;
    }

    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, PutMetricDataconsulta AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Includere i file richiesti.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricDataRequest.h>
#include <iostream>
```

Inserimento dei dati in un parametro.

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("UNIQUE_PAGES");
        dimension.SetValue("URLS");

        Aws::CloudWatch::Model::MetricDatum datum;
        datum.SetMetricName("PAGES_VISITED");
        datum.SetUnit(Aws::CloudWatch::Model::StandardUnit::None);
        datum.SetValue(data_point);
        datum.AddDimensions(dimension);

        Aws::CloudWatch::Model::PutMetricDataRequest request;
        request.SetNamespace("SITE/TRAFFIC");
        request.AddMetricData(datum);

        auto outcome = cw.PutMetricData(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to put sample metric data:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully put sample metric data" << std::endl;
        }
```

- Per i dettagli sull'API, PutMetricDataconsulta AWS SDK for C++ API Reference.

CLI

AWS CLI

Per pubblicare una metrica personalizzata su Amazon CloudWatch

L'esempio seguente utilizza il `put-metric-data` comando per pubblicare una metrica personalizzata su Amazon CloudWatch:

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

I valori del parametro stesso sono memorizzati nel file JSON, `metric.json`.

Ecco i contenuti del file:

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

Per ulteriori informazioni, consulta Publishing Custom Metrics nella Amazon CloudWatch Developer Guide.

Per specificare più dimensioni

Nell'esempio seguente viene illustrato come specificare più dimensioni. Ogni dimensione è specificata come coppia Name=Valore. Più dimensioni sono separate da una virgola:

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace
 MyNameSpace --unit Bytes --value 231434333 --dimensions
 InstanceID=1-23456789,InstanceType=m1.small
```

- Per i dettagli sull'API, consulta PutMetricData AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           // Set an Instant object.
           String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
           Instant instant = Instant.parse(time);

           MetricDatum datum = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1001.00)
                   .timestamp(instant)
                   .build();

           MetricDatum datum2 = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1002.00)
                   .timestamp(instant)
                   .build();

           List<MetricDatum> metricDataList = new ArrayList<>();
           metricDataList.add(datum);
           metricDataList.add(datum2);

           PutMetricDataRequest request = PutMetricDataRequest.builder()
                   .namespace(customMetricNamespace)
                   .metricData(metricDataList)
                   .build();

           cw.putMetricData(request);
```

```
            System.out.println("Added metric values for for metric " +
 customMetricName);

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, PutMetricDataconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come configurarlo ed eseguirlo nel AWS Code Examples Repository.

Importare l'SDK e i moduli client e chiamare l'API.

```
import { PutMetricDataCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  // See https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/
API_PutMetricData.html#API_PutMetricData_RequestParameters
  // and https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
publishingMetrics.html
  // for more information about the parameters in this command.
  const command = new PutMetricDataCommand({
    MetricData: [
      {
        MetricName: "PAGES_VISITED",
        Dimensions: [
          {
            Name: "UNIQUE_PAGES",
            Value: "URLS",
          },
```

```
        ],
        Unit: "None",
        Value: 1.0,
      },
    ],
    Namespace: "SITE/TRAFFIC",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Creare il client in un modulo separato ed esportarlo.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutMetricDataconsulta AWS SDK for JavaScript API Reference.

SDK per JavaScript (v2)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });
```

```
// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

// Create parameters JSON for putMetricData
var params = {
  MetricData: [
    {
      MetricName: "PAGES_VISITED",
      Dimensions: [
        {
          Name: "UNIQUE_PAGES",
          Value: "URLS",
        },
      ],
      Unit: "None",
      Value: 1.0,
    },
  ],
  Namespace: "SITE/TRAFFIC",
};

cw.putMetricData(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", JSON.stringify(data));
  }
});
```

- Per ulteriori informazioni, consulta la Guida per sviluppatori di AWS SDK for JavaScript.
- Per i dettagli sull'API, PutMetricDataconsulta AWS SDK for JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric $customMetricName")
    }
}
```

- Per i dettagli sull'API, PutMetricDataconsulta AWS SDK for Kotlin API reference.

PowerShell

Strumenti per PowerShell

Esempio 1: crea un nuovo MetricDatum oggetto e lo scrive su Amazon Web Services CloudWatch Metrics.

```
### Create a MetricDatum .NET object
$Metric = New-Object -TypeName Amazon.CloudWatch.Model.MetricDatum
$Metric.Timestamp = [DateTime]::UtcNow
$Metric.MetricName = 'CPU'
$Metric.Value = 50

### Write the metric data to the CloudWatch service
Write-CWMetricData -Namespace instance1 -MetricData $Metric
```

- Per i dettagli sull'API, consulta AWS Tools for PowerShell Cmdlet PutMetricDataReference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data(self, namespace, name, value, unit):
        """
```

```
        Sends a single data value to CloudWatch for a metric. This metric is
 given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
 name)
            raise
```

Inserisci un set di dati in una CloudWatch metrica.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
```

```
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise
```

- Per i dettagli sull'API, consulta PutMetricData AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require "aws-sdk-cloudwatch"
```

```
# Adds a datapoint to a metric in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric to add the
#   datapoint to.
# @param metric_name [String] The name of the metric to add the datapoint to.
# @param dimension_name [String] The name of the dimension to add the
#   datapoint to.
# @param dimension_value [String] The value of the dimension to add the
#   datapoint to.
# @param metric_value [Float] The value of the datapoint.
# @param metric_unit [String] The unit of measurement for the datapoint.
# @return [Boolean]
# @example
#   exit 1 unless datapoint_added_to_metric?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC',
#     'UniqueVisitors',
#     'SiteName',
#     'example.com',
#     5_885.0,
#     'Count'
#   )
def datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  metric_name,
  dimension_name,
  dimension_value,
  metric_value,
  metric_unit
)
  cloudwatch_client.put_metric_data(
    namespace: metric_namespace,
    metric_data: [
      {
        metric_name: metric_name,
        dimensions: [
          {
            name: dimension_name,
            value: dimension_value
          }
```

```
        ],
        value: metric_value,
        unit: metric_unit
      }
    ]
  )
  puts "Added data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}'."
  return true
rescue StandardError => e
  puts "Error adding data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}': #{e.message}"
  return false
end
```

- Per i dettagli sull'API, PutMetricDataconsulta AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo con un SDK CloudWatch AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per l' CloudWatch utilizzo AWS degli SDK

I seguenti esempi di codice mostrano come implementare scenari comuni CloudWatch con gli AWS SDK. Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno. CloudWatch Ogni scenario include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire il codice.

Esempi

- Inizia a utilizzare gli CloudWatch allarmi utilizzando un SDK AWS
- Inizia a usare CloudWatch metriche, dashboard e allarmi utilizzando un SDK AWS
- Gestisci CloudWatch metriche e allarmi utilizzando un SDK AWS

Inizia a utilizzare gli CloudWatch allarmi utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Crea un allarme.

- Disattivare le operazioni di allarme.
- Descrivere un allarme.
- Eliminare un allarme.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
DATA lt_alarmnames TYPE /aws1/cl_cwtalarmnames_w=>tt_alarmnames.
DATA lo_alarmname TYPE REF TO /aws1/cl_cwtalarmnames_w.

"Create an alarm"
TRY.
    lo_cwt->putmetricalarm(
      iv_alarmname                = iv_alarm_name
      iv_comparisonoperator       = iv_comparison_operator
      iv_evaluationperiods        = iv_evaluation_periods
      iv_metricname               = iv_metric_name
      iv_namespace                = iv_namespace
      iv_statistic                = iv_statistic
      iv_threshold                = iv_threshold
      iv_actionsenabled           = iv_actions_enabled
      iv_alarmdescription         = iv_alarm_description
      iv_unit                     = iv_unit
      iv_period                   = iv_period
      it_dimensions               = it_dimensions
    ).
    MESSAGE 'Alarm created' TYPE 'I'.
  CATCH /aws1/cx_cwtlimitexceededfault.
    MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
ENDTRY.

"Create an ABAP internal table for the created alarm."
CREATE OBJECT lo_alarmname EXPORTING iv_value = iv_alarm_name.
```

```
    INSERT lo_alarmname INTO TABLE lt_alarmnames.

    "Disable alarm actions."
    TRY.
        lo_cwt->disablealarmactions(
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarm actions disabled' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_disablealarm_exception).
        DATA(lv_disablealarm_error) = |"{ lo_disablealarm_exception-
>av_err_code }" - { lo_disablealarm_exception->av_err_msg }|.
        MESSAGE lv_disablealarm_error TYPE 'E'.
    ENDTRY.

    "Describe alarm using the same ABAP internal table."
    TRY.
        oo_result = lo_cwt->describealarms(                      " oo_result is
 returned for testing purpose "
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarms retrieved' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_describealarms_exception).
        DATA(lv_describealarms_error) = |"{ lo_describealarms_exception-
>av_err_code }" - { lo_describealarms_exception->av_err_msg }|.
        MESSAGE lv_describealarms_error TYPE 'E'.
    ENDTRY.

    "Delete alarm."
    TRY.
        lo_cwt->deletealarms(
          it_alarmnames = lt_alarmnames
        ).
        MESSAGE 'Alarms deleted' TYPE 'I'.
      CATCH /aws1/cx_cwtresourcenotfound .
        MESSAGE 'Resource being access is not found.' TYPE 'E'.
    ENDTRY.
```

- Per informazioni dettagliate sulle API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per SAP ABAP.
  - DeleteAlarms
  - DescribeAlarms

- DisableAlarmActions
- PutMetricAlarm

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo con un SDK CloudWatch AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Inizia a usare CloudWatch metriche, dashboard e allarmi utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Elenca i CloudWatch namespace e le metriche.
- Ottieni le statistiche per un parametro e per la fatturazione stimata.
- Crea e aggiorna un pannello di controllo.
- Crea e aggiungi i dati a un parametro.
- Crea e attiva un allarme, quindi visualizza la cronologia degli allarmi.
- Aggiungi un rilevatore di anomalie.
- Acquisisci uno schema di parametri, quindi elimina le risorse.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
public class CloudWatchScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.
```

```
    To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
    https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

    This .NET example performs the following tasks:
        1. List and select a CloudWatch namespace.
        2. List and select a CloudWatch metric.
        3. Get statistics for a CloudWatch metric.
        4. Get estimated billing statistics for the last week.
        5. Create a new CloudWatch dashboard with two metrics.
        6. List current CloudWatch dashboards.
        7. Create a CloudWatch custom metric and add metric data.
        8. Add the custom metric to the dashboard.
        9. Create a CloudWatch alarm for the custom metric.
       10. Describe current CloudWatch alarms.
       11. Get recent data for the custom metric.
       12. Add data to the custom metric to trigger the alarm.
       13. Wait for an alarm state.
       14. Get history for the CloudWatch alarm.
       15. Add an anomaly detector.
       16. Describe current anomaly detectors.
       17. Get and display a metric image.
       18. Clean up resources.
    */

    private static ILogger logger = null!;
    private static CloudWatchWrapper _cloudWatchWrapper = null!;
    private static IConfiguration _configuration = null!;
    private static readonly List<string> _statTypes = new List<string>
 { "SampleCount", "Average", "Sum", "Minimum", "Maximum" };
    private static SingleMetricAnomalyDetector? anomalyDetector = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the Amazon service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
```

```
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonCloudWatch>()
            .AddTransient<CloudWatchWrapper>()
        )
        .Build();

        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally, load local settings.
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<CloudWatchScenario>();

        _cloudWatchWrapper =
host.Services.GetRequiredService<CloudWatchWrapper>();

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Amazon CloudWatch example scenario.");
        Console.WriteLine(new string('-', 80));

        try
        {
            var selectedNamespace = await SelectNamespace();
            var selectedMetric = await SelectMetric(selectedNamespace);
            await GetAndDisplayMetricStatistics(selectedNamespace,
selectedMetric);
            await GetAndDisplayEstimatedBilling();
            await CreateDashboardWithMetrics();
            await ListDashboards();
            await CreateNewCustomMetric();
            await AddMetricToDashboard();
            await CreateMetricAlarm();
            await DescribeAlarms();
            await GetCustomMetricData();
            await AddMetricDataForAlarm();
            await CheckForMetricAlarm();
            await GetAlarmHistory();
            anomalyDetector = await AddAnomalyDetector();
            await DescribeAnomalyDetectors();
            await GetAndOpenMetricImage();
            await CleanupResources();
```

```
        }
        catch (Exception ex)
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
            await CleanupResources();
        }

    }

    /// <summary>
    /// Select a namespace.
    /// </summary>
    /// <returns>The selected namespace.</returns>
    private static async Task<string> SelectNamespace()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"1. Select a CloudWatch Namespace from a list of
Namespaces.");
        var metrics = await _cloudWatchWrapper.ListMetrics();
        // Get a distinct list of namespaces.
        var namespaces = metrics.Select(m => m.Namespace).Distinct().ToList();
        for (int i = 0; i < namespaces.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {namespaces[i]}");
        }

        var namespaceChoiceNumber = 0;
        while (namespaceChoiceNumber < 1 || namespaceChoiceNumber >
namespaces.Count)
        {
            Console.WriteLine(
                "Select a namespace by entering a number from the preceding
list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out namespaceChoiceNumber);
        }

        var selectedNamespace = namespaces[namespaceChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedNamespace;
    }
```

```
    /// <summary>
    /// Select a metric from a namespace.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <returns>The metric name.</returns>
    private static async Task<Metric> SelectMetric(string metricNamespace)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"2. Select a CloudWatch metric from a namespace.");

        var namespaceMetrics = await
_cloudWatchWrapper.ListMetrics(metricNamespace);

        for (int i = 0; i < namespaceMetrics.Count && i < 15; i++)
        {
            var dimensionsWithValues = namespaceMetrics[i].Dimensions
                .Where(d => !string.Equals("None", d.Value));
            Console.WriteLine($"\t{i + 1}. {namespaceMetrics[i].MetricName} " +
                              $"{string.Join(", :", dimensionsWithValues.Select(d
=> d.Value))}");
        }

        var metricChoiceNumber = 0;
        while (metricChoiceNumber < 1 || metricChoiceNumber >
namespaceMetrics.Count)
        {
            Console.WriteLine(
                "Select a metric by entering a number from the preceding list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out metricChoiceNumber);
        }

        var selectedMetric = namespaceMetrics[metricChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedMetric;
    }

    /// <summary>
    /// Get and display metric statistics for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
```

```
   /// <returns>Async task.</returns>
   private static async Task GetAndDisplayMetricStatistics(string
metricNamespace, Metric metric)
   {
       Console.WriteLine(new string('-', 80));
       Console.WriteLine($"3. Get CloudWatch metric statistics for the last
day.");

       for (int i = 0; i < _statTypes.Count; i++)
       {
           Console.WriteLine($"\t{i + 1}. {_statTypes[i]}");
       }

       var statisticChoiceNumber = 0;
       while (statisticChoiceNumber < 1 || statisticChoiceNumber >
_statTypes.Count)
       {
           Console.WriteLine(
               "Select a metric statistic by entering a number from the
preceding list:");
           var choice = Console.ReadLine();
           Int32.TryParse(choice, out statisticChoiceNumber);
       }

       var selectedStatistic = _statTypes[statisticChoiceNumber - 1];
       var statisticsList = new List<string> { selectedStatistic };

       var metricStatistics = await
_cloudWatchWrapper.GetMetricStatistics(metricNamespace, metric.MetricName,
statisticsList, metric.Dimensions, 1, 60);

       if (!metricStatistics.Any())
       {
           Console.WriteLine($"No {selectedStatistic} statistics found for
{metric} in namespace {metricNamespace}.");
       }

       metricStatistics = metricStatistics.OrderBy(s => s.Timestamp).ToList();
       for (int i = 0; i < metricStatistics.Count && i < 10; i++)
       {
           var metricStat = metricStatistics[i];
           var statValue =
metricStat.GetType().GetProperty(selectedStatistic)!.GetValue(metricStat, null);
```

```
            Console.WriteLine($"\t{i + 1}. Timestamp
{metricStatistics[i].Timestamp:G} {selectedStatistic}: {statValue}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get and display estimated billing statistics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayEstimatedBilling()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"4. Get CloudWatch estimated billing for the last
week.");

        var billingStatistics = await SetupBillingStatistics();

        for (int i = 0; i < billingStatistics.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. Timestamp
{billingStatistics[i].Timestamp:G} : {billingStatistics[i].Maximum}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
```

```
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();

        return billingStatistics;
    }

    /// <summary>
    /// Create a dashboard with metrics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CreateDashboardWithMetrics()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"5. Create a new CloudWatch dashboard with metrics.");
        var dashboardName = _configuration["dashboardName"];
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);
        var newDashboardString = JsonSerializer.Serialize(
            newDashboard,
            new JsonSerializerOptions
            {
                DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull
            });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} was created.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List dashboards.
    /// </summary>
```

```
    /// <returns>Async task.</returns>
    private static async Task ListDashboards()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"6. List the CloudWatch dashboards in the current
account.");

        var dashboards = await _cloudWatchWrapper.ListDashboards();

        for (int i = 0; i < dashboards.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {dashboards[i].DashboardName}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create and add data for a new custom metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateNewCustomMetric()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"7. Create and add data for a new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var customData = await PutRandomMetricData(customMetricName,
customMetricNamespace);

        var valuesString = string.Join(',', customData.Select(d => d.Value));
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");

        Console.WriteLine(new string('-', 80));
    }



    /// <summary>
    /// Add some metric data using a call to a wrapper class.
    /// </summary>
    /// <param name="customMetricName">The metric name.</param>
```

```
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <returns></returns>
    private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
        string customMetricNamespace)
    {
        List<MetricDatum> customData = new List<MetricDatum>();
        Random rnd = new Random();

        // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
        var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
        for (int i = 0; i < 10; i++)
        {
            var metricValue = rnd.Next(0, 100);
            customData.Add(
                new MetricDatum
                {
                    MetricName = customMetricName,
                    Value = metricValue,
                    TimestampUtc = utcNowMinus15.AddMinutes(i)
                }
            );
        }

        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
        return customData;
    }

    /// <summary>
    /// Add the custom metric to the dashboard.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricToDashboard()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"8. Add the new custom metric to the dashboard.");

        var dashboardName = _configuration["dashboardName"];

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
```

```
        var validationMessages = await SetupDashboard(customMetricNamespace,
customMetricName, dashboardName);

        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} updated with metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }


    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
```

```
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Create a CloudWatch alarm for the new metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"9. Create a CloudWatch alarm for the new metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var alarmName = _configuration["exampleAlarmName"];
        var accountId = _configuration["accountId"];
        var region = _configuration["region"];
        var emailTopic = _configuration["emailTopic"];
        var alarmActions = new List<string>();

        if (GetYesNoResponse(
                $"\tAdd an email action for topic {emailTopic} to alarm
{alarmName}? (y/n)"))
        {
```

```
            _cloudWatchWrapper.AddEmailAlarmAction(accountId, region, emailTopic,
alarmActions);
        }

        await _cloudWatchWrapper.PutMetricEmailAlarm(
            "Example metric alarm",
            alarmName,
            ComparisonOperator.GreaterThanOrEqualToThreshold,
            customMetricName,
            customMetricNamespace,
            100,
            alarmActions);

        Console.WriteLine($"\tAlarm {alarmName} added for metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Describe Alarms.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAlarms()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"10. Describe CloudWatch alarms in the current
account.");

        var alarms = await _cloudWatchWrapper.DescribeAlarms();
        alarms = alarms.OrderByDescending(a => a.StateUpdatedTimestamp).ToList();

        for (int i = 0; i < alarms.Count && i < 10; i++)
        {
            var alarm = alarms[i];
            Console.WriteLine($"\t{i + 1}. {alarm.AlarmName}");
            Console.WriteLine($"\tState: {alarm.StateValue} for
{alarm.MetricName} {alarm.ComparisonOperator} {alarm.Threshold}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get the recent data for the metric.
```

```
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetCustomMetricData()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"11. Get current data for new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var accountId = _configuration["accountId"];

        var query = new List<MetricDataQuery>
        {
            new MetricDataQuery
            {
                AccountId = accountId,
                Id = "m1",
                Label = "Custom Metric Data",
                MetricStat = new MetricStat
                {
                    Metric = new Metric
                    {
                        MetricName = customMetricName,
                        Namespace = customMetricNamespace,
                    },
                    Period = 1,
                    Stat = "Maximum"
                }
            }
        };

        var metricData = await _cloudWatchWrapper.GetMetricData(
            20,
            true,
            DateTime.UtcNow.AddMinutes(1),
            20,
            query);

        for (int i = 0; i < metricData.Count; i++)
        {
            for (int j = 0; j < metricData[i].Values.Count; j++)
            {
                Console.WriteLine(
```

```
                    $"\tTimestamp {metricData[i].Timestamps[j]:G} Value:
{metricData[i].Values[j]}");
            }
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add metric data to trigger an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricDataForAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"12. Add metric data to the custom metric to trigger
an alarm.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var nowUtc = DateTime.UtcNow;
        List<MetricDatum> customData = new List<MetricDatum>
        {
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-2)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-1)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc
            }
        };
        var valuesString = string.Join(',', customData.Select(d => d.Value));
```

```
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");
        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Check for a metric alarm using the DescribeAlarmsForMetric action.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CheckForMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"13. Checking for an alarm state.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var hasAlarm = false;
        var retries = 10;
        while (!hasAlarm && retries > 0)
        {
            var alarms = await
_cloudWatchWrapper.DescribeAlarmsForMetric(customMetricNamespace,
customMetricName);
            hasAlarm = alarms.Any(a => a.StateValue == StateValue.ALARM);
            retries--;
            Thread.Sleep(20000);
        }

        Console.WriteLine(hasAlarm
            ? $"\tAlarm state found for {customMetricName}."
            : $"\tNo Alarm state found for {customMetricName} after 10
retries.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get history for an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAlarmHistory()
```

```
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"14. Get alarm history.");

        var exampleAlarmName = _configuration["exampleAlarmName"];

        var alarmHistory = await
_cloudWatchWrapper.DescribeAlarmHistory(exampleAlarmName, 2);

        for (int i = 0; i < alarmHistory.Count; i++)
        {
            var history = alarmHistory[i];
            Console.WriteLine($"\t{i + 1}. {history.HistorySummary}, time
{history.Timestamp:g}");
        }
        if (!alarmHistory.Any())
        {
            Console.WriteLine($"\tNo alarm history data found for
{exampleAlarmName}.");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add an anomaly detector.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task<SingleMetricAnomalyDetector> AddAnomalyDetector()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"15. Add an anomaly detector.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detector = new SingleMetricAnomalyDetector
        {
            MetricName = customMetricName,
            Namespace = customMetricNamespace,
            Stat = "Maximum"
        };
        await _cloudWatchWrapper.PutAnomalyDetector(detector);
```

```
        Console.WriteLine($"\tAdded anomaly detector for metric
{customMetricName}.");

        Console.WriteLine(new string('-', 80));
        return detector;
    }

    /// <summary>
    /// Describe anomaly detectors.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAnomalyDetectors()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"16. Describe anomaly detectors in the current
account.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detectors = await
_cloudWatchWrapper.DescribeAnomalyDetectors(customMetricNamespace,
customMetricName);

        for (int i = 0; i < detectors.Count; i++)
        {
            var detector = detectors[i];
            Console.WriteLine($"\t{i + 1}.
{detector.SingleMetricAnomalyDetector.MetricName}, state
{detector.StateValue}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Fetch and open a metrics image for a CloudWatch metric and namespace.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAndOpenMetricImage()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("17. Get a metric image from CloudWatch.");
```

```
        Console.WriteLine($"\tGetting Image data for custom metric.");
        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var memoryStream = await
_cloudWatchWrapper.GetTimeSeriesMetricImage(customMetricNamespace,
customMetricName, "Maximum", 10);
        var file = _cloudWatchWrapper.SaveMetricImage(memoryStream,
"MetricImages");

        ProcessStartInfo info = new ProcessStartInfo();

        Console.WriteLine($"\tFile saved as {Path.GetFileName(file)}.");
        Console.WriteLine($"\tPress enter to open the image.");
        Console.ReadLine();
        info.FileName = Path.Combine("ms-photos://", file);
        info.UseShellExecute = true;
        info.CreateNoWindow = true;
        info.Verb = string.Empty;

        Process.Start(info);

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Clean up created resources.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CleanupResources()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"18. Clean up resources.");

        var dashboardName = _configuration["dashboardName"];
        if (GetYesNoResponse($"\tDelete dashboard {dashboardName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting dashboard.");
            var dashboardList = new List<string> { dashboardName };
            await _cloudWatchWrapper.DeleteDashboards(dashboardList);
        }
```

```
        var alarmName = _configuration["exampleAlarmName"];
        if (GetYesNoResponse($"\tDelete alarm {alarmName}? (y/n)"))
        {
            Console.WriteLine($"\tCleaning up alarms.");
            var alarms = new List<string> { alarmName };
            await _cloudWatchWrapper.DeleteAlarms(alarms);
        }

        if (GetYesNoResponse($"\tDelete anomaly detector? (y/n)") &&
 anomalyDetector != null)
        {
            Console.WriteLine($"\tCleaning up anomaly detector.");

            await _cloudWatchWrapper.DeleteAnomalyDetector(
                anomalyDetector);
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null &&
                       ynResponse.Equals("y",
                           StringComparison.InvariantCultureIgnoreCase);
        return response;
    }
}
```

Metodi wrapper utilizzati dallo scenario per CloudWatch le azioni.

```
/// <summary>
/// Wrapper class for Amazon CloudWatch methods.
/// </summary>
```

```
public class CloudWatchWrapper
{
    private readonly IAmazonCloudWatch _amazonCloudWatch;
    private readonly ILogger<CloudWatchWrapper> _logger;

    /// <summary>
    /// Constructor for the CloudWatch wrapper.
    /// </summary>
    /// <param name="amazonCloudWatch">The injected CloudWatch client.</param>
    /// <param name="logger">The injected logger for the wrapper.</param>
    public CloudWatchWrapper(IAmazonCloudWatch amazonCloudWatch,
 ILogger<CloudWatchWrapper> logger)

    {
        _logger = logger;
        _amazonCloudWatch = amazonCloudWatch;
    }

    /// <summary>
    /// List metrics available, optionally within a namespace.
    /// </summary>
    /// <param name="metricNamespace">Optional CloudWatch namespace to use when
 listing metrics.</param>
    /// <param name="filter">Optional dimension filter.</param>
    /// <param name="metricName">Optional metric name filter.</param>
    /// <returns>The list of metrics.</returns>
    public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
 DimensionFilter? filter = null, string? metricName = null)
    {
        var results = new List<Metric>();
        var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
            new ListMetricsRequest
            {
                Namespace = metricNamespace,
                Dimensions = filter != null ? new List<DimensionFilter>
 { filter } : null,
                MetricName = metricName
            });
        // Get the entire list using the paginator.
        await foreach (var metric in paginateMetrics.Metrics)
        {
            results.Add(metric);
        }
```

```
        return results;
    }

    /// <summary>
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
 metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
 int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
 dashboardName,
        string dashboardBody)
    {
```

```
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
 was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }


    /// <summary>
    /// Get information on a dashboard.
    /// </summary>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A JSON object with dashboard information.</returns>
    public async Task<string> GetDashboard(string dashboardName)
    {
        var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
            new GetDashboardRequest()
            {
                DashboardName = dashboardName
            });

        return dashboardResponse.DashboardBody;
    }


    /// <summary>
    /// Get a list of dashboards.
    /// </summary>
    /// <returns>A list of DashboardEntry objects.</returns>
    public async Task<List<DashboardEntry>> ListDashboards()
    {
        var results = new List<DashboardEntry>();
        var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
            new ListDashboardsRequest());
        // Get the entire list using the paginator.
        await foreach (var data in paginateDashboards.DashboardEntries)
        {
            results.Add(data);
```

```
        }

        return results;
    }

    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
```

```
                { new() { metricNamespace, metric, new { stat } } }
        };

        var metricImageWidgetString =
 JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }

    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
```

```
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
 ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }

    /// <summary>
    /// Add a metric alarm to send an email when the metric passes a threshold.
    /// </summary>
    /// <param name="alarmDescription">A description of the alarm.</param>
    /// <param name="alarmName">The name for the alarm.</param>
    /// <param name="comparison">The type of comparison to use.</param>
    /// <param name="metricName">The name of the metric for the alarm.</param>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="threshold">The threshold value for the alarm.</param>
    /// <param name="alarmActions">Optional actions to execute when in an alarm
 state.</param>
    /// <returns>True if successful.</returns>
```

```
   public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
alarmName, ComparisonOperator comparison,
       string metricName, string metricNamespace, double threshold, List<string>
alarmActions = null!)
   {
       try
       {
           var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
               new PutMetricAlarmRequest()
               {
                   AlarmActions = alarmActions,
                   AlarmDescription = alarmDescription,
                   AlarmName = alarmName,
                   ComparisonOperator = comparison,
                   Threshold = threshold,
                   Namespace = metricNamespace,
                   MetricName = metricName,
                   EvaluationPeriods = 1,
                   Period = 10,
                   Statistic = new Statistic("Maximum"),
                   DatapointsToAlarm = 1,
                   TreatMissingData = "ignore"
               });
           return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
       }
       catch (LimitExceededException lex)
       {
           _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
has already been reached.");
       }

       return false;
   }

   /// <summary>
   /// Add specific email actions to a list of action strings for a CloudWatch
alarm.
   /// </summary>
   /// <param name="accountId">The AccountId for the alarm.</param>
   /// <param name="region">The region for the alarm.</param>
   /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
topic for the alarm email.</param>
```

```
    /// <param name="alarmActions">Optional list of existing alarm actions to
 append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }

    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
 null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
        var paginatedDescribeAlarms =
 _amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }

    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
 metricNamespace, string metricName)
```

```
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }

    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }

    /// <summary>
    /// Delete a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAlarms(List<string> alarmNames)
```

```
    {
        var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
            new DeleteAlarmsRequest()
            {
                AlarmNames = alarmNames
            });

        return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

```
    /// <summary>
    /// Add an anomaly detector for a single metric.
    /// </summary>
    /// <param name="anomalyDetector">A single metric anomaly detector.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
    {
        var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
            new PutAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Describe anomaly detectors for a metric and namespace.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The metric of the anomaly detectors.</param>
    /// <returns>The list of detectors.</returns>
    public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
    {
        List<AnomalyDetector> detectors = new List<AnomalyDetector>();
        var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
            new DescribeAnomalyDetectorsRequest()
            {
                MetricName = metricName,
                Namespace = metricNamespace
            });

        await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
        {
            detectors.Add(data);
        }

        return detectors;
    }
```

```
    /// <summary>
    /// Delete a single metric anomaly detector.
    /// </summary>
    /// <param name="anomalyDetector">The anomaly detector to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
 anomalyDetector)
    {
        var deleteAnomalyDetectorResponse = await
 _amazonCloudWatch.DeleteAnomalyDetectorAsync(
            new DeleteAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
 _amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - DeleteAlarms
  - DeleteAnomalyDetector

- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import com.fasterxml.jackson.core.JsonFactory;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.ObjectMapper;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.AlarmHistoryItem;
import software.amazon.awssdk.services.cloudwatch.model.AlarmType;
import software.amazon.awssdk.services.cloudwatch.model.AnomalyDetector;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ComparisonOperator;
```

```
import
 software.amazon.awssdk.services.cloudwatch.model.DashboardValidationMessage;
import software.amazon.awssdk.services.cloudwatch.model.Datapoint;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DeleteAnomalyDetectorRequest;
import software.amazon.awssdk.services.cloudwatch.model.DeleteDashboardsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricResponse;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsRequest;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Dimension;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageResponse;
import software.amazon.awssdk.services.cloudwatch.model.HistoryItemType;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;
import software.amazon.awssdk.services.cloudwatch.model.MetricAlarm;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataQuery;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataResult;
import software.amazon.awssdk.services.cloudwatch.model.MetricDatum;
import software.amazon.awssdk.services.cloudwatch.model.MetricStat;
import
 software.amazon.awssdk.services.cloudwatch.model.PutAnomalyDetectorRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardRequest;
```

```
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardResponse;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricAlarmRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.ScanBy;
import
 software.amazon.awssdk.services.cloudwatch.model.SingleMetricAnomalyDetector;
import software.amazon.awssdk.services.cloudwatch.model.StandardUnit;
import software.amazon.awssdk.services.cloudwatch.model.Statistic;
import
 software.amazon.awssdk.services.cloudwatch.paginators.ListDashboardsIterable;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.ZoneOffset;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * To enable billing metrics and statistics for this example, make sure billing
 * alerts are enabled for your account:
 * https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics
 *
 * This Java code example performs the following tasks:
 *
 * 1. List available namespaces from Amazon CloudWatch.
```

```
 * 2. List available metrics within the selected Namespace.
 * 3. Get statistics for the selected metric over the last day.
 * 4. Get CloudWatch estimated billing for the last week.
 * 5. Create a new CloudWatch dashboard with metrics.
 * 6. List dashboards using a paginator.
 * 7. Create a new custom metric by adding data for it.
 * 8. Add the custom metric to the dashboard.
 * 9. Create an alarm for the custom metric.
 * 10. Describe current alarms.
 * 11. Get current data for the new custom metric.
 * 12. Push data into the custom metric to trigger the alarm.
 * 13. Check the alarm state using the action DescribeAlarmsForMetric.
 * 14. Get alarm history for the new alarm.
 * 15. Add an anomaly detector for the custom metric.
 * 16. Describe current anomaly detectors.
 * 17. Get a metric image for the custom metric.
 * 18. Clean up the Amazon CloudWatch resources.
 */
public class CloudWatchScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws IOException {
        final String usage = """

                Usage:
                  <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage> \s

                Where:
                  myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)\s
                  costDateWeek - The start date to use to get AWS/Billinget
 statistics. (For example, 2023-01-11T18:35:24.00Z.)\s
                  dashboardName - The name of the dashboard to create.\s
                  dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)\s
                  dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)\s
                  settings - The location of a JSON file from which various
 values are read. (See Readme file.)\s
                  metricImage - The location of a BMP file that is used to create
 a graph.\s
                """;
```

```
        if (args.length != 7) {
            System.out.println(usage);
            System.exit(1);
        }

        Region region = Region.US_EAST_1;
        String myDate = args[0];
        String costDateWeek = args[1];
        String dashboardName = args[2];
        String dashboardJson = args[3];
        String dashboardAdd = args[4];
        String settings = args[5];
        String metricImage = args[6];

        Double dataPoint = Double.parseDouble("10.0");
        Scanner sc = new Scanner(System.in);
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon CloudWatch example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "1. List at least five available unique namespaces from Amazon
CloudWatch. Select one from the list.");
        ArrayList<String> list = listNameSpaces(cw);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + list.get(z));
        }

        String selectedNamespace = "";
        String selectedMetrics = "";
        int num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedNamespace = list.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
```

```
        }
        System.out.println("You selected " + selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. List available metrics within the selected
namespace and select one from the list.");
        ArrayList<String> metList = listMets(cw, selectedNamespace);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + metList.get(z));
        }
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedMetrics = metList.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + selectedMetrics);
        Dimension myDimension = getSpecificMet(cw, selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get statistics for the selected metric over the
last day.");
        String metricOption = "";
        ArrayList<String> statTypes = new ArrayList<>();
        statTypes.add("SampleCount");
        statTypes.add("Average");
        statTypes.add("Sum");
        statTypes.add("Minimum");
        statTypes.add("Maximum");

        for (int t = 0; t < 5; t++) {
            System.out.println("    " + (t + 1) + ". " + statTypes.get(t));
        }
        System.out.println("Select a metric statistic by entering a number from
the preceding list:");
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            metricOption = statTypes.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
```

```
            System.exit(1);
        }
        System.out.println("You selected " + metricOption);
        getAndDisplayMetricStatistics(cw, selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get CloudWatch estimated billing for the last
week.");
        getMetricStatistics(cw, costDateWeek);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Create a new CloudWatch dashboard with metrics.");
        createDashboardWithMetrics(cw, dashboardName, dashboardJson);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. List dashboards using a paginator.");
        listDashboards(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create a new custom metric by adding data to
it.");
        createNewCustomMetric(cw, dataPoint);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Add an additional metric to the dashboard.");
        addMetricToDashboard(cw, dashboardAdd, dashboardName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9. Create an alarm for the custom metric.");
        String alarmName = createAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Describe ten current alarms.");
        describeAlarms(cw);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("11. Get current data for new custom metric.");
        getCustomMetricData(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Push data into the custom metric to trigger the
alarm.");
        addMetricDataForAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Check the alarm state using the action
DescribeAlarmsForMetric.");
        checkForMetricAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Get alarm history for the new alarm.");
        getAlarmHistory(cw, settings, myDate);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("15. Add an anomaly detector for the custom metric.");
        addAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("16. Describe current anomaly detectors.");
        describeAnomalyDetectors(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("17. Get a metric image for the custom metric.");
        getAndOpenMetricImage(cw, metricImage);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("18. Clean up the Amazon CloudWatch resources.");
        deleteDashboard(cw, dashboardName);
        deleteCWAlarm(cw, alarmName);
        deleteAnomalyDetector(cw, settings);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("The Amazon CloudWatch example scenario is
complete.");
        System.out.println(DASHES);
        cw.close();
    }

    public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.deleteAnomalyDetector(request);
            System.out.println("Successfully deleted the Anomaly Detector.");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
```

```
        try {
            DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.deleteAlarms(request);
            System.out.println("Successfully deleted alarm " + alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
        try {
            DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                    .dashboardNames(dashboardName)
                    .build();
            cw.deleteDashboards(dashboardsRequest);
            System.out.println(dashboardName + " was successfully deleted.");

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
        System.out.println("Getting Image data for custom metric.");
        try {
            String myJSON = "{\n" +
                    "  \"title\": \"Example Metric Graph\",\n" +
                    "  \"view\": \"timeSeries\",\n" +
                    "  \"stacked \": false,\n" +
                    "  \"period\": 10,\n" +
                    "  \"width\": 1400,\n" +
                    "  \"height\": 600,\n" +
                    "  \"metrics\": [\n" +
                    "    [\n" +
                    "      \"AWS/Billing\",\n" +
```

```
                    "      \"EstimatedCharges\",\n" +
                    "      \"Currency\",\n" +
                    "      \"USD\"\n" +
                    "    ]\n" +
                    "  ]\n" +
                    "}";

            GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                    .metricWidget(myJSON)
                    .build();

            GetMetricWidgetImageResponse response =
cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                    .maxResults(10)
                    .metricName(customMetricName)
```

```
                    .namespace(customMetricNamespace)
                    .build();

            DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
            List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
            for (AnomalyDetector detector : anomalyDetectorList) {
                System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
                System.out.println("State: " + detector.stateValue());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.putAnomalyDetector(anomalyDetectorRequest);
```

```
            System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String alarmName = rootNode.findValue("exampleAlarmName").asText();

            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();
            DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                    .startDate(start)
                    .endDate(endDate)
                    .alarmName(alarmName)
                    .historyItemType(HistoryItemType.ACTION)
                    .build();

            DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
            List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
            if (historyItems.isEmpty()) {
                System.out.println("No alarm history data found for " + alarmName
+ ".");
            } else {
                for (AlarmHistoryItem item : historyItems) {
                    System.out.println("History summary: " +
item.historySummary());
                    System.out.println("Time stamp: " + item.timestamp());
                }
            }

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            boolean hasAlarm = false;
            int retries = 10;

            DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            while (!hasAlarm && retries > 0) {
                DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
                hasAlarm = response.hasMetricAlarms();
                retries--;
                Thread.sleep(20000);
                System.out.println(".");
            }
            if (!hasAlarm)
                System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
            else
                System.out.println("Alarm state found for " + customMetricName +
".");

        } catch (CloudWatchException | IOException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
```

```
        }
    }

    public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1001.00)
                    .timestamp(instant)
                    .build();

            MetricDatum datum2 = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1002.00)
                    .timestamp(instant)
                    .build();

            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();
```

```
            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
customMetricName);

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set the date.
            Instant nowDate = Instant.now();

            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
```

```
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
```

```
                System.out.println("Alarm description: " +
alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static String createAlarm(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();
```

```
            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }

    public static void addMetricToDashboard(CloudWatchClient cw, String fileName,
String dashboardName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully updated.");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void createNewCustomMetric(CloudWatchClient cw, Double
dataPoint) {
        try {
            Dimension dimension = Dimension.builder()
                    .name("UNIQUE_PAGES")
                    .value("URLS")
                    .build();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName("PAGES_VISITED")
```

```
                    .unit(StandardUnit.NONE)
                    .value(dataPoint)
                    .timestamp(instant)
                    .dimensions(dimension)
                    .build();

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace("SITE/TRAFFIC")
                    .metricData(datum)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric
PAGES_VISITED");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
```

```
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            PutDashboardResponse response = cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully created.");
            List<DashboardValidationMessage> messages =
response.dashboardValidationMessages();
            if (messages.isEmpty()) {
                System.out.println("There are no messages in the new Dashboard");
            } else {
                for (DashboardValidationMessage message : messages) {
                    System.out.println("Message is: " + message.message());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static String readFileAsString(String file) throws IOException {
        return new String(Files.readAllBytes(Paths.get(file)));
    }

    public static void getMetricStatistics(CloudWatchClient cw, String
costDateWeek) {
        try {
            Instant start = Instant.parse(costDateWeek);
            Instant endDate = Instant.now();
            Dimension dimension = Dimension.builder()
                    .name("Currency")
                    .value("USD")
                    .build();

            List<Dimension> dimensionList = new ArrayList<>();
            dimensionList.add(dimension);
            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .metricName("EstimatedCharges")
                    .namespace("AWS/Billing")
                    .dimensions(dimensionList)
                    .statistics(Statistic.MAXIMUM)
                    .startTime(start)
```

```
                    .endTime(endDate)
                    .period(86400)
                    .build();

            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
            String metricOption, String date, Dimension myDimension) {
        try {
            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();

            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .endTime(endDate)
                    .startTime(start)
                    .dimensions(myDimension)
                    .metricName(metVal)
                    .namespace(nameSpace)
                    .period(86400)
                    .statistics(Statistic.fromValue(metricOption))
                    .build();

            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
```

```
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static Dimension getSpecificMet(CloudWatchClient cw, String namespace)
{
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsResponse response = cw.listMetrics(request);
            List<Metric> myList = response.metrics();
            Metric metric = myList.get(0);
            return metric.dimensions().get(0);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listMets(CloudWatchClient cw, String
namespace) {
        try {
            ArrayList<String> metList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
```

```
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> metList.add(metrics.metricName()));

            return metList;

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listNameSpaces(CloudWatchClient cw) {
        try {
            ArrayList<String> nameSpaceList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> {
                        String data = metrics.namespace();
                        if (!nameSpaceList.contains(data)) {
                            nameSpaceList.add(data);
                        }
                    });

            return nameSpaceList;
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - DeleteAlarms

- DeleteAnomalyDetector
- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/**
 Before running this Kotlin code example, set up your development environment,
 including your credentials.

 For more information, see the following documentation topic:
 https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

 To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
 https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics
```

```
 This Kotlin code example performs the following tasks:

 1. List available namespaces from Amazon CloudWatch. Select a namespace from the
 list.
 2. List available metrics within the selected namespace.
 3. Get statistics for the selected metric over the last day.
 4. Get CloudWatch estimated billing for the last week.
 5. Create a new CloudWatch dashboard with metrics.
 6. List dashboards using a paginator.
 7. Create a new custom metric by adding data for it.
 8. Add the custom metric to the dashboard.
 9. Create an alarm for the custom metric.
 10. Describe current alarms.
 11. Get current data for the new custom metric.
 12. Push data into the custom metric to trigger the alarm.
 13. Check the alarm state using the action DescribeAlarmsForMetric.
 14. Get alarm history for the new alarm.
 15. Add an anomaly detector for the custom metric.
 16. Describe current anomaly detectors.
 17. Get a metric image for the custom metric.
 18. Clean up the Amazon CloudWatch resources.
 */

val DASHES: String? = String(CharArray(80)).replace("\u0000", "-")
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage>

        Where:
            myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)
            costDateWeek - The start date to use to get AWS Billing and Cost
 Management statistics. (For example, 2023-01-11T18:35:24.00Z.)
            dashboardName - The name of the dashboard to create.
            dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)
            dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)
            settings - The location of a JSON file from which various values are
 read. (See Readme file.)
```

```
            metricImage - The location of a BMP file that is used to create a
graph.
    """

    if (args.size != 7) {
        println(usage)
        System.exit(1)
    }

    val myDate = args[0]
    val costDateWeek = args[1]
    val dashboardName = args[2]
    val dashboardJson = args[3]
    val dashboardAdd = args[4]
    val settings = args[5]
    var metricImage = args[6]
    val dataPoint = "10.0".toDouble()
    val inOb = Scanner(System.`in`)

    println(DASHES)
    println("Welcome to the Amazon CloudWatch example scenario.")
    println(DASHES)

    println(DASHES)
    println("1. List at least five available unique namespaces from Amazon
CloudWatch. Select a CloudWatch namespace from the list.")
    val list: ArrayList<String> = listNameSpaces()
    for (z in 0..4) {
        println("    ${z + 1}. ${list[z]}")
    }

    var selectedNamespace: String
    var selectedMetrics = ""
    var num = inOb.nextLine().toInt()
    println("You selected $num")

    if (1 <= num && num <= 5) {
        selectedNamespace = list[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $selectedNamespace")
    println(DASHES)
```

```
    println(DASHES)
    println("2. List available metrics within the selected namespace and select
one from the list.")
    val metList = listMets(selectedNamespace)
    for (z in 0..4) {
        println("    ${ z + 1}. ${metList?.get(z)}")
    }
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        selectedMetrics = metList!![num - 1]
    } else {
        println("You did not select a valid option.")
        System.exit(1)
    }
    println("You selected $selectedMetrics")
    val myDimension = getSpecificMet(selectedNamespace)
    if (myDimension == null) {
        println("Error - Dimension is null")
        exitProcess(1)
    }
    println(DASHES)

    println(DASHES)
    println("3. Get statistics for the selected metric over the last day.")
    val metricOption: String
    val statTypes = ArrayList<String>()
    statTypes.add("SampleCount")
    statTypes.add("Average")
    statTypes.add("Sum")
    statTypes.add("Minimum")
    statTypes.add("Maximum")

    for (t in 0..4) {
        println("    ${t + 1}. ${statTypes[t]}")
    }
    println("Select a metric statistic by entering a number from the preceding
list:")
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        metricOption = statTypes[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
```

```
    }
    println("You selected $metricOption")
    getAndDisplayMetricStatistics(selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension)
    println(DASHES)

    println(DASHES)
    println("4. Get CloudWatch estimated billing for the last week.")
    getMetricStatistics(costDateWeek)
    println(DASHES)

    println(DASHES)
    println("5. Create a new CloudWatch dashboard with metrics.")
    createDashboardWithMetrics(dashboardName, dashboardJson)
    println(DASHES)

    println(DASHES)
    println("6. List dashboards using a paginator.")
    listDashboards()
    println(DASHES)

    println(DASHES)
    println("7. Create a new custom metric by adding data to it.")
    createNewCustomMetric(dataPoint)
    println(DASHES)

    println(DASHES)
    println("8. Add an additional metric to the dashboard.")
    addMetricToDashboard(dashboardAdd, dashboardName)
    println(DASHES)

    println(DASHES)
    println("9. Create an alarm for the custom metric.")
    val alarmName: String = createAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("10. Describe 10 current alarms.")
    describeAlarms()
    println(DASHES)

    println(DASHES)
    println("11. Get current data for the new custom metric.")
    getCustomMetricData(settings)
```

```
    println(DASHES)

    println(DASHES)
    println("12. Push data into the custom metric to trigger the alarm.")
    addMetricDataForAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("13. Check the alarm state using the action
 DescribeAlarmsForMetric.")
    checkForMetricAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("14. Get alarm history for the new alarm.")
    getAlarmHistory(settings, myDate)
    println(DASHES)

    println(DASHES)
    println("15. Add an anomaly detector for the custom metric.")
    addAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("16. Describe current anomaly detectors.")
    describeAnomalyDetectors(settings)
    println(DASHES)

    println(DASHES)
    println("17. Get a metric image for the custom metric.")
    getAndOpenMetricImage(metricImage)
    println(DASHES)

    println(DASHES)
    println("18. Clean up the Amazon CloudWatch resources.")
    deleteDashboard(dashboardName)
    deleteAlarm(alarmName)
    deleteAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("The Amazon CloudWatch example scenario is complete.")
    println(DASHES)
}
```

```
suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}

suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}

suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
```

```
}

suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}

suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
```

```
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}

suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}

suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
```

```
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}

suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
```

```
                hasAlarm = true
            }
            retries--
            delay(20000)
            println(".")
        }
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}

suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
```

```
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric $customMetricName")
    }
}

suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }

    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
```

```
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}

suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}

suspend fun createAlarm(fileName: String): String {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode: JsonNode = ObjectMapper().readTree(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val emailTopic = rootNode.findValue("emailTopic").asText()
    val accountId = rootNode.findValue("accountId").asText()
```

```
    val region2 = rootNode.findValue("region").asText()

    // Create a List for alarm actions.
    val alarmActionObs: MutableList<String> = ArrayList()
    alarmActionObs.add("arn:aws:sns:$region2:$accountId:$emailTopic")
    val alarmRequest = PutMetricAlarmRequest {
        alarmActions = alarmActionObs
        alarmDescription = "Example metric alarm"
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanOrEqualToThreshold
        threshold = 100.00
        metricName = customMetricName
        namespace = customMetricNamespace
        evaluationPeriods = 1
        period = 10
        statistic = Statistic.Maximum
        datapointsToAlarm = 1
        treatMissingData = "ignore"
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(alarmRequest)
        println("$alarmNameVal was successfully created!")
        return alarmNameVal
    }
}

suspend fun addMetricToDashboard(fileNameVal: String, dashboardNameVal: String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully updated.")
    }
}

suspend fun createNewCustomMetric(dataPoint: Double) {
    val dimension = Dimension {
        name = "UNIQUE_PAGES"
        value = "URLS"
    }
```

```
    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = "PAGES_VISITED"
        unit = StandardUnit.None
        value = dataPoint
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
        dimensions = listOf(dimension)
    }

    val request = PutMetricDataRequest {
        namespace = "SITE/TRAFFIC"
        metricData = listOf(datum)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric PAGES_VISITED")
    }
}

suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
}

suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
```

```
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}

fun readFileAsString(file: String): String {
    return String(Files.readAllBytes(Paths.get(file)))
}

suspend fun getMetricStatistics(costDateWeek: String?) {
    val start = Instant.parse(costDateWeek)
    val endDate = Instant.now()
    val dimension = Dimension {
        name = "Currency"
        value = "USD"
    }

    val dimensionList: MutableList<Dimension> = ArrayList()
    dimensionList.add(dimension)

    val statisticsRequest = GetMetricStatisticsRequest {
        metricName = "EstimatedCharges"
        namespace = "AWS/Billing"
        dimensions = dimensionList
        statistics = listOf(Statistic.Maximum)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        period = 86400
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data: List<Datapoint>? = response.datapoints
        if (data != null) {
            if (!data.isEmpty()) {
                for (datapoint in data) {
```

```
                    println("Timestamp:  ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
```

```
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}

suspend fun getSpecificMet(namespaceVal: String?): Dimension? {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(request)
        val myList = response.metrics
        if (myList != null) {
            return myList[0].dimensions?.get(0)
        }
    }
    return null
}

suspend fun listNameSpaces(): ArrayList<String> {
    val nameSpaceList = ArrayList<String>()
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(ListMetricsRequest {})
        response.metrics?.forEach { metrics ->
            val data = metrics.namespace
            if (!nameSpaceList.contains(data)) {
                nameSpaceList.add(data!!)
            }
        }
    }
    return nameSpaceList
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Kotlin.
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory
  - DescribeAlarms
  - DescribeAlarmsForMetric
  - DescribeAnomalyDetectors
  - GetMetricData
  - GetMetricStatistics
  - GetMetricWidgetImage
  - ListMetrics
  - PutAnomalyDetector
  - PutDashboard
  - PutMetricAlarm
  - PutMetricData

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo con un SDK CloudWatch AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Gestisci CloudWatch metriche e allarmi utilizzando un SDK AWS

L'esempio di codice seguente mostra come:

- Crea una sveglia per monitorare una CloudWatch metrica.
- Inserisci i dati in un parametro e attiva l'allarme.
- Ottenere i dati dall'allarme.
- Eliminare l'allarme.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea una classe che racchiuda le operazioni. CloudWatch

```
from datetime import datetime, timedelta
import logging
from pprint import pprint
import random
import time
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
```

```
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise


    def create_metric_alarm(
        self,
        metric_namespace,
        metric_name,
        alarm_name,
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
```

```
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
the
                             alarm threshold before the alarm is set into an
alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
statistic.
        :param comparison_op: The comparison operation used to compare the
threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm


    def put_metric_data(self, namespace, name, value, unit):
```

```
        """
        Sends a single data value to CloudWatch for a metric. This metric is
given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
name)
            raise


    def get_metric_statistics(self, namespace, name, start, end, period,
stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
value
                           or maximum value.
```

```
        :return: The retrieved statistics for the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
metric_name)
        return alarm_iter


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
```

```
            metric = self.cloudwatch_resource.Metric(metric_namespace,
 metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

Usa la classe wrapper per inserire i dati in un parametro, attivare un allarme che osserva il parametro e ottenere dati dall'allarme.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon CloudWatch metrics and alarms demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    cw_wrapper = CloudWatchWrapper(boto3.resource("cloudwatch"))

    minutes = 20
    metric_namespace = "doc-example-metric"
    metric_name = "page_views"
    start = datetime.utcnow() - timedelta(minutes=minutes)
    print(
        f"Putting data into metric {metric_namespace}.{metric_name} spanning the
 "
        f"last {minutes} minutes."
    )
    for offset in range(0, minutes):
        stamp = start + timedelta(minutes=offset)
        cw_wrapper.put_metric_data_set(
            metric_namespace,
            metric_name,
```

```
            stamp,
            "Count",
            {
                "values": [
                    random.randint(bound, bound * 2)
                    for bound in range(offset + 1, offset + 11)
                ],
                "counts": [random.randint(1, offset + 1) for _ in range(10)],
            },
        )

    alarm_name = "high_page_views"
    period = 60
    eval_periods = 2
    print(f"Creating alarm {alarm_name} for metric {metric_name}.")
    alarm = cw_wrapper.create_metric_alarm(
        metric_namespace,
        metric_name,
        alarm_name,
        "Maximum",
        period,
        eval_periods,
        100,
        "GreaterThanThreshold",
    )
    print(f"Alarm ARN is {alarm.alarm_arn}.")
    print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Sending data to trigger the alarm. This requires data over the
threshold "
        f"for {eval_periods} periods of {period} seconds each."
    )
    while alarm.state_value == "INSUFFICIENT_DATA":
        print("Sending data for the metric.")
        cw_wrapper.put_metric_data(
            metric_namespace, metric_name, random.randint(100, 200), "Count"
        )
        alarm.load()
        print(f"Current alarm state is: {alarm.state_value}.")
        if alarm.state_value == "INSUFFICIENT_DATA":
            print(f"Waiting for {period} seconds...")
            time.sleep(period)
        else:
```

```
            print("Wait for a minute for eventual consistency of metric data.")
            time.sleep(period)
            if alarm.state_value == "OK":
                alarm.load()
                print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Getting data for metric {metric_namespace}.{metric_name} during
timespan "
        f"of {start} to {datetime.utcnow()} (times are UTC)."
    )
    stats = cw_wrapper.get_metric_statistics(
        metric_namespace,
        metric_name,
        start,
        datetime.utcnow(),
        60,
        ["Average", "Minimum", "Maximum"],
    )
    print(
        f"Got {len(stats['Datapoints'])} data points for metric "
        f"{metric_namespace}.{metric_name}."
    )
    pprint(sorted(stats["Datapoints"], key=lambda x: x["Timestamp"]))

    print(f"Getting alarms for metric {metric_name}.")
    alarms = cw_wrapper.get_metric_alarms(metric_namespace, metric_name)
    for alarm in alarms:
        print(f"Alarm {alarm.name} is currently in state {alarm.state_value}.")

    print(f"Deleting alarms for metric {metric_name}.")
    cw_wrapper.delete_metric_alarms(metric_namespace, metric_name)

    print("Thanks for watching!")
    print("-" * 88)
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - DeleteAlarms

- DescribeAlarmsForMetric
- DisableAlarmActions
- EnableAlarmActions
- GetMetricStatistics
- ListMetrics
- PutMetricAlarm
- PutMetricData

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo con un SDK CloudWatch AWS Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi multidisciplinari per CloudWatch l'utilizzo degli SDK AWS

Le seguenti applicazioni di esempio utilizzano AWS gli SDK per combinarsi CloudWatch con altri. Servizi AWS Ogni esempio include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire l'applicazione.

Esempi

- Monitora le prestazioni di Amazon DynamoDB utilizzando un SDK AWS

Monitora le prestazioni di Amazon DynamoDB utilizzando un SDK AWS

Il seguente esempio di codice mostra come configurare l'uso di DynamoDB da parte di un'applicazione per monitorare le prestazioni.

Java

SDK per Java 2.x

Questo esempio mostra come configurare un'applicazione Java per monitorare le prestazioni di DynamoDB. L'applicazione invia i dati metrici a CloudWatch cui è possibile monitorare le prestazioni.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- CloudWatch
- DynamoDB

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo con un SDK CloudWatch AWS. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Sicurezza in Amazon CloudWatch

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili CloudWatch, consulta AWS Servizi nell'ambito del programma di conformitàAWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i tuoi requisiti aziendali e le leggi e le normative applicabili

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi Amazon CloudWatch. Ti mostra come configurare Amazon per CloudWatch soddisfare i tuoi obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere CloudWatch le tue risorse.

Indice

Protezione dei dati in Amazon CloudWatch

Il modello di responsabilità AWS condivisa di si applica alla protezione dei dati in Amazon CloudWatch. Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori o Servizi AWS utilizzi la console, l'API CloudWatch o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Crittografia in transito

CloudWatch utilizza end-to-end la crittografia dei dati in transito.

Gestione delle identità e degli accessi per Amazon CloudWatch

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. CloudWatch IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Riferimento alle CloudWatch autorizzazioni Amazon

Destinatari

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che CloudWatch svolgi.

Utente del servizio: se utilizzi il CloudWatch servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più CloudWatch funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in CloudWatch, consultaRisoluzione dei problemi relativi all' CloudWatch identità e all'accesso ad Amazon.

Amministratore del servizio: se sei responsabile delle CloudWatch risorse della tua azienda, probabilmente hai pieno accesso a CloudWatch. È tuo compito determinare a quali CloudWatch funzionalità e risorse devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con CloudWatch, consultaCome CloudWatch funziona Amazon con IAM.

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy a cui gestire l'accesso CloudWatch. Per visualizzare esempi di policy CloudWatch basate sull'identità che puoi utilizzare in IAM, consulta. Esempi di policy basate sull'identità per Amazon CloudWatch

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta Signing AWS API request nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conservare le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente di IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di

utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni sul Centro identità IAM, consulta Cos'è Centro identità IAM? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente di IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Quando creare un utente IAM (invece di un ruolo) nella Guida per l'utente di IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene

autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per ulteriori informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I

ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta Quando creare un ruolo IAM (invece di un utente) nella Guida per l'utente di IAM.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente di IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'azione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruoli IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente di IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta Panoramica delle liste di controllo degli accessi (ACL) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Politiche di controllo dei servizi (SCP): le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Come CloudWatch funziona Amazon con IAM

Prima di utilizzare IAM per gestire l'accesso a CloudWatch, scopri con quali funzionalità IAM è disponibile l'uso CloudWatch.

Funzionalità IAM che puoi utilizzare con Amazon CloudWatch

Funzionalità IAM	CloudWatch supporto
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
Liste di controllo degli accessi (ACL)	No
ABAC (tag nelle policy)	Parziale
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	No

Per avere una panoramica di alto livello su come CloudWatch e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Politiche basate sull'identità per CloudWatch

Supporta le policy basate su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di politiche basate sull'identità per CloudWatch

Per visualizzare esempi di politiche basate sull' CloudWatch identità, vedere. Esempi di policy basate sull'identità per Amazon CloudWatch

Politiche basate sulle risorse all'interno CloudWatch

Supporta le policy basate su risorse	No

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste

ulteriori policy basate su identità. Per ulteriori informazioni, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Azioni politiche per CloudWatch

Supporta le operazioni di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di CloudWatch azioni, consulta Azioni definite da Amazon CloudWatch nel Service Authorization Reference.

Le azioni politiche in CloudWatch uso utilizzano il seguente prefisso prima dell'azione:

```
cloudwatch
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "cloudwatch:action1",
      "cloudwatch:action2"
         ]
```

Per visualizzare esempi di politiche CloudWatch basate sull'identità, vedere. Esempi di policy basate sull'identità per Amazon CloudWatch

Risorse politiche per CloudWatch

Supporta le risorse di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di CloudWatch risorse e dei relativi ARN, consulta Resources defined by Amazon CloudWatch nel Service Authorization Reference. Per sapere con quali azioni puoi specificare l'ARN di ogni risorsa, consulta Azioni definite da Amazon. CloudWatch

Per visualizzare esempi di politiche CloudWatch basate sull'identità, consulta. Esempi di policy basate sull'identità per Amazon CloudWatch

Chiavi relative alle condizioni delle politiche per CloudWatch

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni

condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND` logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco di chiavi di CloudWatch condizione, consulta Condition keys for Amazon CloudWatch nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta Azioni definite da Amazon CloudWatch.

Per visualizzare esempi di politiche CloudWatch basate sull'identità, consulta. Esempi di policy basate sull'identità per Amazon CloudWatch

ACL in CloudWatch

Supporta le ACL	No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

ABAC con CloudWatch

Supporta ABAC (tag nelle policy)	Parziale

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Che cos'è ABAC? nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con CloudWatch

Supporta le credenziali temporanee	Sì

Alcuni Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Cambio di un ruolo (console) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare

dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni principali multiservizio per CloudWatch

Supporta l'inoltro delle sessioni di accesso (FAS)	Sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

Ruoli di servizio per CloudWatch

Supporta i ruoli di servizio	Sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità. CloudWatch Modifica i ruoli di servizio solo quando viene CloudWatch fornita una guida in tal senso.

Esempi di policy basate sull'identità per Amazon CloudWatch

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare risorse. CloudWatch Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS l'API. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Per dettagli sulle azioni e sui tipi di risorse definiti da CloudWatch, incluso il formato degli ARN per ciascun tipo di risorsa, consulta Azioni, risorse e chiavi di condizione per Amazon CloudWatch nel Service Authorization Reference.

Argomenti

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare CloudWatch risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.

- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso ad azioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console di CloudWatch

Per accedere alla CloudWatch console Amazon, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle CloudWatch risorse del tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la CloudWatch console, allega anche la policy CloudWatch `ConsoleAccess` o la policy `ReadOnly` AWS gestita alle entità. Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente IAM.

Autorizzazioni necessarie per la console CloudWatch

Di seguito è riportato il set completo di autorizzazioni necessarie per utilizzare la CloudWatch console. Queste autorizzazioni forniscono l'accesso completo in scrittura e lettura alla CloudWatch console.

- scalabilità automatica delle applicazioni: DescribeScalingPolicies
- scalabilità automatica: DescribeAutoScalingGroups
- scalabilità automatica: DescribePolicies
- pista nuvolosa: DescribeTrails
- orologio cloud: DeleteAlarms
- orologio nuvoloso: DescribeAlarmHistory
- orologio nuvoloso: DescribeAlarms
- orologio nuvoloso: GetMetricData
- orologio nuvoloso: GetMetricStatistics
- orologio nuvoloso: ListMetrics
- orologio nuvoloso: PutMetricAlarm
- orologio nuvoloso: PutMetricData
- ec2: DescribeInstances
- ec2: DescribeTags
- ec2: DescribeVolumes
- Sì: DescribeElasticsearchDomain
- Sì: ListDomainNames
- eventi: DeleteRule
- eventi: DescribeRule
- eventi: DisableRule
- eventi: EnableRule
- eventi: ListRules
- eventi: PutRule
- sono: AttachRolePolicy
- Io sono: CreateRole
- Io sono: GetPolicy

- Io sono: GetPolicyVersion
- Io sono: GetRole
- Io sono: ListAttachedRolePolicies
- Io sono: ListRoles
- cinesi: DescribeStream
- cinesi: ListStreams
- lambda: AddPermission
- lambda: CreateFunction
- lambda: GetFunctionConfiguration
- lambda: ListAliases
- lambda: ListFunctions
- lambda: ListVersionsByFunction
- lambda: RemovePermission
- registri: CancelExportTask
- registri: CreateExportTask
- registri: CreateLogGroup
- registri: CreateLogStream
- registri: DeleteLogGroup
- registri: DeleteLogStream
- registri: DeleteMetricFilter
- registri: DeleteRetentionPolicy
- registri: DeleteSubscriptionFilter
- registri: DescribeExportTasks
- registri: DescribeLogGroups
- registri: DescribeLogStreams
- registri: DescribeMetricFilters
- registri: DescribeQueries
- registri: DescribeSubscriptionFilters
- registri: FilterLogEvents
- registri: GetLogGroupFields

- registri: GetLogRecord
- registri: GetLogEvents
- registri: GetQueryResults
- registri: PutMetricFilter
- registri: PutRetentionPolicy
- registri: PutSubscriptionFilter
- registri: StartQuery
- registri: StopQuery
- registri: TestMetricFilter
- s3: CreateBucket
- s3: ListBucket
- figlio: CreateTopic
- nss: GetTopicAttributes
- nss: ListSubscriptions
- nss: ListTopics
- nss: SetTopicAttributes
- sns:Subscribe
- sns:Unsubscribe
- sqs: GetQueueAttributes
- sq: GetQueueUrl
- sq: ListQueues
- sq: SetQueueAttributes
- swf: CreateAction
- file swf: DescribeAction
- file swf: ListActionTemplates
- file swf: RegisterAction
- file swf: RegisterDomain
- file swf: UpdateAction

Inoltre, per visualizzare la mappa di tracciamento X-Ray, è necessario `AWSXrayReadOnlyAccess`

Risoluzione dei problemi relativi all' CloudWatch identità e all'accesso ad Amazon

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con CloudWatch e IAM.

Argomenti

Non sono autorizzato a eseguire alcuna azione in CloudWatch

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `cloudwatch:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cloudwatch:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `cloudwatch:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di assegnare un ruolo a CloudWatch.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in. CloudWatch Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie CloudWatch risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se CloudWatch supporta queste funzionalità, consultaCome CloudWatch funziona Amazon con IAM.
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente IAM.

CloudWatch aggiornamento delle autorizzazioni del pannello di controllo

Il 1° maggio 2018, AWS sono state modificate le autorizzazioni necessarie per accedere alle dashboard. CloudWatch L'accesso alla dashboard nella CloudWatch console ora richiede le autorizzazioni introdotte nel 2017 per supportare le operazioni delle API della dashboard:

- cloudwatch: GetDashboard
- orologio nuvoloso: ListDashboards
- orologio nuvoloso: PutDashboard
- orologio nuvoloso: DeleteDashboards

Per accedere ai CloudWatch dashboard, è necessario uno dei seguenti:

- La AdministratorAccesspolitica.
- La CloudWatchFullAccesspolitica.
- Una policy personalizzata che includa una o più di queste autorizzazioni specifiche:
  - `cloudwatch:GetDashboard` e `cloudwatch:ListDashboards` per poter visualizzare i pannelli di controllo
  - `cloudwatch:PutDashboard` per poter creare o modificare i pannelli di controllo
  - `cloudwatch:DeleteDashboards` per poter eliminare i pannelli di controllo

Per ulteriori informazioni sull'utilizzo delle policy per modificare le autorizzazioni per un utente IAM, consulta Modifica delle autorizzazioni per un utente di IAM.

Per ulteriori informazioni sulle CloudWatch autorizzazioni, vedereRiferimento alle CloudWatch autorizzazioni Amazon.

Per ulteriori informazioni sulle operazioni delle API della dashboard, PutDashboardconsulta Amazon CloudWatch API Reference.

AWS politiche gestite (predefinite) per CloudWatch

AWS affronta molti casi d'uso comuni fornendo politiche IAM autonome create e amministrate da AWS. Queste policy AWS gestite concedono le autorizzazioni necessarie per i casi d'uso comuni in modo da evitare di dover esaminare quali autorizzazioni sono necessarie. Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

Le seguenti politiche AWS gestite, che puoi allegare agli utenti del tuo account, sono specifiche per. CloudWatch

Argomenti

CloudWatchFullAccessV2

AWS ha recentemente aggiunto la policy IAM gestita CloudWatchFullAccessV2. Questa politica garantisce l'accesso completo alle CloudWatch azioni e alle risorse e definisce anche in modo più adeguato le autorizzazioni concesse per altri servizi come Amazon SNS e. Amazon EC2 Auto Scaling Ti consigliamo di iniziare a utilizzare questa politica anziché utilizzare. CloudWatchFullAccess AWS piani per diventare obsoleti CloudWatchFullAccessnelle prossime future.

Include `application-signals:` le autorizzazioni che consentono agli utenti di accedere a tutte le funzionalità dalla CloudWatch console in Application Signals. Include alcune `autoscaling:Describe` autorizzazioni in modo che gli utenti con questo criterio possano vedere le azioni di Auto Scaling associate CloudWatch agli allarmi. Include alcune `sns` autorizzazioni in modo che gli utenti con questa politica possano recuperare, creare argomenti Amazon SNS e associarli agli allarmi. CloudWatch Include le autorizzazioni IAM in modo che gli utenti con questa policy possano visualizzare le informazioni sui ruoli collegati ai servizi associati a. CloudWatch Include le autorizzazioni `oam:ListSinks` e le `oam:ListAttachedLinks` autorizzazioni in modo che gli utenti con questa politica possano utilizzare la console per visualizzare i dati condivisi dagli account di origine in modo osservabile su più account. CloudWatch

Include `rum` e `xray` autorizzazioni in modo che gli utenti possano avere pieno accesso a CloudWatch Synthetics CloudWatch e RUM AWS X-Ray, tutti inclusi nel servizio. `synthetics` CloudWatch

I contenuti della CloudWatchFullAccessV2 sono i seguenti:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchFullAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:*",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribePolicies",
                "cloudwatch:*",
                "logs:*",
                "sns:CreateTopic",
                "sns:ListSubscriptions",
                "sns:ListSubscriptionsByTopic",
                "sns:ListTopics",
                "sns:Subscribe",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks",
                "rum:*",
                "synthetics:*",
                "xray:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsServiceLinkedRolePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "application-
signals.cloudwatch.amazonaws.com"
                }
```

```
            }
        },
        {
            "Sid": "EventsServicePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
         {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        }
    ]
}
```

CloudWatchFullAccess

La CloudWatchFullAccesspolitica è sulla buona strada verso la deprecazione. Ti consigliamo di smettere di usarla e di utilizzare invece la V2. CloudWatchFullAccess

I contenuti di CloudWatchFullAccesssono i seguenti:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:*",
                "logs:*",
                "sns:*",
                "iam:GetPolicy",
```

```
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        }
    ]
}
```

CloudWatchReadOnlyAccess

La CloudWatchReadOnlyAccesspolitica garantisce l'accesso in sola lettura a. CloudWatch

La politica include alcune `logs:` autorizzazioni, in modo che gli utenti con questa politica possano utilizzare la console per visualizzare le informazioni di Logs e le query di CloudWatch Logs Insights. CloudWatch Include`autoscaling:Describe*`, in modo che gli utenti con questo criterio possano vedere le azioni di Auto Scaling associate CloudWatch agli allarmi. Include le `application-signals:` autorizzazioni che consentono agli utenti di utilizzare Application Signals per monitorare lo stato dei propri servizi. Include `application-autoscaling:DescribeScalingPolicies`, in modo che gli utenti con questa policy possano accedere alle informazioni sulle policy Application Auto Scaling. Include `sns:Get*` `esns:List*`, in modo che gli utenti con questa politica possano recuperare informazioni sugli argomenti di Amazon SNS che ricevono notifiche CloudWatch sugli allarmi. Include le `oam:ListAttachedLinks` autorizzazioni `oam:ListSinks` e, in modo che gli utenti con questa politica possano utilizzare la console per visualizzare i dati condivisi dagli account

di origine in modo osservabile su più account. CloudWatch Include le `iam:GetRole` autorizzazioni in modo che gli utenti possano verificare se CloudWatch Application Signals è stato configurato.

Include `rum` e `xray` autorizzazioni in modo che gli utenti possano avere accesso in sola lettura a Synthetics CloudWatch e CloudWatch RUM AWS X-Ray, tutti inclusi nel servizio. `synthetics` CloudWatch

Di seguito è riportato il contenuto della politica. CloudWatchReadOnlyAccess

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:BatchGet*",
                "application-signals:Get*",
                "application-signals:List*",
                "autoscaling:Describe*",
                "cloudwatch:BatchGet*",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:Describe*",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "oam:ListSinks",
                "sns:Get*",
                "sns:List*",
                "rum:BatchGet*",
                "rum:Get*",
                "rum:List*",
                "synthetics:Describe*",
                "synthetics:Get*",
```

```
                "synthetics:List*",
                "xray:BatchGet*",
                "xray:Get*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        },
        {
            "Sid": "CloudWatchReadOnlyGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
        }
     ]
}
```

CloudWatchActionsEC2Access

La policy CloudWatchActionsEC2Access garantisce l'accesso in sola lettura ad CloudWatch allarmi e metriche oltre ai metadati di Amazon EC2. Concede inoltre l'accesso alle operazioni API di arresto, termine e riavvio per le istanze EC2.

Di seguito è riportato il contenuto della policy EC2Access. CloudWatchActions

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:Describe*",
        "ec2:Describe*",
        "ec2:RebootInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

CloudWatchAutomaticDashboardsAccess

La policy CloudWatch- CrossAccountAccess gestita viene utilizzata dal ruolo CloudWatch-CrossAccountSharingRole IAM. Questo ruolo e la policy consentono agli utenti dei pannelli di controllo tra più account di visualizzare pannelli di controllo automatici in ciascun account che condivide i pannelli di controllo.

Di seguito è riportato il contenuto di CloudWatchAutomaticDashboardsAccess:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "cloudfront:GetDistribution",
        "cloudfront:ListDistributions",
        "dynamodb:DescribeTable",
        "dynamodb:ListTables",
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "elasticache:DescribeCacheClusters",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticfilesystem:DescribeFileSystems",
        "elasticloadbalancing:DescribeLoadBalancers",
        "kinesis:DescribeStream",
        "kinesis:ListStreams",
        "lambda:GetFunction",
        "lambda:ListFunctions",
        "rds:DescribeDBClusters",
        "rds:DescribeDBInstances",
        "resource-groups:ListGroupResources",
```

```
        "resource-groups:ListGroups",
        "route53:GetHealthCheck",
        "route53:ListHealthChecks",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sns:ListTopics",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "sqs:ListQueues",
        "synthetics:DescribeCanariesLastRun",
        "tag:GetResources"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "apigateway:GET"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:apigateway:*::/restapis*"
      ]
    }
  ]
```

CloudWatchAgentServerPolicy

La CloudWatchAgentServerPolicypolicy può essere utilizzata nei ruoli IAM collegati alle istanze Amazon EC2 per consentire all' CloudWatch agente di leggere le informazioni dall'istanza e scriverle su. CloudWatch I suoi contenuti sono i seguenti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchServerPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeVolumes",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
```

```
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMServerPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
    ]
}
```

CloudWatchAgentAdminPolicy

La CloudWatchAgentAdminPolicypolicy può essere utilizzata nei ruoli IAM collegati alle istanze Amazon EC2. Questa policy consente all' CloudWatch agente di leggere le informazioni dall'istanza e di scriverle CloudWatch, nonché di scrivere informazioni su Parameter Store. I suoi contenuti sono i seguenti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
```

```
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
    ]
}
```

> **Note**
>
> Per esaminare queste policy di autorizzazione, accedi alla console IAM ed esegui la ricerca delle policy specifiche.

Puoi anche creare policy IAM personalizzate per consentire CloudWatch autorizzazioni per azioni e risorse. Puoi associare queste policy personalizzate agli utenti o ai gruppi IAM che richiedono tali autorizzazioni.

AWS politiche gestite (predefinite) per CloudWatch l'osservabilità tra account

Le politiche di questa sezione concedono le autorizzazioni relative all'osservabilità tra account. CloudWatch Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

CloudWatchCrossAccountSharingConfiguration

La CloudWatchCrossAccountSharingConfigurationpolitica consente l'accesso per creare, gestire e visualizzare i collegamenti di Observability Access Manager per la condivisione di risorse tra account.

CloudWatch Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account. I contenuti sono come indicato di seguito:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Link",
                "oam:ListLinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

OAM FullAccess

La FullAccess politica OAM consente l'accesso per creare, gestire e visualizzare i sink e i link di Observability Access Manager, utilizzati per l'osservabilità tra account. CloudWatch

La FullAccess politica OAM di per sé non consente di condividere i dati di osservabilità tra link. Per creare un link per condividere le CloudWatch metriche, è necessario anche uno o. CloudWatchFullAccessCloudWatchCrossAccountSharingConfiguration Per creare un link per condividere i gruppi di log di CloudWatch Logs, è necessario anche uno o CloudWatchLogsFullAccess. CloudWatchLogsCrossAccountSharingConfiguration Per creare un link per condividere le tracce a raggi X, è necessario anche uno o AWSXRayFullAccess. AWSXRayCrossAccountSharingConfiguration

Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account. I contenuti sono come indicato di seguito:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:*"
            ],
            "Resource": "*"
        }
    ]
}
```

OAM ReadOnlyAccess

La ReadOnlyAccess politica OAM garantisce l'accesso in sola lettura alle risorse di Observability Access Manager, che vengono utilizzate per l'osservabilità tra account. CloudWatch Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account. I contenuti sono come indicato di seguito:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:Get*",
                "oam:List*"
            ],
            "Resource": "*"
```

```
        }
    ]
}
```

AWS politiche gestite (predefinite) per CloudWatch Synthetics

Le policy CloudWatchSyntheticsReadOnlyAccess AWS gestite sono disponibili CloudWatchSyntheticsFullAccesse possono essere assegnate agli utenti che gestiranno o utilizzeranno CloudWatch Synthetics. Sono rilevanti anche le seguenti policy aggiuntive:

- AmazonS3 ReadOnlyAccess e CloudWatchReadOnlyAccess— Questi sono necessari per poter leggere tutti i dati Synthetics nella console. CloudWatch
- AWSLambdaReadOnlyAccess— Per poter visualizzare il codice sorgente usato dai canarini.
- CloudWatchSyntheticsFullAccessconsente di creare canarini. Inoltre, per creare ed eliminare canarini a cui è stato creato un nuovo ruolo IAM, è necessaria anche la seguente dichiarazione politica in linea:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

> **Important**
>
> Concedere a un utente le autorizzazioni `iam:CreateRole`, `iam:DeleteRole`, `iam:CreatePolicy`, `iam:DeletePolicy`, `iam:AttachRolePolicy` e `iam:DetachRolePolicy` fornisce all'utente l'accesso amministrativo completo per creare, allegare ed eliminare ruoli e policy con ARN corrispondenti`arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*earn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*`. Ad esempio, un utente con queste autorizzazioni può creare una policy che dispone di autorizzazioni complete per tutte le risorse e collegare tale policy a qualsiasi ruolo che coincide con il modello ARN. Presta molta attenzione a chi concedi queste autorizzazioni.

Per informazioni su come collegare policy e concedere autorizzazioni a utenti, consulta Modifica delle autorizzazioni per un utente IAM e Per incorporare una policy inline per un utente o un ruolo.

CloudWatchSyntheticsFullAccess

Di seguito è riportato il contenuto della policy. CloudWatchSyntheticsFullAccess

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutEncryptionConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::cw-syn-results-*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "s3:ListAllMyBuckets",
                "xray:GetTraceSummaries",
                "xray:BatchGetTraces",
                "apigateway:GET"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::cw-syn-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::aws-synthetics-library-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ],
            "Condition": {
```

```
                "StringEquals": {
                    "iam:PassedToService": [
                        "lambda.amazonaws.com",
                        "synthetics.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:ListAttachedRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:AddPermission",
                "lambda:PublishVersion",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "lambda:GetFunctionConfiguration",
                "lambda:DeleteFunction"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:function:cwsyn-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:GetLayerVersion",
                "lambda:PublishLayerVersion",
                "lambda:DeleteLayerVersion"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:layer:cwsyn-*",
                "arn:aws:lambda:*:*:layer:Synthetics:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:*:*:key/*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "s3.*.amazonaws.com"
                    ]
                }
            }
        }
    ]
```

```
}
```

CloudWatchSyntheticsReadOnlyAccess

Di seguito è riportato il contenuto della CloudWatchSyntheticsReadOnlyAccesspolitica.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politiche gestite (predefinite) per Amazon RUM CloudWatch

Le AmazonCloudWatch politiche ReadOnlyAccess AWS gestite AmazonCloudWatchRUM FullAccess e RUM possono essere assegnate agli utenti che gestiranno o utilizzeranno CloudWatch RUM.

AmazonCloudWatchRUM FullAccess

Di seguito sono riportati i contenuti della FullAccess politica AmazonCloudWatchRUM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rum:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "iam:GetRole",
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-service-role/rum.amazonaws.com/
AWSServiceRoleForRealUserMonitoring"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/RUM-Monitor*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "cognito-identity.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": "arn:aws:cloudwatch:*:*:alarm:*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "cognito-identity:CreateIdentityPool",
                "cognito-identity:ListIdentityPools",
                "cognito-identity:DescribeIdentityPool",
                "cognito-identity:GetIdentityPoolRoles",
                "cognito-identity:SetIdentityPoolRoles"
            ],
            "Resource": "arn:aws:cognito-identity:*:*:identitypool/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:DeleteLogGroup",
                "logs:PutRetentionPolicy",
                "logs:CreateLogStream"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:*RUMService*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:*:*:log-group::log-stream:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:describeCanaries",
                "synthetics:describeCanariesLastRun"
            ],
            "Resource": "arn:aws:synthetics:*:*:canary:*"
```

```
        }
    ]
}
```

AmazonCloudWatchRUM ReadOnlyAccess

Di seguito sono riportati i contenuti della ReadOnlyAccess politica AmazonCloudWatchRUM.

```
{

 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "rum:GetAppMonitor",
    "rum:GetAppMonitorData",
    "rum:ListAppMonitors",
    "rum:ListRumMetricsDestinations",
    "rum:BatchGetRumMetricDefinitions"
   ],
   "Resource": "*"
  }
 ]
}
```

AmazonCloudWatchRUM ServiceRolePolicy

Non puoi collegare AmazonCloudWatchRUM ServiceRolePolicy alle tue entità IAM. Questa policy è associata a un ruolo collegato al servizio che consente a CloudWatch RUM di pubblicare i dati di monitoraggio su altri servizi pertinenti AWS . Per ulteriori informazioni su questo ruolo collegato alservizio, consultare Utilizzo di ruoli collegati ai servizi per RUM CloudWatch.

Il contenuto completo di AmazonCloudWatchRUM è il ServiceRolePolicy seguente.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
```

```
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
      "AWS/RUM"
     ]
    }
   }
  }
 ]
}
```

AWS politiche gestite (predefinite) per CloudWatch Evidently

Le politiche CloudWatchEvidentlyReadOnlyAccess AWS gestite sono disponibili CloudWatchEvidentlyFullAccesse possono essere assegnate agli utenti che gestiranno o CloudWatch utilizzeranno Evidently.

CloudWatchEvidentlyFullAccess

Di seguito sono riportati i contenuti della CloudWatchEvidentlyFullAccesspolitica.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evidently:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "cloudwatch:TagResource",
                "cloudwatch:UnTagResource"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "cloudtrail:LookupEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Evidently-Alarm-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Evidently-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ]
        }
```

```
    ]
}
```

CloudWatchEvidentlyReadOnlyAccess

Di seguito sono riportati i contenuti della CloudWatchEvidentlyReadOnlyAccesspolitica.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
                "evidently:ListExperiments",
                "evidently:ListFeatures",
                "evidently:ListLaunches",
                "evidently:ListProjects",
                "evidently:ListSegments",
                "evidently:ListSegmentReferencs"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS policy gestita per AWS Systems Manager Incident Manager

La AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicypolicy è associata a un ruolo collegato al servizio che consente di CloudWatch avviare incidenti in AWS Systems Manager Incident Manager per conto dell'utente. Per ulteriori informazioni, consulta la pagina Autorizzazioni di ruolo collegate ai servizi per le azioni di CloudWatch Systems Manager Incident Manager degli allarmi.

La policy ha le seguenti autorizzazioni:

- incidenti ssm: StartIncident

Esempi di policy gestite dal cliente

In questa sezione, puoi trovare esempi di politiche utente che concedono autorizzazioni per varie CloudWatch azioni. Queste politiche funzionano quando utilizzi l' CloudWatch API, gli AWS SDK o il. AWS CLI

Esempi

Esempio 1: consenti all'utente l'accesso completo a CloudWatch

Per concedere a un utente l'accesso completo a CloudWatch, puoi utilizzare la politica CloudWatchFullAccessgestita anziché creare una politica gestita dal cliente. I contenuti di CloudWatchFullAccesssono elencati in. CloudWatchFullAccess

Esempio 2: consentire l'accesso in sola lettura a CloudWatch

La seguente policy consente a un utente di accedere CloudWatch e visualizzare in sola lettura le CloudWatch azioni, i CloudWatch parametri, i dati di log e i dati di Amazon SNS relativi agli allarmi di Amazon EC2 Auto Scaling.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "logs:Get*",
        "logs:Describe*",
        "logs:StartQuery",
        "logs:StopQuery",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "logs:StartLiveTail",
        "logs:StopLiveTail",
```

```
        "sns:Get*",
        "sns:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Esempio 3: interrompere o terminare un'istanza Amazon EC2

La seguente policy consente a un'azione di allarme di interrompere o terminare un'istanza EC2 CloudWatch . Nell'esempio seguente GetMetricData ListMetrics, le DescribeAlarms azioni e sono facoltative. Ti consigliamo di includere tali operazioni per assicurarti di aver arrestato o terminato correttamente l'istanza.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
```

```
  ]
}
```

CloudWatch aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite CloudWatch da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei CloudWatch documenti.

Modifica	Descrizione	Data
CloudWatchFullAccessV2 — Aggiornamento a una politica esistente	CloudWatch ha aggiornato la politica denominata CloudWatchFullAccessV2. L'ambito della `CloudWatchFullAccessPermissions` politica è stato aggiornato per consentire agli utenti di utilizzare CloudWatch Application Signals per visualizzare, analizzare e diagnosticare problemi relativi allo stato dei propri servizi. `application-signals:*`	20 maggio 2024
CloudWatchReadOnlyAccess: aggiornamento a una policy esistente	CloudWatch ha aggiornato la politica denominata CloudWatchReadOnlyAccess. L'ambito della `CloudWatchReadOnlyAccessPermissions` politica è stato aggiornato per aggiungere `application-signals:BatchGet*`	20 maggio 2024

Modifica	Descrizione	Data
	e `application-signals:Get*` consentire agli utenti di utilizzare CloudWatch Application Signals per visualizzare, analizzare e diagnosticare problemi relativi allo stato dei propri servizi. `application-signals:List*` L'ambito di `CloudWatchReadOnly` `GetRolePermissions` è stato aggiornato per aggiungere l'`iam:GetRole` azione in modo che gli utenti possano verificare se CloudWatch Application Signals è configurato.	
CloudWatchApplicationSignalsServiceRolePolicy: aggiornamento a una policy esistente	CloudWatch ha aggiornato la politica denominata CloudWatchApplicationSignalsServiceRolePolicy. L'ambito delle `logs:GetQueryResults` autorizzazioni `logs:StartQuery` e è stato modificato per aggiungere gli `arn:aws:logs:*:*:log-group:/aws/application-signals/data:*` ARN `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` e abilitare Application Signals su più architetture.	18 aprile 2024

Modifica	Descrizione	Data
CloudWatchApplicationSignalsServiceRolePolicy : aggiornamento a una policy esistente	CloudWatch ha modificato l'ambito di un'autorizzazione in CloudWatchApplicationSignalsServiceRolePolicy. L'ambito dell'`cloudwatch:GetMetricData` autorizzazione è stato modificato `*` in modo che Application Signals possa recuperare le metriche dalle fonti negli account collegati.	08 aprile 2024
CloudWatchAgentServerPolicy : aggiornamento a una policy esistente	CloudWatch ha aggiunto i permessi a. CloudWatchAgentServerPolicy Le `logs:PutRetentionPolicy` autorizzazioni `xray:PutTraceSegments` `xray:PutTelemetryRecords` ,`xray:GetSamplingRules` ,`xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` e sono state aggiunte in modo che l' CloudWatch agente possa pubblicare tracce X-Ray e modificare i periodi di conservazione dei gruppi di log.	12 febbraio 2024

Modifica	Descrizione	Data
CloudWatchAgentAdminPolicy: aggiornamento a una policy esistente	CloudWatch ha aggiunto autorizzazioni a. CloudWatchAgentAdminPolicy Le `logs:PutRetentionPolicy` autorizzazioni `xray:PutTraceSegments` `xray:PutTelemetryRecords` ,`xray:GetSamplingRules` ,`xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` e sono state aggiunte in modo che l' CloudWatch agente possa pubblicare tracce X-Ray e modificare i periodi di conservazione dei gruppi di log.	12 febbraio 2024

Modifica	Descrizione	Data
CloudWatchFullAccessV2 — Aggiornamento a una politica esistente	CloudWatch autorizzazioni aggiunte alla CloudWatc hFullAccess V2. Sono state aggiunte le autorizzazioni esistenti per le azioni CloudWatch Synthetic s, CloudWatch X-Ray e RUM e nuove CloudWatch autorizza zioni per Application Signals in modo che gli utenti con questa politica possano gestire Application Signals. CloudWatch L'autorizzazione a creare il ruolo collegato al servizio CloudWatch Application Signals è stata aggiunta per consentire ad CloudWatch Application Signals di scoprire i dati di telemetria in log, metriche, tracce e tag.	5 dicembre 2023

Modifica	Descrizione	Data
CloudWatchReadOnlyAccess: aggiornamento a una policy esistente	CloudWatch CloudWatchReadOnlyAccessha aggiunto autorizzazioni a. Sono state aggiunte le autorizzazioni di sola lettura esistenti per le azioni CloudWatch Synthetics, X-Ray CloudWatch e RUM e le nuove autorizzazioni di sola lettura CloudWatch per Application Signals in modo che gli utenti con questa politica possano valutare e diagnosticare i problemi di integrità del servizio, come riportato da Application Signals. CloudWatch L'`cloudwatch:GenerateQuery` autorizzazione è stata aggiunta in modo che gli utenti con questa policy possano generare una stringa di query Metrics Insights da un prompt in linguaggio naturale. CloudWatch	5 dicembre 2023

Modifica	Descrizione	Data
CloudWatchApplicationSignalsServiceRolePolicy: nuova policy	CloudWatch ha aggiunto una nuova politica. CloudWatchApplicationSignalsServiceRolePolicy CloudWatchApplicationSignalsServiceRolePolicyConcede a una funzionalità imminente le autorizzazioni per raccogliere dati di CloudWatch registro, dati di traccia a raggi X, dati di CloudWatch metrica e dati di etichettatura.	9 novembre 2023
AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy: nuova policy	CloudWatch AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicyha aggiunto una nuova politica. AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicyConcede l'autorizzazione CloudWatch a recuperare le metriche di Performance Insights dai database per tuo conto.	20 settembre 2023

Modifica	Descrizione	Data
CloudWatchReadOnlyAccess: aggiornamento a una policy esistente	CloudWatch ha aggiunto un'autorizzazione a. CloudWatchReadOnlyAccess L'autorizzazione `application-autoscaling:DescribeScalingPolicies` è stata aggiunta in modo che gli utenti con questa policy possano accedere alle informazioni sulle policy Application Auto Scaling.	14 settembre 2023
CloudWatchFullAccessV2 — Nuova politica	CloudWatch ha aggiunto una nuova politica CloudWatchFullAccessV2. La CloudWatchFullAccessV2 garantisce l'accesso completo ad CloudWatch azioni e risorse, definendo al contempo meglio le autorizzazioni concesse ad altri servizi come Amazon SNS e. Amazon EC2 Auto ScalingPer ulteriori informazioni, consulta V2. CloudWatchFullAccess	1° agosto 2023

Modifica	Descrizione	Data
AWSServiceRoleForInternetMonitor: aggiornamento a una policy esistente	Amazon CloudWatch Internet Monitor ha aggiunto nuove autorizzazioni per monitorare le risorse di Network Load Balancer. Le autorizzazioni `elasticloadbalancing:DescribeLoadBalancers` e `ec2:DescribeNetworkInterfaces` sono necessarie affinché Monitor Internet possa monitorare il traffico Network Load Balancer dei clienti tramite l'analisi dei log di flusso per le risorse NLB. Per ulteriori informazioni, consulta Utilizzo di Amazon CloudWatch Internet Monitor.	15 luglio 2023

Modifica	Descrizione	Data
CloudWatchReadOnlyAccess: aggiornamento a una policy esistente	CloudWatch ha aggiunto autorizzazioni a. CloudWatchReadOnlyAccess Le `logs:StopLiveTail` autorizzazioni `logs:StartLiveTail` e sono state aggiunte in modo che gli utenti con questo criterio possano utilizzare la console per avviare e interrompere le sessioni live tail di CloudWatch Logs. Per ulteriori informazioni, consulta Use live tail to view logs in near real time.	6 giugno 2023
CloudWatchCrossAccountSharingConfiguration: nuova policy	CloudWatch ha aggiunto una nuova politica per consentire di gestire i link di osservabilità CloudWatch tra account che condividono le metriche. CloudWatch Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.	27 novembre 2022
FullAccessOAM — Nuova politica	CloudWatch ha aggiunto una nuova politica per consentirti di gestire completamente i link e i CloudWatch sink di osservabilità tra account. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.	27 novembre 2022

Modifica	Descrizione	Data
ReadOnlyAccessOAM — Nuova politica	CloudWatch ha aggiunto una nuova politica che consente di visualizzare le informazioni sui link e sui CloudWatch sink di osservabilità tra account. Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.	27 novembre 2022
CloudWatchFullAccess: aggiornamento a una policy esistente	CloudWatch ha aggiunto i permessi a. CloudWatchFullAccess Le `oam:ListAttachedLinks` autorizzazioni `oam:ListSinks` e sono state aggiunte in modo che gli utenti con questo criterio possano utilizzare la console per visualizzare i dati condivisi dagli account di origine in modo osservabile CloudWatch tra più account.	27 novembre 2022

Modifica	Descrizione	Data
CloudWatchReadOnlyAccess: aggiornamento a una policy esistente	CloudWatch ha aggiunto le autorizzazioni a. CloudWatchReadOnlyAccess Le `oam:ListAttachedLinks` autorizzazioni `oam:ListSinks` e sono state aggiunte in modo che gli utenti con questo criterio possano utilizzare la console per visualizzare i dati condivisi dagli account di origine in modo osservabile CloudWatch tra più account.	27 novembre 2022

Modifica	Descrizione	Data
AmazonCloudWatchRUMServiceRolePolicy: aggiornamento a una politica esistente	CloudWatch RUM ha aggiornato una chiave di condizione in AmazonCloudWatchRUM ServiceRolePolicy. La chiave di `"Condition": { "StringEquals": { "cloudwatch:namespace": "AWS/RUM" } }` condizione è stata modificata nella seguente in modo che CloudWatch RUM possa inviare metriche personalizzate a namespace di metriche personalizzate. `"Condition": { "StringLike": { "cloudwatch:namespace": [ "RUM/CustomMetrics/*", "AWS/RUM" ] } }`	2 febbraio 2023

Modifica	Descrizione	Data
AmazonCloudWatchRUM — Politica aggiornata ReadOnlyAccess	CloudWatch ha aggiunto i permessi alla ReadOnlyAccess politica AmazonCloudWatchRUM. Le `rum:BatchGetRumMetricsDefinitions` autorizzazioni `rum:ListRumMetricsDestinations` e sono state aggiunte in modo che CloudWatch RUM possa inviare metriche estese a CloudWatch ed Evidently.	27 ottobre 2022
AmazonCloudWatchRUMServiceRolePolicy: aggiornamento a una politica esistente	CloudWatch RUM ha aggiunto le autorizzazioni al AmazonCloudWatchRUM ServiceRolePolicy. L'`cloudwatch:PutMetricData` autorizzazione è stata aggiunta in modo che CloudWatch RUM possa inviare metriche estese a. CloudWatch	26 ottobre 2022

Modifica	Descrizione	Data
CloudWatchEvidentlyReadOnlyAccess: aggiornamento a una policy esistente	CloudWatch Evidentemente sono state aggiunte autorizzazioni a. CloudWatchEvidentlyReadOnlyAccess Sono state aggiunte le autorizzazioni `evidently:GetSegment` , `evidently:ListSegments` e `evidently:ListSegmentReferences` , in modo che gli utenti con questa policy possano visualizzare i segmenti di destinatari Evidently che sono stati creati.	12 agosto 2022

Modifica	Descrizione	Data
CloudWatchSyntheticsFullAccess: aggiornamento a una policy esistente	CloudWatch Synthetics ha aggiunto le autorizzazioni a. CloudWatchSyntheticsFullAccess Le `lambda:DeleteLayerVersion` autorizzazioni `lambda:DeleteFunction` e sono state aggiunte in modo che CloudWatch Synthetics possa eliminare le risorse correlate quando viene eliminato un canarino. La `iam:ListAttachedRolePolicies` è stata aggiunta in modo che i clienti possano visualizzare le policy collegate al ruolo IAM di Canary.	6 maggio 2022
AmazonCloudWatchRUM — Nuova politica FullAccess	CloudWatch ha aggiunto una nuova politica per consentire la gestione completa del CloudWatch RUM. CloudWatch RUM ti consente di eseguire il monitoraggio reale degli utenti della tua applicazione web. Per ulteriori informazioni, consulta la pagina Usa CloudWatch RUM.	29 novembre 2021

Modifica	Descrizione	Data
AmazonCloudWatchRUMReadOnlyAccess — Nuova politica	CloudWatch ha aggiunto una nuova politica per abilitare l'accesso in sola lettura al CloudWatch RUM. CloudWatch RUM ti consente di eseguire il monitoraggio reale degli utenti della tua applicazione web. Per ulteriori informazioni, consulta la pagina Usa CloudWatch RUM.	29 novembre 2021
CloudWatchEvidentlyFullAccess: nuova policy	CloudWatch ha aggiunto una nuova politica per consentire la gestione completa di CloudWatch Evidently. CloudWatch Evidently ti consente di eseguire esperimenti A/B sulle tue applicazioni web e di implementarle gradualmente. Per ulteriori informazioni, consulta la pagina Esegui lanci ed esperimenti A/B con Evidently CloudWatch .	29 novembre 2021

Modifica	Descrizione	Data
CloudWatchEvidentlyReadOnlyAccess: nuova policy	CloudWatch ha aggiunto una nuova policy per abilitare l'accesso in sola lettura a Evidently. CloudWatch CloudWatch Evidentemente ti consente di eseguire esperimenti A/B sulle tue applicazioni web e di implementarle gradualmente. Per ulteriori informazioni, consulta la pagina Esegui lanci ed esperimenti A/B con Evidently CloudWatch .	29 novembre 2021
AWSServiceRoleForCloudWatchRUM— Nuova politica gestita	CloudWatch ha aggiunto una politica per un nuovo ruolo collegato ai servizi per consentire a CloudWatch RUM di pubblicare i dati di monitoraggio su altri servizi pertinenti. AWS	29 novembre 2021

Modifica	Descrizione	Data
CloudWatchSyntheticsFullAccess: aggiornamento a una policy esistente	CloudWatch Synthetics ha aggiunto le autorizzazioni CloudWatchSyntheticsFullAccesse ha anche modificato l'ambito di un'autorizzazione. L'`kms:ListAliases` autorizzazione è stata aggiunta in modo che gli utenti possano elencare AWS KMS le chiavi disponibili che possono essere utilizzate per crittografare gli artefatti dei canarini. L'autorizzazione `kms:DescribeKey` è stata aggiunta in modo che gli utenti possano vedere i dettagli delle chiavi che verranno utilizzate per crittografare gli artefatti canary. L'autorizzazione `kms:Decrypt` è stata aggiunta per consentire agli utenti di decrittare artefatti canary. Questa capacità di decrittografia è limitata all'utilizzo delle risorse all'interno dei bucket Amazon S3. L'ambito `Resource` del permesso `s3:GetBucketLocation` è stato modificato da `*` a `arn:aws:s3:::*` .	29 settembre 2021

Modifica	Descrizione	Data
CloudWatchSyntheticsFullAccess: aggiornamento a una policy esistente	CloudWatch Synthetics ha aggiunto un'autorizzazione a. CloudWatchSyntheticsFullAccess L'autorizzazione `lambda:UpdateFunctionCode` è stata aggiunta in modo che gli utenti con questa policy possano modificare la versione di runtime dei canary.	20 luglio 2021
AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicy— Nuova politica gestita	CloudWatch ha aggiunto una nuova policy IAM gestita CloudWatch per consentire la creazione di incidenti in AWS Systems Manager Incident Manager.	10 maggio 2021
CloudWatchAutomaticDashboardsAccess: aggiornamento a una policy esistente	CloudWatch ha aggiunto un'autorizzazione alla policy CloudWatchAutomaticDashboardsAccessgestita. L'`synthetics:DescribeCanariesLastRun` autorizzazione è stata aggiunta a questa politica per consentire agli utenti della dashboard con più account di visualizzare i dettagli sulle corse canarie di CloudWatch Synthetics.	20 aprile 2021

Modifica	Descrizione	Data
CloudWatch ha iniziato a tenere traccia delle modifiche	CloudWatch ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	14 aprile 2021

Utilizzo delle chiavi condizionali per limitare l'accesso ai CloudWatch namespace

Utilizza le chiavi di condizione IAM per limitare gli utenti a pubblicare le metriche solo nei CloudWatch namespace che specifichi.

Consentire la pubblicazione in un solo spazio dei nomi

La policy seguente limita l'utente alla pubblicazione dei parametri solo nello spazio dei nomi denominato `MyCustomNamespace`.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "MyCustomNamespace"
            }
        }
    }
}
```

Esclusione della pubblicazione da uno spazio dei nomi

La policy seguente consente all'utente di pubblicare i parametri in qualsiasi spazio dei nomi, ad eccezione di `CustomNamespace2`.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData"
        },
        {
            "Effect": "Deny",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CustomNamespace2"
                }
            }
        }
    ]
}
```

Utilizzo delle chiavi di condizione per limitare l'accesso degli utenti di Contributor Insights ai gruppi di log

Per creare una regola in Contributor Insights e visualizzarne i risultati, un utente deve disporre dell'autorizzazione `cloudwatch:PutInsightRule`. Per impostazione predefinita, un utente con questa autorizzazione può creare una regola di Contributor Insights che valuta qualsiasi gruppo di log in CloudWatch Logs e quindi visualizza i risultati. I risultati possono contenere dati dei collaboratori per tali gruppi di log.

È possibile creare policy IAM con chiavi di condizione per concedere agli utenti l'autorizzazione a scrivere regole di Contributor Insights per alcuni gruppi di log impedendo loro di scrivere regole per e visualizzare questi dati da altri gruppi di log.

Per ulteriori informazioni sull'elemento `Condition` nelle policy IAM, consulta Elementi JSON della policy IAM: Condizione.

Consentire l'accesso alla scrittura di regole e visualizzare i risultati solo per determinati gruppi di log

La policy seguente consente all'utente di accedere alla scrittura di regole e visualizzare i risultati per il gruppo di log denominato `AllowedLogGroup` e tutti i gruppi di log i cui nomi iniziano con `AllowedWildCard`. Non concede l'accesso alla scrittura di regole o alla visualizzazione dei risultati delle regole per altri gruppi di log.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCertainLogGroups",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "AllowedLogGroup",
                        "AllowedWildcard*"
                    ]
                }
            }
        }
    ]
}
```

Negare regole di scrittura per gruppi di log specifici ma consente la scrittura di regole per tutti gli altri gruppi di log

La policy seguente nega esplicitamente all'utente l'accesso per scrivere regole e visualizzare i risultati delle regole per il gruppo di log denominato `ExplicitlyDeniedLogGroup`, ma consente la scrittura di regole e la visualizzazione dei risultati delle regole per tutti gli altri gruppi di log.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowInsightRulesOnLogGroupsByDefault",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*"

        },
        {
            "Sid": "ExplicitDenySomeLogGroups",
            "Effect": "Deny",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
```

```
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "/test/alpine/ExplicitlyDeniedLogGroup"
                    ]
                }
            }
        }
    ]
}
```

Utilizzo dei tasti di condizione per limitare le operazioni di allarme

Quando gli CloudWatch allarmi cambiano stato, possono eseguire diverse azioni come l'arresto e la chiusura delle istanze EC2 e l'esecuzione di azioni di Systems Manager. Queste operazioni possono essere avviate quando l'avviso cambia in qualsiasi stato, inclusi ALARM, OK o INSUFFICIENT_DATA.

Usa la chiave di condizione `cloudwatch:AlarmActions` per consentire a un utente di creare allarmi che possono eseguire solo le operazioni specificate quando lo stato dell'allarme cambia. Ad esempio, è possibile consentire a un utente di creare allarmi che possono eseguire solo operazioni che non sono operazioni EC2.

Consentire a un utente di creare allarmi che possono solo inviare notifiche Amazon SNS o eseguire operazioni di Systems Manager

La seguente policy limita l'utente alla creazione di allarmi che possono solo inviare notifiche Amazon SNS ed eseguire operazioni di Systems Manager. L'utente non può creare allarmi che eseguano le operazioni EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAlarmsThatCanPerformOnlySNSandSSMActions",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricAlarm",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "cloudwatch:AlarmActions": [
```

```
                    "arn:aws:sns:*",
                    "arn:aws:ssm:*"
                ]
            }
        }
    }
  ]
}
```

Utilizzo di ruoli collegati ai servizi per CloudWatch

Amazon CloudWatch utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. CloudWatch I ruoli collegati ai servizi sono predefiniti CloudWatch e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio CloudWatch consente di configurare CloudWatch allarmi in grado di terminare, arrestare o riavviare un'istanza Amazon EC2 senza richiedere l'aggiunta manuale delle autorizzazioni necessarie. Un altro ruolo collegato al servizio consente a un account di monitoraggio di accedere ai CloudWatch dati di altri account da te specificati, per creare dashboard interregionali tra account.

CloudWatch definisce le autorizzazioni di questi ruoli collegati ai servizi e, se non diversamente definito, solo può assumere il ruolo. CloudWatch Le autorizzazioni definite includono la policy di trust e la policy delle autorizzazioni. Una policy delle autorizzazioni specifica non può essere collegata a un'altra entità IAM.

È possibile eliminare i ruoli solo dopo aver eliminato le risorse correlate. Questa restrizione protegge le CloudWatch risorse perché non è possibile rimuovere inavvertitamente le autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta la sezione Servizi AWS che funzionano con IAM e cerca i servizi che riportano Sì nella colonna Ruolo associato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni di ruolo legate al servizio per le azioni EC2 degli allarmi CloudWatch

CloudWatch utilizza il ruolo collegato al servizio denominato AWSServiceRoleForCloudWatchEvents: CloudWatch utilizza questo ruolo collegato al servizio per eseguire azioni di allarme di Amazon EC2.

Il ruolo AWSServiceRoleForCloudWatchEvents collegato al servizio si fida del servizio Events per l'assunzione del ruolo. CloudWatch CloudWatch Events richiama le azioni di terminazione, arresto o riavvio dell'istanza quando viene richiamata dall'allarme.

La politica AWSServiceRoleForCloudWatchEvents di autorizzazione dei ruoli collegati al servizio consente a CloudWatch Events di completare le seguenti azioni sulle istanze Amazon EC2:

- `ec2:StopInstances`
- `ec2:TerminateInstances`
- `ec2:RecoverInstances`
- `ec2:DescribeInstanceRecoveryAttribute`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceStatus`

La politica di autorizzazione dei ruoli AWSServiceRoleForCloudWatchCrossAccountcollegati al servizio consente di completare le seguenti azioni: CloudWatch

- `sts:AssumeRole`

Autorizzazioni di ruolo collegate al servizio per Application Signals CloudWatch

CloudWatch Application Signals utilizza il ruolo collegato al servizio denominato AWSServiceRoleForCloudWatchApplicationSignals: CloudWatch utilizza questo ruolo collegato al servizio per raccogliere dati di CloudWatch log, dati di traccia a raggi X, dati di metrica e dati CloudWatch di tagging dalle applicazioni che hai abilitato per Application Signals. CloudWatch

Il ruolo collegato al AWSServiceRoleForCloudWatchApplicationSignalsservizio prevede che Application Signals assuma il ruolo. CloudWatch Application Signals raccoglie i log, le tracce, i parametri e i dati dei tag dal tuo account.

AWSServiceRoleForCloudWatchApplicationSignalsHa una politica IAM allegata e questa politica è denominata. CloudWatchApplicationSignalsServiceRolePolicy Questa politica concede l'autorizzazione ad CloudWatch Application Signals di raccogliere dati di monitoraggio e etichettatura da altri servizi pertinenti AWS . Include autorizzazioni che consentono ad Application Signals di completare le operazioni seguenti:

- `xray:GetServiceGraph`

- logs:StartQuery
- logs:GetQueryResults
- cloudwatch:GetMetricData
- cloudwatch:ListMetrics
- tag:GetResources

Il contenuto completo di è il CloudWatchApplicationSignalsServiceRolePolicyseguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "XRayPermission",
            "Effect": "Allow",
            "Action": [
                "xray:GetServiceGraph"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWLogsPermission",
            "Effect": "Allow",
            "Action": [
                "logs:StartQuery",
                "logs:GetQueryResults"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/appsignals/*:*",
                "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
```

```
            }
        },
        {
            "Sid": "CWListMetricsPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWGetMetricDataPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "TagsPermission",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
```

```
}
```

Autorizzazioni di ruolo collegate al servizio per le azioni di CloudWatch Systems Manager degli allarmi OpsCenter

CloudWatch utilizza il ruolo collegato al servizio denominato AWSServiceRoleForCloudWatchAlarms_ActionSSM: CloudWatch utilizza questo ruolo collegato al servizio per OpsCenter eseguire azioni di Systems Manager quando un CloudWatch allarme entra nello stato ALARM.

Il ruolo AWSServiceRoleForCloudWatchAlarms_ActionSSM collegato al servizio si fida che il servizio assuma il ruolo. CloudWatch CloudWatch gli allarmi richiamano le OpsCenter azioni di Systems Manager quando vengono richiamati dall'allarme.

La politica AWSServiceRoleForCloudWatchAlarms_ActionSSMdi autorizzazione dei ruoli collegati al servizio consente a Systems Manager di completare le seguenti azioni:

- `ssm:CreateOpsItem`

Autorizzazioni di ruolo collegate ai servizi per le azioni di CloudWatch Systems Manager Incident Manager degli allarmi

CloudWatch utilizza il ruolo collegato al servizio denominato AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents: CloudWatch utilizza questo ruolo collegato al servizio per avviare gli incidenti di Incident Manager quando un allarme entra nello stato ALARM. CloudWatch

Il ruolo AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidentscollegato al servizio si fida che il servizio assuma il ruolo. CloudWatch CloudWatch gli allarmi richiamano l'azione Systems Manager Incident Manager quando vengono richiamati dall'allarme.

La politica AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidentsdi autorizzazione dei ruoli collegati al servizio consente a Systems Manager di completare le seguenti azioni:

- `ssm-incidents:StartIncident`

Autorizzazioni di ruolo collegate al servizio per più account e più regioni CloudWatch

CloudWatch utilizza il ruolo collegato al servizio denominato AWSServiceRoleForCloudWatchCrossAccount: CloudWatch utilizza questo ruolo per accedere ai CloudWatch dati di altri account specificati dall'utente. AWS La SLR fornisce solo l'autorizzazione ad assumere il ruolo per consentire al CloudWatch servizio di assumere il ruolo nell'account di condivisione. È il ruolo di condivisione che fornisce l'accesso ai dati.

La politica AWSServiceRoleForCloudWatchCrossAccountdi autorizzazione dei ruoli collegati al servizio consente di CloudWatch completare le seguenti azioni:

- `sts:AssumeRole`

Il ruolo AWSServiceRoleForCloudWatchCrossAccountcollegato al servizio si fida che il servizio assuma il CloudWatch ruolo.

Autorizzazioni di ruolo collegate ai servizi per il database Performance Insights CloudWatch

CloudWatch utilizza il ruolo collegato al servizio denominato. AWSServiceRoleForCloudWatchMetrics_DbPerfInsights — CloudWatch utilizza questo ruolo per recuperare le metriche di Performance Insights per la creazione di allarmi e istantanee.

Al ruolo collegato al AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsservizio è associata la policy IAM. `AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy` I contenuti di questa policy sono i seguenti:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "pi:GetResourceMetrics"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
    }
   }
```

```
  }
 ]
}
```

Il ruolo AWSServiceRoleForCloudWatchMetrics_DbPerfInsightscollegato al servizio si fida che il CloudWatch servizio assuma il ruolo.

Creazione di un ruolo collegato al servizio per CloudWatch

Non è necessario creare manualmente nessuno di questi ruoli collegati ai servizi. La prima volta che crei un allarme in AWS Management Console, l'IAM CLI o l'API IAM CloudWatch crea AWSServiceRoleForCloudWatchEvents e AWSServiceRoleForCloudWatchAlarms_ActionSSMper te.

La prima volta che abiliti il rilevamento di servizi e topologie, Application Signals crea AWSServiceRoleForCloudWatchApplicationSignalsper te.

Quando abiliti per la prima volta un account come account di monitoraggio per la funzionalità interregionale tra più account, lo CloudWatch crea AWSServiceRoleForCloudWatchCrossAccountautomaticamente.

Quando crei per la prima volta un allarme che utilizza la funzione matematica `DB_PERF_INSIGHTS` metrica, CloudWatch lo crea per te. AWSServiceRoleForCloudWatchMetrics_DbPerfInsights

Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente IAM.

Modifica di un ruolo collegato al servizio per CloudWatch

CloudWatch non consente di modificare i ruoli AWSServiceRoleForCloudWatchEvents, AWSServiceRoleForCloudWatchAlarms_ActionSSMAWSServiceRoleForCloudWatchCrossAccount, o AWSServiceRoleForCloudWatchMetrics_DbPerfInsights. Dopo aver creato questi ruoli, non è possibile modificarne i nomi perché varie entità potrebbero fare riferimento a tali ruoli. Puoi tuttavia modificare la descrizione dei ruoli utilizzando IAM.

Modifica della descrizione di un ruolo collegato ai servizi (console IAM)

È possibile utilizzare la console IAM per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (console)

1. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli).
2. Scegliere il nome del ruolo da modificare.

3. Nella parte destra di Role description (Descrizione ruolo), scegliere Edit (Modifica).
4. Digita una nuova descrizione nella casella e scegli Save (Salva).

Modifica della descrizione di un ruolo collegato ai servizi (AWS CLI)

Puoi utilizzare i comandi IAM di AWS Command Line Interface per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato ai servizi (AWS CLI)

1. (Facoltativo) Per visualizzare la descrizione attuale di un ruolo, utilizza i seguenti comandi:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Per fare riferimento ai ruoli con i comandi AWS CLI , utilizza il nome del ruolo, non l'ARN. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **`myrole`**.
2. Per aggiornare la descrizione di un ruolo collegato ai servizi, utilizza il seguente comando:

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

Modifica della descrizione di un ruolo collegato ai servizi (API IAM)

È possibile utilizzare l'API IAM per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (API)

1. (Facoltativo) Per visualizzare la descrizione attuale di un ruolo, utilizza il seguente comando:

   GetRole
2. Per aggiornare la descrizione di un ruolo, utilizza il seguente comando:

   UpdateRoleDescription

Eliminazione di un ruolo collegato al servizio per CloudWatch

Se non hai più allarmi che interrompono, terminano o riavviano automaticamente le istanze EC2, ti consigliamo di eliminare il ruolo. AWSServiceRoleForCloudWatchEvents

Se non hai più allarmi che eseguono OpsCenter azioni di Systems Manager, ti consigliamo di eliminare il AWSServiceRoleForCloudWatchAlarms_ActionSSM ruolo.

Se elimini tutti gli allarmi che utilizzano la funzione matematica `DB_PERF_INSIGHTS` metrica, ti consigliamo di eliminare il ruolo collegato al servizio. AWSServiceRoleForCloudWatchMetrics_DbPerfInsights

In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare.

Pulizia di un ruolo collegato ai servizi

Prima di utilizzare IAM per eliminare un ruolo collegato ai servizi, devi innanzitutto verificare che il ruolo non abbia sessioni attive ed eliminare tutte le risorse utilizzate dal ruolo.

Per verificare se il ruolo collegato ai servizi dispone di una sessione attiva nella console IAM

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli. Scegli il nome (non la casella di controllo) del ruolo. AWSServiceRoleForCloudWatchEvents
3. Nella pagina Summary (Riepilogo) per il ruolo selezionato, seleziona Access Advisor (Consulente accessi) ed esamina l'attività recente per il ruolo collegato ai servizi.

> **Note**
>
> Se non sei sicuro che CloudWatch stia utilizzando il AWSServiceRoleForCloudWatchEvents ruolo, prova a eliminarlo. Se il servizio sta utilizzando il ruolo, l'eliminazione non andrà a buon fine e potrai visualizzare le regioni in cui il ruolo viene utilizzato. Se il ruolo è in uso, prima di poterlo eliminare dovrai attendere il termine della sessione. Non puoi revocare la sessione per un ruolo collegato al servizio.

Eliminazione di un ruolo collegato ai servizi (console IAM)

È possibile utilizzare la console IAM per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (console)

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli. Selezionare la casella di controllo accanto al nome del ruolo che desideri eliminare, non il nome o la riga stessa.
3. In Role actions (Operazioni per ruolo), seleziona Delete role (Elimina ruolo).
4. Nella finestra di dialogo di conferma controlla i dati relativi all'ultimo accesso ai servizi, che indicano quando ognuno dei ruoli selezionati ha effettuato l'accesso a un servizio AWS . In questo modo potrai verificare se il ruolo è attualmente attivo. Per procedere, seleziona Yes, Delete (Sì, elimina).
5. Controlla le notifiche della console IAM per monitorare lo stato dell'eliminazione del ruolo collegato ai servizi. Poiché l'eliminazione del ruolo collegato ai servizi IAM è asincrona, una volta richiesta l'eliminazione del ruolo, il task di eliminazione può essere eseguito correttamente o meno. Se il task non viene eseguito correttamente, seleziona View details (Visualizza dettagli) o View Resources (Visualizza risorse) dalle notifiche per capire perché l'eliminazione non è stata effettuata. Se l'eliminazione non viene eseguita perché vi sono risorse nel servizio che sono usate dal ruolo, il motivo dell'errore include un elenco di risorse.

Eliminazione del ruolo collegato ai servizi (AWS CLI)

Puoi utilizzare i comandi IAM di AWS Command Line Interface per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato ai servizi (AWS CLI)

1. Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `deletion-task-id`dalla risposta per controllare lo stato del task di eliminazione. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, digita il seguente comando:

```
$ aws iam delete-service-linked-role --role-name service-linked-role-name
```

2. Digita il seguente comando per verificare lo stato del task di eliminazione:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema.

Eliminazione di un ruolo collegato ai servizi (API IAM)

È possibile utilizzare l'API di IAM; per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (API)

1. Per inviare una richiesta di eliminazione per un ruolo collegato al servizio, chiama. DeleteServiceLinkedRole Nella richiesta specificare il nome del ruolo che si desidera eliminare.

   Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `DeletionTaskId`dalla risposta per controllare lo stato del task di eliminazione.
2. Per verificare lo stato dell'eliminazione, chiama. GetServiceLinkedRoleDeletionStatus Nella richiesta, specificare il `DeletionTaskId`.

   Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema.

CloudWatch aggiornamenti ai ruoli AWS collegati al servizio

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite CloudWatch da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei CloudWatch documenti.

Modifica	Descrizione	Data
AWSServiceRoleForCloudWatchApplicationSignals — Aggiornamento delle	CloudWatch aggiungere altri gruppi di log all'ambito e alle `logs:GetQueryResults`	24 aprile 2024

Modifica	Descrizione	Data
autorizzazioni della politica relativa ai ruoli collegati ai servizi	autorizzazioni `logs:StartQuery` concesse da questo ruolo.	
AWSServiceRoleForCloudWatchApplicationSignals — Nuovo ruolo collegato ai servizi	CloudWatch ha aggiunto questo nuovo ruolo collegato al servizio per consentire ad CloudWatch Application Signals di raccogliere dati di CloudWatch log, dati di traccia a raggi X, dati di CloudWatch metrica e dati di etichettatura dalle applicazioni abilitate per Application Signals. CloudWatch	9 novembre 2023
AWSServiceRoleForCloudWatchMetrics_DbPerfInsights— Nuovo ruolo collegato ai servizi	CloudWatch ha aggiunto questo nuovo ruolo collegato al servizio per consentire di recuperare le metriche CloudWatch di Performance Insights per allarmi e snapshot. A questo ruolo è associata una policy IAM che concede l'autorizzazione CloudWatch a recuperare le metriche di Performance Insights per tuo conto.	13 settembre 2023

Modifica	Descrizione	Data
AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents— Nuovo ruolo collegato ai servizi	CloudWatch ha aggiunto un nuovo ruolo collegato al servizio per consentire CloudWatch la creazione di incidenti in Incident Manager. AWS Systems Manager	26 Aprile 2021
CloudWatch ha iniziato a tenere traccia delle modifiche	CloudWatch ha iniziato a tenere traccia delle modifiche per i ruoli collegati ai servizi.	26 Aprile 2021

Utilizzo di ruoli collegati ai servizi per RUM CloudWatch

CloudWatch RUM utilizza un ruolo collegato al servizio AWS Identity and Access Management (IAM). Un ruolo collegato al servizio è un tipo di ruolo IAM univoco collegato direttamente a RUM. Il ruolo collegato al servizio è predefinito da RUM e include tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per conto dell'utente. AWS

RUM definisce le autorizzazioni del ruolo collegato al servizio e, salvo diversamente definito, solo RUM potrà assumere tale ruolo. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Puoi eliminare il ruolo solo dopo aver rimosso le risorse correlate. Questa limitazione protegge le risorse RUM poiché impedisce la rimozione involontaria delle autorizzazioni ad accedere alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati al servizio, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Sì nella colonna Ruoli collegati al servizio. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per RUM

RUM utilizza il ruolo collegato al servizio denominato AWSServiceRoleForCloudWatchRUM: questo ruolo consente a RUM di inviare dati di AWS X-Ray traccia al tuo account, per i monitor delle app per cui abiliti il tracciamento X-Ray.

Il ruolo AWSServiceRoleForCloudWatchRUMcollegato al servizio si fida che il servizio X-Ray assuma il ruolo. X-Ray invia i dati di traccia al tuo account.

Al ruolo AWSServiceRoleForCloudWatchRUMcollegato al servizio è associata una policy IAM denominata RUM. AmazonCloudWatch ServiceRolePolicy Questa politica concede il permesso a CloudWatch RUM di pubblicare i dati di monitoraggio su altri servizi pertinenti. AWS Include autorizzazioni che consentono a RUM di completare le operazioni seguenti:

- `xray:PutTraceSegments`
- `cloudwatch:PutMetricData`

Il contenuto completo di AmazonCloudWatchRUM ServiceRolePolicy è il seguente.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
      "AWS/RUM"
     ]
    }
   }
  }
 ]
}
```

Creazione di un ruolo collegato ai servizi per RUM

Non è necessario creare manualmente il ruolo collegato al servizio per CloudWatch RUM. La prima volta che crei un monitor di app con il tracciamento a raggi X abilitato o aggiorni un monitor di app per utilizzare il tracciamento a raggi X, RUM crea per te. AWSServiceRoleForCloudWatchRUM

Per ulteriori informazioni, consulta Creazione di un ruolo collegato ai servizi nella Guida per l'utente IAM.

Modifica di un ruolo collegato ai servizi per RUM

CloudWatch RUM non consente di modificare il ruolo. AWSServiceRoleForCloudWatchRUM Dopo aver creato questi ruoli, non è possibile modificarne i nomi perché varie entità potrebbero fare riferimento a tali ruoli. Puoi tuttavia modificare la descrizione dei ruoli utilizzando IAM.

Modifica della descrizione di un ruolo collegato ai servizi (console IAM)

È possibile utilizzare la console IAM per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (console)

1. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli).
2. Scegliere il nome del ruolo da modificare.
3. Nella parte destra di Role description (Descrizione ruolo), scegliere Edit (Modifica).
4. Digita una nuova descrizione nella casella e scegli Save (Salva).

Modifica della descrizione di un ruolo collegato ai servizi (AWS CLI)

È possibile utilizzare i comandi IAM di AWS Command Line Interface per modificare la descrizione di un ruolo collegato al servizio.

Per modificare la descrizione di un ruolo collegato ai servizi (AWS CLI)

1. (Facoltativo) Per visualizzare la descrizione attuale di un ruolo, utilizza i seguenti comandi:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Per fare riferimento ai ruoli con i comandi AWS CLI , utilizza il nome del ruolo, non l'ARN. Ad esempio, per fare riferimento a un ruolo il cui ARN è `arn:aws:iam::123456789012:role/myrole`, puoi usare **`myrole`**.

2. Per aggiornare la descrizione di un ruolo collegato ai servizi, utilizza il seguente comando:

```
$ aws iam update-role-description --role-name role-name --description description
```

Modifica della descrizione di un ruolo collegato ai servizi (API IAM)

È possibile utilizzare l'API IAM per modificare la descrizione di un ruolo collegato ai servizi.

Per modificare la descrizione di un ruolo collegato ai servizi (API)

1. (Facoltativo) Per visualizzare la descrizione attuale di un ruolo, utilizza il seguente comando:

   GetRole
2. Per aggiornare la descrizione di un ruolo, utilizza il seguente comando:

   UpdateRoleDescription

Eliminazione di un ruolo collegato ai servizi per RUM

Se non disponi più di monitor per app con X-Ray abilitato, ti consigliamo di eliminare AWSServiceRoleForCloudWatchRUMil ruolo.

In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare.

Pulizia di un ruolo collegato ai servizi

Prima di utilizzare IAM per eliminare un ruolo collegato ai servizi, devi innanzitutto verificare che il ruolo non abbia sessioni attive ed eliminare tutte le risorse utilizzate dal ruolo.

Per verificare se il ruolo collegato ai servizi dispone di una sessione attiva nella console IAM

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli. Scegli il nome (non la casella di controllo) del AWSServiceRoleForCloudWatchRUMruolo.
3. Nella pagina Summary (Riepilogo) per il ruolo selezionato, seleziona Access Advisor (Consulente accessi) ed esamina l'attività recente per il ruolo collegato ai servizi.

> **Note**
>
> Se non sei sicuro che RUM stia utilizzando il AWSServiceRoleForCloudWatchRUMruolo, prova a eliminarlo. Se il servizio sta utilizzando il ruolo, l'eliminazione non andrà a buon fine e potrai visualizzare le regioni in cui il ruolo viene utilizzato. Se il ruolo è in uso, prima di poterlo eliminare dovrai attendere il termine della sessione. Non puoi revocare la sessione per un ruolo collegato al servizio.

Eliminazione di un ruolo collegato ai servizi (console IAM)

È possibile utilizzare la console IAM per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (console)

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli. Selezionare la casella di controllo accanto al nome del ruolo che desideri eliminare, non il nome o la riga stessa.
3. In Role actions (Operazioni per ruolo), seleziona Delete role (Elimina ruolo).
4. Nella finestra di dialogo di conferma controlla i dati relativi all'ultimo accesso ai servizi, che indicano quando ognuno dei ruoli selezionati ha effettuato l'accesso a un servizio AWS . In questo modo potrai verificare se il ruolo è attualmente attivo. Per procedere, seleziona Yes, Delete (Sì, elimina).
5. Controlla le notifiche della console IAM per monitorare lo stato dell'eliminazione del ruolo collegato ai servizi. Poiché l'eliminazione del ruolo collegato ai servizi IAM è asincrona, una volta richiesta l'eliminazione del ruolo, il task di eliminazione può essere eseguito correttamente o meno. Se il task non viene eseguito correttamente, seleziona View details (Visualizza dettagli) o View Resources (Visualizza risorse) dalle notifiche per capire perché l'eliminazione non è stata effettuata. Se l'eliminazione non viene eseguita perché vi sono risorse nel servizio che sono usate dal ruolo, il motivo dell'errore include un elenco di risorse.

Eliminazione del ruolo collegato ai servizi (AWS CLI)

Puoi utilizzare i comandi IAM di AWS Command Line Interface per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato ai servizi (AWS CLI)

1. Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `deletion-task-id`dalla risposta per controllare lo stato del task di eliminazione. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, digita il seguente comando:

   ```
   $ aws iam delete-service-linked-role --role-name service-linked-role-name
   ```

2. Digita il seguente comando per verificare lo stato del task di eliminazione:

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema.

Eliminazione di un ruolo collegato ai servizi (API IAM)

È possibile utilizzare l'API di IAM; per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (API)

1. Per inviare una richiesta di eliminazione per un ruolo collegato al servizio, chiama. DeleteServiceLinkedRole Nella richiesta specificare il nome del ruolo che si desidera eliminare.

   Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `DeletionTaskId`dalla risposta per controllare lo stato del task di eliminazione.

2. Per verificare lo stato dell'eliminazione, chiama. GetServiceLinkedRoleDeletionStatus Nella richiesta, specificare il `DeletionTaskId`.

   Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema.

Utilizzo di ruoli collegati ai servizi per CloudWatch Application Insights

CloudWatch Application Insights utilizza ruoli AWS Identity and Access Management collegati ai servizi (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente ad Application Insights. CloudWatch I ruoli collegati ai servizi sono predefiniti da CloudWatch Application Insights e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per conto dell'utente. AWS

Un ruolo collegato ai servizi semplifica la configurazione di CloudWatch Application Insights perché non è necessario aggiungere manualmente le autorizzazioni necessarie. CloudWatch Application Insights definisce le autorizzazioni dei suoi ruoli collegati ai servizi e, se non diversamente definito, solo CloudWatch Application Insights può assumerne i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consultare Servizi AWS che funzionano con IAM e cercare i servizi che riportano Sì nella colonna Ruolo associato ai servizi. Scegli un collegamento Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni di ruolo collegate al servizio per Application Insights CloudWatch

CloudWatch Application Insights utilizza il ruolo collegato al servizio denominato. AWSServiceRoleForApplicationInsights Application Insights utilizza questo ruolo per eseguire operazioni come l'analisi dei gruppi di risorse del cliente, la creazione di CloudFormation stack per creare allarmi sulle metriche e la configurazione dell'agente sulle istanze EC2. CloudWatch Questo ruolo collegato ai servizi dispone di una policy IAM collegata denominata `CloudwatchApplicationInsightsServiceLinkedRolePolicy`. Per gli aggiornamenti a questa policy, consulta Aggiornamenti a policy gestite da AWS per Application Insights.

La politica di autorizzazione dei ruoli consente ad CloudWatch Application Insights di completare le seguenti azioni sulle risorse.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmHistory",
```

```
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:PutAnomalyDetector",
        "cloudwatch:DeleteAnomalyDetector",
        "cloudwatch:DescribeAnomalyDetectors"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:CreateStack",
        "cloudFormation:UpdateStack",
        "cloudFormation:DeleteStack",
        "cloudFormation:DescribeStackResources"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/ApplicationInsights-*"
```

```
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:DescribeStacks",
        "cloudFormation:ListStackResources",
        "cloudFormation:ListStacks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:ListGroupResources",
        "resource-groups:GetGroupQuery",
        "resource-groups:GetGroup"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:DeleteGroup"
      ],
      "Resource": [
        "arn:aws:resource-groups:*:*:group/ApplicationInsights-*"
      ]
    },
    {
```

```
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter",
        "ssm:DeleteParameter",
        "ssm:AddTagsToResource",
        "ssm:RemoveTagsFromResource",
        "ssm:GetParameters"
      ],
      "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-ApplicationInsights-*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:CreateAssociation",
        "ssm:UpdateAssociation",
        "ssm:DeleteAssociation",
        "ssm:DescribeAssociation"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:association/*",
        "arn:aws:ssm:*:*:managed-instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure",
```

```
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:CreateOpsItem",
        "ssm:DescribeOpsItems",
        "ssm:UpdateOpsItem",
        "ssm:DescribeInstanceInformation"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:AddTagsToResource"
      ],
      "Resource": "arn:aws:ssm:*:*:opsitem/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommandInvocations",
        "ssm:GetCommandInvocation"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-CheckPerformanceCounterSets",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AWSEC2-DetectWorkload",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeNatGateways"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:ListFunctions",
        "lambda:GetFunctionConfiguration",
        "lambda:ListEventSourceMappings"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:DeleteRule"
      ],
```

```
      "Resource": [
        "arn:aws:events:*:*:rule/AmazonCloudWatch-ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "xray:GetServiceGraph",
        "xray:GetTraceSummaries",
        "xray:GetTimeSeriesServiceStatistics",
        "xray:GetTraceGraph"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:ListTables",
        "dynamodb:DescribeTable",
        "dynamodb:DescribeContributorInsights",
        "dynamodb:DescribeTimeToLive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:DescribeScalableTargets"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetMetricsConfiguration",
        "s3:GetReplicationConfiguration"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "states:ListStateMachines",
        "states:DescribeExecution",
        "states:DescribeStateMachine",
        "states:GetExecutionHistory"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:DescribeServices",
        "ecs:DescribeTaskDefinition",
        "ecs:DescribeTasks",
        "ecs:DescribeTaskSets",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "ecs:ListTasks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
```

```
      "Effect": "Allow",
      "Action": [
        "ecs:UpdateClusterSettings"
      ],
      "Resource": [
        "arn:aws:ecs:*:*:cluster/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:DescribeFargateProfile",
        "eks:DescribeNodegroup",
        "eks:ListClusters",
        "eks:ListFargateProfiles",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:GetSubscriptionAttributes",
        "sns:GetTopicAttributes",
        "sns:GetSMSAttributes",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ListQueues"
      ],
      "Resource": "*"
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "logs:DeleteSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*",
        "arn:aws:logs:*:*:destination:AmazonCloudWatch-ApplicationInsights-
LogIngestionDestination*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53:GetHostedZone",
        "route53:GetHealthCheck",
        "route53:ListHostedZones",
        "route53:ListHealthChecks",
        "route53:ListQueryLoggingConfigs"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "route53resolver:ListFirewallRuleGroupAssociations",
        "route53resolver:GetFirewallRuleGroup",
        "route53resolver:ListFirewallRuleGroups",
        "route53resolver:ListResolverEndpoints",
        "route53resolver:GetResolverQueryLogConfig",
        "route53resolver:ListResolverQueryLogConfigs",
        "route53resolver:ListResolverQueryLogConfigAssociations",
        "route53resolver:GetResolverEndpoint",
        "route53resolver:GetFirewallRuleGroupAssociation"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato al servizio per Application Insights CloudWatch

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei una nuova applicazione Application Insights in AWS Management Console, CloudWatch Application Insights crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato ai servizi e desideri ricrearlo, puoi utilizzare lo stesso processo per ricreare il ruolo nel tuo account. Quando crei una nuova applicazione Application Insights, CloudWatch Application Insights crea nuovamente il ruolo collegato al servizio per te.

Modifica di un ruolo collegato al servizio per Application Insights CloudWatch

CloudWatch Application Insights non consente di modificare il ruolo collegato al AWSServiceRoleForApplicationInsights servizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato al servizio per Application Insights CloudWatch

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo si evita di avere un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario eliminare tutte le applicazioni in Application Insights prima di poter eliminare manualmente il ruolo.

> **Note**
>
> Se il servizio CloudWatch Application Insights utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse di CloudWatch Application Insights utilizzate da AWSServiceRoleForApplicationInsights

- Eliminare tutte le applicazioni CloudWatch Application Insights. Per ulteriori informazioni, consulta «Eliminazione delle applicazioni» nella Guida per l'utente di CloudWatch Application Insights.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al AWSServiceRoleForApplicationInsights servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di CloudWatch Application Insights

CloudWatch Application Insights supporta l'utilizzo di ruoli collegati ai servizi in tutte le AWS regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta CloudWatch Application Insights Regions and Endpoints.

AWS politiche gestite per Amazon CloudWatch Application Insights

Una policy AWS gestita è una policy autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWSnella Guida per l'utente di IAM.

AWS politica gestita: CloudWatchApplicationInsightsFullAccess

È possibile allegare la policy `CloudWatchApplicationInsightsFullAccess`alle identità IAM.

Questa policy concede le autorizzazioni amministrative che consentono l'accesso completo alla funzionalità di Application Insights.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `applicationinsights`: consente l'accesso completo alle funzionalità di Application Insights.
- `iam`— Consente ad Application Insights di creare il ruolo collegato al servizio,. AWSServiceRoleForApplicationInsights Ciò è necessario affinché Application Insights possa eseguire operazioni come l'analisi dei gruppi di risorse di un cliente, la creazione di CloudFormation stack per creare allarmi sulle metriche e la configurazione dell' CloudWatchagente sulle istanze EC2. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per CloudWatch Application Insights.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "applicationinsights:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "sqs:ListQueues",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "autoscaling:DescribeAutoScalingGroups",
        "lambda:ListFunctions",
        "dynamodb:ListTables",
        "s3:ListAllMyBuckets",
        "sns:ListTopics",
        "states:ListStateMachines",
        "apigateway:GET",
        "ecs:ListClusters",
        "ecs:DescribeTaskDefinition",
        "ecs:ListServices",
        "ecs:ListTasks",
        "eks:ListClusters",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "logs:DescribeLogGroups",
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
```

```
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/application-insights.amazonaws.com/
AWSServiceRoleForApplicationInsights"
      ],
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "application-insights.amazonaws.com"
        }
      }
    }
  ]
}
```

AWS politica gestita: CloudWatchApplicationInsightsReadOnlyAccess

È possibile allegare la policy `CloudWatchApplicationInsightsReadOnlyAccess`alle identità IAM.

Questa policy concede le autorizzazioni amministrative che consentono l'accesso in sola lettura a tutte le funzionalità di Application Insights.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `applicationinsights`: consente di accedere in sola lettura alle funzionalità di Application Insights.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "applicationinsights:Describe*",
                "applicationinsights:List*"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politica gestita: CloudwatchApplicationInsightsServiceLinkedRolePolicy

Non puoi collegarti CloudwatchApplicationInsightsServiceLinkedRolePolicy alle tue entità IAM. Questa policy è associata a un ruolo collegato al servizio che consente ad Application Insights di monitorare le risorse dei clienti. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per CloudWatch Application Insights.

Aggiornamenti a policy gestite da AWS per Application Insights

Visualizza i dettagli sugli aggiornamenti alle policy AWS gestite per Application Insights da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivi il feed RSS nella pagina Cronologia dei documenti di Application Insights.

Modifica	Descrizione	Data
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni alle CloudFormation pile di elenchi. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di analizzare e monitorare AWS le risorse annidate nello CloudFormation stack.	24 aprile 2023

Modifica	Descrizione	Data
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Approfondimenti sulle applicazioni ha aggiunto nuove autorizzazioni per ottenere un elenco delle risorse di Amazon VPC e Route 53. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di configurare automaticamente il monitoraggio della rete basato sulle best practice. Amazon CloudWatch	23 gennaio 2023
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Approfondimenti sulle applicazioni ha aggiunto nuove autorizzazioni per ottenere i risultati delle chiamate al comando SSM. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di rilevare e monitorare automaticamente i carichi di lavoro in esecuzione su istanze Amazon EC2.	19 dicembre 2022

Modifica	Descrizione	Data
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Approfondimenti sulle applicazioni ha aggiunto nuove autorizzazioni per descrivere le risorse di Amazon VPC e Route 53. Queste autorizzazioni sono necessarie per consentir e ad Amazon CloudWatch Application Insights di leggere le configurazioni delle risorse Amazon VPC e Route 53 dei clienti e di aiutare i clienti a configurare automaticamente il monitoraggio della rete basato sulle best practice. Amazon CloudWatch	19 dicembre 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per descrivere le risorse EFS. Queste autorizzazioni sono necessarie per consentir e ad Amazon CloudWatch Application Insights di leggere le configurazioni delle risorse dei clienti di Amazon EFS e di aiutare i clienti a configura re automaticamente le migliori pratiche per il monitoraggio EFS. CloudWatch	3 ottobre 2022

Modifica	Descrizione	Data
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Approfondimenti sulle applicazioni ha aggiunto nuove autorizzazioni per descrivere il file system EFS. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di creare applicazioni basate su account interrogando tutte le risorse supportate in un account.	3 ottobre 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Approfondimenti sulle applicazioni ha aggiunto nuove autorizzazioni per richiamare informazioni sulle risorse FSx. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di monitorare i carichi di lavoro recuperando informazioni sufficienti sui volumi FSx sottostanti.	12 settembre 2022

Modifica	Descrizione	Data
AWS politica gestita: CloudWatchApplicationInsightsFullAccess: aggiornamento a una policy esistente	Application Insights ha aggiunto una nuova autorizzazione per descrivere i gruppi di log. Queste autorizzazioni sono necessarie per Amazon CloudWatch Application Insights per garantire che le autorizzazioni corrette per il monitoraggio dei gruppi di log siano disponibili in un account quando si crea una nuova applicazione.	24 gennaio 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per creare ed eliminare i filtri di iscrizione ai CloudWatch registri. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di creare filtri di abbonamento per facilitare il monitoraggio dei log delle risorse all'interno delle applicazioni configurate.	24 gennaio 2022

Modifica	Descrizione	Data
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per descrivere i gruppi target e lo stato di destinazione per Elastic Load Balancer. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di creare applicazioni basate su account interrogando tutte le risorse supportate in un account.	4 novembre 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per eseguire il Documento SSM di `AmazonCloudWatch-ManageAgent` sulle istanze Amazon EC2. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di pulire i file di configurazione degli CloudWatch agenti creati da Application Insights.	30 settembre 2021

Modifica	Descrizione	Data
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per supportare il monitoraggio delle applicazioni dell'account per inserire e monitorare tutte le risorse supportate nel tuo account. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di interrogare, etichettare le risorse e creare gruppi per queste risorse. Application Insights ha aggiunto nuove autorizzazioni per supportare il monitoraggio degli argomenti SNS. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di raccogliere metadati dalle risorse SNS per configurare il monitoraggio per argomenti SNS.	15 settembre 2021

Modifica	Descrizione	Data
AWS politica gestita: CloudWatchApplicationInsightsFullAccess: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per descrivere ed elencare le risorse supportate. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di creare applicazioni basate su account interrogando tutte le risorse supportate in un account.	15 settembre 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per descrivere le risorse FSx. Queste autorizzazioni sono necessarie per consentire ad Amazon CloudWatch Application Insights di leggere le configurazioni delle risorse FSx dei clienti e per aiutare i clienti a configurare automaticamente il monitoraggio FSx basato sulle best practice. CloudWatch	31 agosto 2021

Modifica	Descrizione	Data
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per descrivere ed elencare le risorse del servizio ECS ed EKS. Questa autorizzazione è necessaria per consentire ad Amazon CloudWatch Application Insights di leggere la configurazione delle risorse dei container dei clienti e per aiutare i clienti a configurare automaticamente il monitoraggio dei container con le migliori pratiche CloudWatch.	18 maggio 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy: aggiornamento a una policy esistente	Application Insights ha aggiunto nuove autorizzazioni per consentire OpsCenter di etichettare OpsItems l'`ssm:AddTagsToResource` azione sulle risorse con il tipo di `opsitem` risorsa. Questa autorizzazione è richiesta da OpsCenter. Amazon CloudWatch Application Insights crea OpsItems soluzioni che consentono al cliente di risolvere i problemi utilizzando AWS SSM. OpsCenter	13 aprile 2021

Modifica	Descrizione	Data
Application Insights ha iniziato a monitorare le modifiche	Application Insights ha iniziato a tenere traccia delle modifiche alle sue politiche AWS gestite.	13 aprile 2021

Riferimento alle CloudWatch autorizzazioni Amazon

La tabella seguente elenca ogni operazione CloudWatch API e le azioni corrispondenti per le quali è possibile concedere le autorizzazioni per eseguire l'azione. Puoi specificare le operazioni nel campo della policy `Action` e un carattere jolly (*) come valore della risorsa nel campo della policy `Resource`.

Puoi utilizzare i tasti AWS-wide condition nelle tue CloudWatch politiche per esprimere condizioni. Per un elenco completo delle chiavi AWS-wide, consulta AWS Global and IAM Condition Context Keys nella IAM User Guide.

> **Note**
>
> Per specificare un'operazione, utilizza il prefisso `cloudwatch:` seguito dal nome dell'operazione API. Ad esempio: `cloudwatch:GetMetricData`, `cloudwatch:ListMetrics` o `cloudwatch:*` (per tutte le operazioni CloudWatch).

Argomenti

- CloudWatch Operazioni API e autorizzazioni richieste per le azioni
- CloudWatch Operazioni dell'API Contributor Insights e autorizzazioni richieste per le azioni
- CloudWatch Operazioni dell'API Events e autorizzazioni richieste per le azioni
- CloudWatch Registra le operazioni API e le autorizzazioni richieste per le azioni
- Operazioni API Amazon EC2 e autorizzazioni necessarie per le operazioni
- Operazioni API Amazon EC2 Auto Scaling e autorizzazioni necessarie per le operazioni

CloudWatch Operazioni API e autorizzazioni richieste per le azioni

CloudWatch Operazioni API	Autorizzazioni necessarie (operazioni API)
DeleteAlarms	`cloudwatch:DeleteAlarms` Necessario per eliminare un allarme.
DeleteDashboards	`cloudwatch:DeleteDashboards` Necessario per eliminare un pannello di controllo.
DeleteMetricStream	`cloudwatch:DeleteMetricStream` Necessario per eliminare un flusso di parametri.
DescribeAlarmHistory	`cloudwatch:DescribeAlarmHistory` Necessario per visualizzare la cronologia allarmi. Per recuperare informazioni sugli allarmi compositi, l'autorizzazione `cloudwatch:DescribeAlarmHistory` deve avere un ambito `*`. Non è possibile restituire informazioni sugli allarmi compositi se l'autorizzazione `cloudwatch:DescribeAlarmHistory` ha un ambito più limitato.
DescribeAlarms	`cloudwatch:DescribeAlarms` Necessario per recuperare informazioni sugli allarmi. Per recuperare informazioni sugli allarmi compositi, l'autorizzazione `cloudwatch:DescribeAlarms` deve avere un ambito

CloudWatch Operazioni API	Autorizzazioni necessarie (operazioni API)
	*. Non è possibile restituire informazioni sugli allarmi compositi se l'autorizzazione `cloudwatch:DescribeAlarms` ha un ambito più limitato.
DescribeAlarmsForMetric	`cloudwatch:DescribeAlarmsForMetric` Necessario per visualizzare gli allarmi per una metrica.
DisableAlarmActions	`cloudwatch:DisableAlarmActions` Necessario per disattivare un'operazione di allarme.
EnableAlarmActions	`cloudwatch:EnableAlarmActions` Necessario per attivare un'operazione di allarme.
GetDashboard	`cloudwatch:GetDashboard` Necessario per visualizzare i dati sui pannelli di controllo esistenti.
GetMetricData	`cloudwatch:GetMetricData` Necessario per rappresentare graficamente i dati metrici nella CloudWatch console, recuperare grandi quantità di dati metrici ed eseguire calcoli metrici su tali dati.

CloudWatch Operazioni API	Autorizzazioni necessarie (operazioni API)
GetMetricStatistics	`cloudwatch:GetMetricStatistics` Necessario per visualizzare i grafici in altre parti della console e nei widget della dashboard. CloudWatch
GetMetricStream	`cloudwatch:GetMetricStream` Necessario per visualizzare le informazioni su un flusso di parametri.
GetMetricWidgetImage	`cloudwatch:GetMetricWidgetImage` Necessario per recuperare un grafico istantane o di una o più CloudWatch metriche come immagine bitmap.
ListDashboards	`cloudwatch:ListDashboards` Necessario per visualizzare l'elenco delle CloudWatch dashboard del tuo account.
ListMetrics	`cloudwatch:ListMetrics` Necessario per visualizzare o cercare i nomi delle metriche all'interno della CloudWatch console e nella CLI. Necessario per selezionare i parametri nei widget del pannello di controllo.
ListMetricStreams	`cloudwatch:ListMetricStreams` Necessario per visualizzare o cercare nell'elen co di flussi di parametri nell'account.

CloudWatch Operazioni API	Autorizzazioni necessarie (operazioni API)
PutCompositeAlarm	`cloudwatch:PutCompositeAlarm` Necessario per creare un allarme composito. Per creare un allarme composito, l'autorizzazione `cloudwatch:PutCompositeAlarm` deve avere un ambito `*`. Non è possibile restituire informazioni sugli allarmi compositi se l'autorizzazione `cloudwatch:PutCompositeAlarm` ha un ambito più limitato.
PutDashboard	`cloudwatch:PutDashboard` Necessario per creare un pannello di controllo o aggiornarne uno esistente.
PutMetricAlarm	`cloudwatch:PutMetricAlarm` Necessario per creare o aggiornare un allarme.
PutMetricData	`cloudwatch:PutMetricData` Necessario per creare i parametri.
PutMetricStream	`cloudwatch:PutMetricStream` Necessario per creare un flusso di parametri.
SetAlarmState	`cloudwatch:SetAlarmState` Necessario per impostare manualmente lo stato di un allarme.

CloudWatch Operazioni API	Autorizzazioni necessarie (operazioni API)
StartMetricStreams	`cloudwatch:StartMetricStreams` Necessario per avviare lo scorrimento dei parametri in un flusso di parametri.
StopMetricStreams	`cloudwatch:StopMetricStreams` Necessario per arrestare temporaneamente lo scorrimento dei parametri in un flusso di parametri.
TagResource	`cloudwatch:TagResource` Necessario per aggiungere o aggiornare tag su CloudWatch risorse come allarmi e regole di Contributor Insights.
UntagResource	`cloudwatch:UntagResource` Necessario per rimuovere i tag dalle risorse CloudWatch .

CloudWatch Operazioni dell'API Contributor Insights e autorizzazioni richieste per le azioni

> **Important**
>
> Quando concedi l'`cloudwatch:PutInsightRule`autorizzazione a un utente, per impostazione predefinita quell'utente può creare una regola che valuta qualsiasi gruppo di log in Logs. CloudWatch È possibile aggiungere condizioni di policy IAM che limitano queste autorizzazioni affinché un utente includa ed escluda gruppi di log specifici. Per ulteriori informazioni, consulta la pagina Utilizzo delle chiavi di condizione per limitare l'accesso degli utenti di Contributor Insights ai gruppi di log.

CloudWatch Operazioni dell'API Contributor Insights	Autorizzazioni necessarie (operazioni API)
DeleteInsightRules	`cloudwatch:DeleteInsightRules` Necessaria per eliminare le regole di Contributor Insights.
DescribeInsightRules	`cloudwatch:DescribeInsightRules` Necessaria per visualizzare le regole di Contributor Insights nel tuo account.
EnableInsightRules	`cloudwatch:EnableInsightRules` Necessaria per abilitare le regole di Contributor Insights.
GetInsightRuleReport	`cloudwatch:GetInsightRuleReport` Necessaria per recuperare i dati di serie temporali e altre statistiche raccolte dalle regole di Contributor Insights.
PutInsightRule	`cloudwatch:PutInsightRule` Necessaria per creare le regole di Contributor Insights. Consulta la nota Importante all'inizio di questa tabella.

CloudWatch Operazioni dell'API Events e autorizzazioni richieste per le azioni

CloudWatch Operazioni dell'API Events	Autorizzazioni necessarie (operazioni API)

CloudWatch Operazioni dell'API Events	Autorizzazioni necessarie (operazioni API)
DeleteRule	`events:DeleteRule` Necessario per eliminare una regola.
DescribeRule	`events:DescribeRule` Necessario per elencare i dettagli di una regola.
DisableRule	`events:DisableRule` Necessario per disabilitare una regola.
EnableRule	`events:EnableRule` Necessario per abilitare una regola.
ListRuleNamesByTarget	`events:ListRuleNamesByTarget` Necessario per elencare le regole associate a un target.
ListRules	`events:ListRules` Necessario per elencare tutte le regole nel tuo account.
ListTargetsByRule	`events:ListTargetsByRule` Necessario per elencare tutti i target associati a una regola.

CloudWatch Operazioni dell'API Events	Autorizzazioni necessarie (operazioni API)
PutEvents	`events:PutEvents` Necessario per aggiungere eventi personalizzati per i quali può essere trovata una corrispondenza alle regole.
PutRule	`events:PutRule` Necessario per creare o aggiornare una regola.
PutTargets	`events:PutTargets` Necessario per aggiungere target a una regola.
RemoveTargets	`events:RemoveTargets` Necessario per rimuovere un target da una regola.
TestEventPattern	`events:TestEventPattern` Necessario per testare un modello di evento in un dato evento.

CloudWatch Registra le operazioni API e le autorizzazioni richieste per le azioni

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
CancelExportTask	`logs:CancelExportTask` Necessaria per eliminare un'attività di esportazione in esecuzione o pendente.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
CreateExportTask	`logs:CreateExportTask` Necessaria per esportare dati da un gruppo di log a un bucket Amazon S3.
CreateLogGroup	`logs:CreateLogGroup` Necessaria per creare un nuovo gruppo di log.
CreateLogStream	`logs:CreateLogStream` Necessaria per creare un nuovo flusso di log in un gruppo di log.
DeleteDestination	`logs:DeleteDestination` Necessaria per eliminare una destinazione di log e disabilita tutti i filtri di sottoscrizione.
DeleteLogGroup	`logs:DeleteLogGroup` Necessario per eliminare un gruppo di log e tutti i log eventi archiviati associati.
DeleteLogStream	`logs:DeleteLogStream` Necessaria per eliminare un flusso di log e tutti gli eventi di log archiviati associati.
DeleteMetricFilter	`logs:DeleteMetricFilter` Necessaria per eliminare un filtro parametri associato a un gruppo di log.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Necessario per eliminare una definizione di query salvata in CloudWatch Logs Insights.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Necessario per eliminare una politica delle risorse CloudWatch di Logs.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Necessaria per eliminare una policy di retention di un gruppo di log.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Necessaria per eliminare un filtro sottoscrizioni associato a un gruppo di log.
DescribeDestinations	`logs:DescribeDestinations` Necessaria per visualizzare tutte le destinazioni associate all'account.
DescribeExportTasks	`logs:DescribeExportTasks` Necessaria per visualizzare tutte le attività di esportazione associate all'account.
DescribeLogGroups	`logs:DescribeLogGroups` Necessaria per visualizzare tutti i gruppi di log associati all'account.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
DescribeLogStreams	`logs:DescribeLogStreams` Necessaria per visualizzare tutti i flussi di log associati a un gruppo di log.
DescribeMetricFilters	`logs:DescribeMetricFilters` Necessario per visualizzare tutti i parametri associati a un gruppo di log.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Necessario per visualizzare l'elenco delle definizioni delle query salvate in CloudWatch Logs Insights.
DescribeQueries	`logs:DescribeQueries` Necessario per visualizzare l'elenco delle query di CloudWatch Logs Insights pianificate, in esecuzione o eseguite di recente.
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Necessario per visualizzare un elenco delle politiche relative alle risorse di Logs. CloudWatch
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Necessaria per visualizzare tutti i filtri di sottoscrizione associati a un gruppo di log.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
FilterLogEvents	`logs:FilterLogEvents` Necessaria per ordinare log eventi in base a un modello filtro di gruppo di log.
GetLogEvents	`logs:GetLogEvents` Necessaria per recuperare eventi di log da un flusso di log.
GetLogGroupFields	`logs:GetLogGroupFields` Necessaria per recuperare l'elenco dei campi inclusi negli eventi di log in un gruppo di log.
GetLogRecord	`logs:GetLogRecord` Necessaria per recuperare i dettagli da un singolo log eventi.
GetQueryResults	`logs:GetQueryResults` Necessario per recuperare i risultati delle interrogazioni di CloudWatch Logs Insights.
ListTagsLogGroup	`logs:ListTagsLogGroup` Necessaria per elencare i tag associati a un gruppo di log.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
PutDestination	`logs:PutDestination` Necessaria per creare o aggiornare un flusso di log di destinazione (ad esempio, un flusso Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Necessaria per creare o aggiornare una policy di accesso predefinita associata a una destinazi one di log esistente.
PutLogEvents	`logs:PutLogEvents` Necessaria per caricare un batch di eventi di log in un flusso di eventi.
PutMetricFilter	`logs:PutMetricFilter` Necessaria per creare o aggiornare un filtro di parametri e associarlo a un gruppo di log.
PutQueryDefinition	`logs:PutQueryDefinition` Necessario per salvare una query in CloudWatc h Logs Insights.
PutResourcePolicy	`logs:PutResourcePolicy` Necessario per creare una politica delle risorse CloudWatch di Logs.

CloudWatch Registra le operazioni API	Autorizzazioni necessarie (operazioni API)
PutRetentionPolicy	`logs:PutRetentionPolicy` Necessaria per impostare il numero di giorni per conservare log eventi (retention) in un gruppo di log.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Necessaria per creare o aggiornare un filtro sottoscrizioni e associarlo a un gruppo di log.
StartQuery	`logs:StartQuery` Necessario per avviare le query di CloudWatch Logs Insights.
StopQuery	`logs:StopQuery` Necessario per interrompere una query di CloudWatch Logs Insights in corso.
TagLogGroup	`logs:TagLogGroup` Necessaria per aggiungere o aggiornare i tag dei gruppi di log.
TestMetricFilter	`logs:TestMetricFilter` Necessaria per verificare un modello di filtro su un campionamento di messaggi di log eventi.

Operazioni API Amazon EC2 e autorizzazioni necessarie per le operazioni

Operazioni API Amazon EC2	Autorizzazioni necessarie (operazioni API)
DescribeInstanceStatus	`ec2:DescribeInstanceStatus` Necessario per visualizzare i dettagli di stato dell'istanza EC2.
DescribeInstances	`ec2:DescribeInstances` Necessario per visualizzare i dettagli dell'istanza EC2.
RebootInstances	`ec2:RebootInstances` Necessario per riavviare un'istanza EC2.
StopInstances	`ec2:StopInstances` Necessario per arrestare un'istanza EC2.
TerminateInstances	`ec2:TerminateInstances` Necessario per terminare un'istanza EC2.

Operazioni API Amazon EC2 Auto Scaling e autorizzazioni necessarie per le operazioni

Operazioni API Amazon EC2 Auto Scaling	Autorizzazioni necessarie (operazioni API)
Dimensionamento	`autoscaling:Scaling`

Operazioni API Amazon EC2 Auto Scaling	Autorizzazioni necessarie (operazioni API)
	Necessario per dimensionare un gruppo Auto Scaling.
Trigger	`autoscaling:Trigger` Necessario per attivare un'operazione Auto Scaling.

Convalida della conformità per Amazon CloudWatch

I revisori di terze parti valutano la sicurezza e la conformità di Amazon nell' CloudWatch ambito di diversi programmi di AWS conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco dei AWS servizi che rientrano nell'ambito di specifici programmi di conformità, consulta AWS Services in Scope by Compliance Program AWS . Per informazioni generali, consulta Programmi di conformitàAWS.

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La tua responsabilità di conformità quando usi Amazon CloudWatch è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Security and Compliance Quick Start Guides (Guide Quick Start Sicurezza e compliance): queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper descrive come le aziende possono utilizzare per creare applicazioni conformi allo standard HIPAA. AWS
- AWS Risorse per la conformità Risorse per la conformità: questa raccolta di potrebbe riguardare il settore e la località in cui operate.
- Evaluating Resources with Rules nella AWS Config Developer Guide: AWS Config valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.

- AWS Security Hub— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente di verificare la conformità agli standard e alle best practice del settore della sicurezza. AWS

Resilienza in Amazon CloudWatch

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, connesse tramite reti altamente ridondanti, a bassa latenza e throughput elevato. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Infrastruttura AWS globale.

Sicurezza dell'infrastruttura in Amazon CloudWatch

In quanto servizio gestito, Amazon CloudWatch è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere CloudWatch attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Isolamento della rete

Un cloud privato virtuale (Virtual Private Cloud (VPC)) è una rete virtuale nell'area logicamente isolata in Amazon Web Services Cloud. Una sottorete è un intervallo di indirizzi IP in un VPC. È possibile

distribuire varie risorse AWS nelle sottoreti dei VPC. Ad esempio, nelle sottoreti è possibile distribuire istanze Amazon EC2, cluster EMR e tabelle Dynamo DB. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon VPC.

CloudWatch Per consentire la comunicazione con le risorse in un VPC senza passare attraverso la rete Internet pubblica, usa. AWS PrivateLink Per ulteriori informazioni, consulta Utilizzo CloudWatch e CloudWatch Synthetics con endpoint VPC di interfaccia.

Una sottorete privata è una sottorete senza instradamento predefinito a Internet pubblica. La distribuzione di una AWS risorsa in una sottorete privata non impedisce ad Amazon CloudWatch di raccogliere metriche integrate dalla risorsa.

Se devi pubblicare metriche personalizzate da una AWS risorsa in una sottorete privata, puoi farlo utilizzando un server proxy. Il server proxy inoltra tali richieste HTTPS agli endpoint API pubblici per. CloudWatch

AWS Security Hub

Monitora il tuo utilizzo CloudWatch in relazione alle migliori pratiche di sicurezza utilizzando AWS Security Hub. Security Hub utilizza controlli di sicurezza per valutare le configurazioni delle risorse e gli standard di sicurezza per aiutarti a rispettare vari framework di conformità. Per ulteriori informazioni sull'utilizzo di Security Hub per valutare CloudWatch le risorse, consulta CloudWatch i controlli Amazon nella AWS Security Hub User Guide.

Utilizzo CloudWatch e CloudWatch Synthetics con endpoint VPC di interfaccia

Se utilizzi Amazon Virtual Private Cloud (Amazon VPC) per ospitare AWS le tue risorse, puoi stabilire una connessione privata tra il tuo VPC e Synthetics. CloudWatch CloudWatch Puoi utilizzare queste connessioni per consentire CloudWatch a CloudWatch Synthetics di comunicare con le tue risorse sul tuo VPC senza passare attraverso la rete Internet pubblica.

Amazon VPC è un AWS servizio che puoi utilizzare per avviare AWS risorse in una rete virtuale definita dall'utente. Con un VPC;, detieni il controllo delle impostazioni della rete, come l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete. Per connettere il tuo VPC a o CloudWatch Synthetics CloudWatch , definisci un endpoint VPC di interfaccia per connettere il tuo VPC ai servizi. AWS L'endpoint fornisce una connettività affidabile e scalabile a o CloudWatch

CloudWatch Synthetics senza richiedere un gateway Internet, un'istanza NAT (Network Address Translation) o una connessione VPN. Per ulteriori informazioni, consulta Che cos'è Amazon VPC? nella Guida per l'utente Amazon VPC.

Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, una AWS tecnologia che consente la comunicazione privata tra AWS i servizi utilizzando un'interfaccia di rete elastica con indirizzi IP privati. Per ulteriori informazioni, consulta il post del blog New — AWS PrivateLink for AWS Services.

Le fasi seguenti sono per gli utenti Amazon VPC. Per ulteriori informazioni, consulta l'argomento relativo alle nozioni di base nella Guida per l'utente di Amazon VPC.

CloudWatch Endpoint VPC

CloudWatch attualmente supporta gli endpoint VPC nelle seguenti regioni: AWS

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- US West (Oregon)
- Asia Pacifico (Hong Kong)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europa (Parigi)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti orientali)

- AWS GovCloud (Stati Uniti occidentali)

Creazione di un endpoint VPC per CloudWatch

Per iniziare a utilizzarlo CloudWatch con il tuo VPC, crea un endpoint VPC di interfaccia per. CloudWatch Il nome del servizio da scegliere è `com.amazonaws.region.monitoring`. Per ulteriori informazioni, consulta Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

Non è necessario modificare le impostazioni per. CloudWatch CloudWatch chiama altri AWS servizi utilizzando endpoint pubblici o endpoint VPC con interfaccia privata, a seconda di quali siano in uso. Ad esempio, se crei un endpoint VPC di interfaccia per e hai già metriche CloudWatch provenienti da risorse situate sul tuo VPC CloudWatch, queste metriche iniziano a fluire attraverso l'endpoint VPC dell'interfaccia per impostazione predefinita.

Controllo dell'accesso all'endpoint CloudWatch VPC

Una policy endpoint VPC è una policy della risorsa IAM che viene collegata a un endpoint durante la creazione o la modifica dell'endpoint. Se non colleghi una policy durante la creazione di un endpoint, Amazon VPC collega una policy predefinita che consente l'accesso completo al servizio. Una policy endpoint non esclude né sostituisce policy dell'utente o policy specifiche del servizio. Si tratta di una policy separata per controllare l'accesso dall'endpoint al servizio specificato.

Le policy endpoint devono essere scritte in formato JSON.

Per ulteriori informazioni, consultare Controllo degli accessi ai servizi con endpoint VPC nella Guida per l'utente di Amazon VPC.

Di seguito è riportato un esempio di policy sugli endpoint per. CloudWatch Questa policy consente agli utenti che si connettono CloudWatch tramite il VPC di inviare dati metrici CloudWatch e impedisce loro di eseguire altre azioni. CloudWatch

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
```

```
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Per modificare la policy degli endpoint VPC per CloudWatch

1. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione, seleziona Endpoints (Endpoint).
3. Se non hai ancora creato l'endpoint per CloudWatch, scegli Crea endpoint. Seleziona quindi com.amazonaws.***region***.monitoring e scegli Create endpoint (Crea endpoint).
4. Seleziona l'endpoint com.amazonaws.***region***.monitoring quindi scegli la scheda Policy (Policy).
5. Scegli Edit Policy (Modifica policy), quindi apporta le modifiche.

CloudWatch Endpoint VPC Synthetics

CloudWatch Synthetics attualmente supporta gli endpoint VPC nelle seguenti regioni: AWS

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- US West (Oregon)
- Asia Pacifico (Hong Kong)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europa (Parigi)

- Sud America (San Paolo)

Creazione di un endpoint VPC per Synthetics CloudWatch

Per iniziare a utilizzare CloudWatch Synthetics con il tuo VPC, crea un endpoint VPC di interfaccia per Synthetics. CloudWatch Il nome del servizio da scegliere è `com.amazonaws.region.synthetics`. Per ulteriori informazioni, consulta Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

Non è necessario modificare le impostazioni di CloudWatch Synthetics. CloudWatch Synthetics comunica con AWS altri servizi utilizzando endpoint pubblici o endpoint VPC con interfaccia privata, a seconda di quale siano in uso. Ad esempio, se crei un endpoint VPC di interfaccia per Synthetics e disponi già di un endpoint di interfaccia per Amazon S3, CloudWatch Synthetics CloudWatch inizia a comunicare con Amazon S3 tramite l'endpoint VPC di interfaccia per impostazione predefinita.

Controllo dell'accesso all'endpoint VPC CloudWatch Synthetics

Una policy endpoint VPC è una policy della risorsa IAM che viene collegata a un endpoint durante la creazione o la modifica dell'endpoint. Se non colleghi una policy durante la creazione di un endpoint, viene collegata una policy predefinita che consente l'accesso completo al servizio. Una policy endpoint non esclude né sostituisce policy dell'utente o policy specifiche del servizio. Si tratta di una policy separata per controllare l'accesso dall'endpoint al servizio specificato.

Le policy degli endpoint influiscono sulle canary gestite privatamente da VPC. Non sono necessarie per canary che funzionano su sottoreti private.

Le policy endpoint devono essere scritte in formato JSON.

Per ulteriori informazioni, consultare Controllo degli accessi ai servizi con endpoint VPC nella Guida per l'utente di Amazon VPC.

Di seguito è riportato un esempio di policy sugli endpoint per CloudWatch Synthetics. Questa politica consente agli utenti che si connettono a CloudWatch Synthetics tramite VPC di visualizzare informazioni sui canarini e sulle loro corse, ma non di creare, modificare o eliminare i canarini.

```
{
    "Statement": [
        {
            "Action": [
                "synthetics:DescribeCanaries",
```

```
                "synthetics:GetCanaryRuns"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

Per modificare la policy degli endpoint VPC per Synthetics CloudWatch

1. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nel pannello di navigazione, seleziona Endpoints (Endpoint).
3. Se non avete già creato l'endpoint per CloudWatch Synthetics, scegliete Crea endpoint. Seleziona com.amazonaws.***region***.synthetics, quindi scegli Crea endpoint.
4. Seleziona l'endpoint com.amazonaws.***region***.synthetics, quindi scegli la scheda Policy (Policy).
5. Scegli Edit Policy (Modifica policy), quindi apporta le modifiche.

Considerazioni sulla sicurezza per Canary Synthetics

Nelle sezioni seguenti vengono descritti i problemi di sicurezza che occorre considerare durante la creazione e l'esecuzione di Canary in Synthetics.

Utilizzo di connessioni sicure

Poiché il codice Canary e i risultati delle sessioni di test Canary possono contenere informazioni riservate, non connettere il Canary a endpoint su connessioni non crittografate. Utilizza sempre connessioni crittografate, ad esempio quelle che iniziano con `https://`.

Considerazioni sulle convenzioni di denominazione dei Canary

L'Amazon Resource Name (ARN) di un canary è incluso nell'intestazione user-agent come parte delle chiamate in uscita effettuate dal browser Chromium basato su Puppeteer, incluso nella libreria wrapper Synthetics. CloudWatch Questo aiuta a identificare CloudWatch il traffico canarino di Synthetics e a ricollegarlo ai canarini che effettuano chiamate.

L'ARN del Canary include il nome Canary. Scegli nomi Canary che non rivelano informazioni proprietarie.

Inoltre, assicurati di indirizzare i Canary solo ai siti Web e agli endpoint che controlli.

Segreti e informazioni sensibili in codice canary

Se trasmetti il codice canarino direttamente al canarino utilizzando un file zip, il contenuto dello script può essere visualizzato nei log. AWS CloudTrail

Se lo script canary contiene informazioni sensibili o segreti (come chiavi di accesso o credenziali del database), ti consigliamo di archiviare lo script come oggetto con controllo delle versioni in Amazon S3 e trasferire la posizione Amazon S3 al canary, invece di trasferire il codice canary tramite un file con estensione zip.

Se vuoi utilizzare un file con estensione zip per trasferire lo script canary, ti consigliamo di non includere segreti o informazioni sensibili nel codice sorgente del canary. Per ulteriori informazioni su come utilizzare per AWS Secrets Manager proteggere i tuoi segreti, vedi Cos'è? AWS Secrets Manager .

Considerazioni sulle autorizzazioni

Ti consigliamo di limitare l'accesso alle risorse create o utilizzate da CloudWatch Synthetics. Utilizza le autorizzazioni strette sui bucket Amazon S3 in cui i Canary archiviano i risultati della sessione di test e altri artefatti, come log e screenshot.

Allo stesso modo, mantieni autorizzazioni strette sulle posizioni in cui è archiviato il codice sorgente del Canary, in modo che nessun utente cancelli accidentalmente o intenzionalmente i livelli Lambda o le funzioni Lambda utilizzate per il Canary.

Per essere sicuro di eseguire il codice Canary desiderato, puoi utilizzare funzione Versioni multiple degli oggetti sul bucket Amazon S3 in cui è archiviato il codice del Canary. Quindi, quando specifichi questo codice da eseguire come un Canary, puoi includere l'oggetto `versionId` come parte del percorso, come negli esempi seguenti.

```
https://bucket.s3.amazonaws.com/path/object.zip?versionId=version-id
https://s3.amazonaws.com/bucket/path/object.zip?versionId=version-id
https://bucket.s3-region.amazonaws.com/path/object.zip?versionId=version-id
```

Tracce di stack e messaggi di eccezione

Per impostazione predefinita, CloudWatch Synthetics canaries cattura qualsiasi eccezione generata dal tuo script canary, indipendentemente dal fatto che lo script sia personalizzato o provenga da un

blueprint. CloudWatch Synthetics registra sia il messaggio di eccezione che la traccia dello stack in tre posizioni:

- Torna al servizio CloudWatch Synthetics per velocizzare il debug quando descrivi le esecuzioni dei test
- Into CloudWatch Logs in base alla configurazione con cui vengono create le funzioni Lambda
- Nel file di log Synthetics, che è un file di testo normale che viene caricato nella posizione Amazon S3 specificata dal valore impostato per il `resultsLocation` del Canary

Se desideri inviare e archiviare meno informazioni, puoi acquisire le eccezioni prima che tornino alla libreria wrapper CloudWatch Synthetics.

Puoi anche inserire gli URL di richiesta nei tuoi errori. CloudWatch Synthetics analizza tutti gli URL presenti nell'errore generato dallo script e ne rimuove i parametri URL limitati in base alla configurazione. restrictedUrlParameters Se registri messaggi di errore nello script, puoi utilizzare getSanitizedErrorMessaggio per redigere gli URL prima della registrazione.

Restrizione dell'ambito dei ruoli IAM

Ti consigliamo di non configurare il Canary per visitare URL o endpoint potenzialmente dannosi. Indirizzare il Canary a siti Web o endpoint non attendibili o sconosciuti potrebbe esporre il codice della funzione Lambda a script di utenti malintenzionati. Se un sito Web dannoso interrompe Chromium, potrebbe accedere al codice Lambda in modo simile a quanto accade quando ci si connette utilizzando un browser Internet.

Esegui la funzione Lambda con un ruolo di esecuzione IAM che dispone di un numero minore di autorizzazioni. In questo modo, se la tua funzione Lambda è compromessa da uno script dannoso, è limitata nelle azioni che può intraprendere quando è in esecuzione come account del tuo canarino. AWS

Quando usi la CloudWatch console per creare un canarino, questa viene creata con un ruolo di esecuzione IAM ristretto.

Redazione dei dati sensibili

CloudWatch Synthetics acquisisce URL, codice di stato, motivo dell'errore (se presente) e intestazioni e corpi di richieste e risposte. Ciò consente a un utente canary di comprendere, monitorare ed eseguire il debug dei canary.

Le configurazioni descritte nelle sezioni seguenti possono essere impostate in qualsiasi momento dell'esecuzione del canary. Puoi anche scegliere di applicare configurazioni diverse a diversi passaggi di Synthetics.

URL della richiesta

Per impostazione predefinita, CloudWatch Synthetics registra gli URL delle richieste, i codici di stato e il motivo dello stato per ogni URL nei registri di Canary. Gli URL della richiesta possono essere visualizzati anche nei report di esecuzione dei canary, nei file HAR e così via. L'URL della richiesta potrebbe contenere parametri di query sensibili, ad esempio token di accesso o password. È possibile impedire che le informazioni sensibili vengano registrate da CloudWatch Synthetics.

Per oscurare le informazioni sensibili, imposta la proprietà di configurazione. restrictedUrlParameters Per ulteriori informazioni, consulta SyntheticsConfiguration classe. Ciò fa sì che CloudWatch Synthetics rediga i parametri URL, inclusi i valori dei parametri di percorso e query, in base a prima della registrazione. restrictedUrlParameters Se registri gli URL nello script, puoi utilizzare getSanitizedUrl(url, stepConfig = null) per redigere gli URL prima della registrazione. Per ulteriori informazioni, consulta SyntheticsLogHelper classe.

Headers

Per impostazione predefinita, CloudWatch Synthetics non registra le intestazioni di richiesta/risposta. Per i canary dell'interfaccia utente, questo è il comportamento predefinito per i canary che utilizzano la versione di runtime `syn-nodejs-puppeteer-3.2` e versioni successive.

Se le intestazioni non contengono informazioni sensibili, puoi abilitare le intestazioni nei file HAR e nei report HTTP impostando le proprietà and su. includeRequestHeadersincludeResponseHeaders`true` Puoi abilitare tutte le intestazioni, ma scegliere di limitare i valori delle chiavi di intestazione sensibili. Puoi ad esempio scegliere di redigere solo intestazioni `Authorization` da artefatti prodotti dai canary.

Corpo della richiesta e della risposta

Per impostazione predefinita, CloudWatch Synthetics non registra il corpo della richiesta/risposta nei registri o nei report di Canary. Queste informazioni sono particolarmente utili per i canary delle API. Synthetics acquisisce tutte le richieste HTTP e può mostrare intestazioni e corpi delle richieste e della risposta. Per ulteriori informazioni, consulta executeHttpStep(StepName, RequestOptions, [callback], [StepConfig]). È possibile scegliere di abilitare il corpo della richiesta/risposta impostando le proprietà and su. includeRequestBodyincludeResponseBody`true`

Registrazione delle chiamate CloudWatch API Amazon con AWS CloudTrail

Amazon CloudWatch e CloudWatch Synthetics sono integrati AWS CloudTrail con un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o servizio. AWS CloudTrail acquisisce le chiamate API effettuate da o per conto del tuo account. AWS Le chiamate acquisite includono chiamate dalla console e chiamate di codice alle operazioni API.

Se crei un trail, puoi abilitare la consegna continua di CloudTrail eventi a un bucket S3, inclusi gli eventi per. CloudWatch Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviata CloudWatch, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e altri dettagli.

Per ulteriori informazioni CloudTrail, incluso come configurarlo e abilitarlo, consulta la Guida per l'AWS CloudTrail utente.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, vedete l'elemento CloudTrail userIdentity.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per CloudWatch e CloudWatch Synthetics, crea un percorso. Un trail consente di CloudTrail inviare file di registro a un bucket S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il percorso registra gli eventi da tutte le regioni della AWS partizione e consegna i file di registro al bucket S3 specificato. Puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un trail

- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

> **Note**
>
> Per informazioni sulle chiamate API CloudWatch Logs effettuate CloudTrail, consulta CloudWatch Logs information in. CloudTrail

Argomenti

- CloudWatch informazioni in CloudTrail
- CloudWatch Internet Monitor in CloudTrail
- CloudWatch Informazioni Synthetics in CloudTrail

CloudWatch informazioni in CloudTrail

CloudWatch supporta la registrazione delle seguenti azioni come eventi nei file di CloudTrail registro:

- DeleteAlarms
- DeleteAnomalyDetector
- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- DisableAlarmActions
- EnableAlarmActions
- GetDashboard
- ListDashboards
- PutAnomalyDetector

- PutDashboard
- PutMetricAlarm
- SetAlarmState

Esempio: voci dei file di CloudWatch registro

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`PutMetricAlarm`azione.


```
{
    "Records": [{
        "eventVersion": "1.01",
        "userIdentity": {
            "type": "Root",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID"
        },
        "eventTime": "2014-03-23T21:50:34Z",
        "eventSource": "monitoring.amazonaws.com",
        "eventName": "PutMetricAlarm",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "threshold": 50.0,
            "period": 60,
            "metricName": "CloudTrail Test",
            "evaluationPeriods": 3,
            "comparisonOperator": "GreaterThanThreshold",
            "namespace": "AWS/CloudWatch",
            "alarmName": "CloudTrail Test Alarm",
            "statistic": "Sum"
        },
        "responseElements": null,
        "requestID": "29184022-b2d5-11e3-a63d-9b463e6d0ff0",
        "eventID": "b096d5b7-dcf2-4399-998b-5a53eca76a27"
    },
    ..additional entries
  ]
  }
```

La seguente voce del file di registro mostra che un utente ha chiamato l'`PutRule`azione CloudWatch Events.


```
{
         "eventVersion":"1.03",
         "userIdentity":{
            "type":"Root",
            "principalId":"123456789012",
            "arn":"arn:aws:iam::123456789012:root",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext":{
               "attributes":{
                  "mfaAuthenticated":"false",
                  "creationDate":"2015-11-17T23:56:15Z"
               }
            }
         },
         "eventTime":"2015-11-18T00:11:28Z",
         "eventSource":"events.amazonaws.com",
         "eventName":"PutRule",
         "awsRegion":"us-east-1",
         "sourceIPAddress":"AWS Internal",
         "userAgent":"AWS CloudWatch Console",
         "requestParameters":{
            "description":"",
            "name":"cttest2",
            "state":"ENABLED",
            "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}",
            "scheduleExpression":""
         },
         "responseElements":{
            "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
         },
         "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
         "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
         "eventType":"AwsApiCall",
         "apiVersion":"2015-10-07",
         "recipientAccountId":"123456789012"
}
```

La seguente voce del file di registro mostra che un utente ha chiamato l'`CreateExportTask`azione CloudWatch Logs.


```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```


CloudWatch Internet Monitor in CloudTrail

CloudWatch Internet Monitor supporta la registrazione delle seguenti azioni come eventi nei file di CloudTrail registro.

- CreateMonitor

- DeleteMonitor
- GetHealthEvent
- GetMonitor
- GetQueryResults
- GetQueryStatus
- ListHealthEvents
- ListMonitors
- ListTagsForResource
- StartQuery
- StopQuery
- UpdateMonitor

Esempio: voci dei file di registro di CloudWatch Internet Monitor

L'esempio seguente mostra una voce di registro di CloudTrail Internet Monitor che illustra l'`ListMonitors`azione.


```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
                    "mfaAuthenticated": "false"
                }
```

```
            }
        },
        "eventTime": "2022-10-11T17:30:18Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "ListMonitors",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": null,
        "responseElements": null,
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```

L'esempio seguente mostra una voce di registro di CloudTrail Internet Monitor che illustra l'`CreateMonitor`azione.

```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
                    "mfaAuthenticated": "false"
                }
```

```
            }
        },
        "eventTime": "2022-10-11T17:30:08Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "CreateMonitor",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": {
            "MonitorName": "TestMonitor",
            "Resources": ["arn:aws:ec2:us-east-2:444455556666:vpc/vpc-febc0b95"],
            "ClientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
        },
        "responseElements": {
            "Arn": "arn:aws:internetmonitor:us-east-2:444455556666:monitor/ct-
onboarding-test",
            "Status": "PENDING"
        },
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```

CloudWatch Informazioni Synthetics in CloudTrail

CloudWatch Synthetics supporta la registrazione delle seguenti azioni come eventi nei file di registro: CloudTrail

- CreateCanary
- DeleteCanary
- DescribeCanaries
- DescribeCanariesLastRun
- DescribeRuntimeVersions
- GetCanary
- GetCanaryRuns

- ListTagsForResource
- StartCanary
- StopCanary
- TagResource
- UntagResource
- UpdateCanary

Esempio: voci del CloudWatch file di registro Synthetics

L'esempio seguente mostra una voce di registro CloudTrail Synthetics che illustra l'azione. `DescribeCanaries`

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "DescribeCanaries",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
```

```
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "201ed5f3-15db-4f87-94a4-123456789",
    "eventID": "73ddbd81-3dd0-4ada-b246-123456789",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

L'esempio seguente mostra una voce di registro CloudTrail Synthetics che illustra l'azione. `UpdateCanary`

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
               "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
       "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "UpdateCanary",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
```

```
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Schedule": {
            "Expression": "rate(1 minute)"
        },
        "name": "sample_canary_name",
        "Code": {
            "Handler": "myOwnScript.handler",
            "ZipFile": "SAMPLE_ZIP_FILE"
        }
    },
    "responseElements": null,
    "requestID": "fe4759b0-0849-4e0e-be71-1234567890",
    "eventID": "9dc60c83-c3c8-4fa5-bd02-1234567890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

L'esempio seguente mostra una voce di registro CloudTrail Synthetics che illustra l'azione. `GetCanaryRuns`

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:assumed-role/role_name",
        "accountId":"123456789012",
        "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::111222333444:role/Administrator",
       "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
```

```
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "GetCanaryRuns",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Filter": "TIME_RANGE",
        "name": "sample_canary_name",
        "FilterValues": [
            "2020-04-08T23:00:00.000Z",
            "2020-04-08T23:10:00.000Z"
        ]
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Taggare le tue risorse Amazon CloudWatch

Un tag è un'etichetta di attributo personalizzata che tu o AWS assegnate a una risorsa. AWS Ogni tag è costituito da due parti:

- Una chiave del tag (ad esempio, `CostCenter`, `Environment` o `Project`). Le chiavi dei tag prevedono una distinzione tra lettere maiuscole e minuscole.
- Un campo facoltativo noto come valore del tag (ad esempio, `111122223333` o `Production`). Non specificare il valore del tag equivale a utilizzare una stringa vuota. Analogamente alle chiavi dei tag, i valori dei tag prevedono una distinzione tra lettere maiuscole e minuscole.

I tag consentono di:

- Identifica e organizza le tue AWS risorse. Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, puoi assegnare lo stesso tag a una CloudWatch regola che assegni a un'istanza EC2.

Le seguenti sezioni forniscono ulteriori informazioni sui tag per. CloudWatch

Risorse supportate in CloudWatch

Le seguenti risorse per il tagging di CloudWatch supporto:

- Allarmi: puoi etichettare gli allarmi utilizzando il comando tag-resource AWS CLI e l'API. TagResource Puoi anche visualizzare e gestire i tag degli allarmi utilizzando la pagina dei dettagli degli allarmi nella console. CloudWatch
- Canarie: puoi taggare i canarini utilizzando la console. CloudWatch Per ulteriori informazioni, consulta Creazione di un Canary.
- Regole di Contributor Insights: puoi taggare le regole di Contributor Insights quando le crei utilizzando il put-insight-rule AWS CLI comando e l'API. PutInsightRule Puoi aggiungere tag alle regole esistenti utilizzando il AWS CLI comando tag-resource e l'API. TagResource
- Flussi metrici: puoi taggare i flussi metrici quando li crei utilizzando il comando e l'API. put-metric-stream AWS CLIPutMetricStream Puoi aggiungere tag ai flussi di metrici esistenti utilizzando il comando tag-resource e l'API. AWS CLITagResource

Per informazioni sull'aggiunta e la gestione dei tag, consulta Gestione dei tag.

Gestione dei tag

I tag si compongono delle proprietà `Key` e `Value` in una risorsa. Puoi utilizzare la CloudWatch console AWS CLI, l'o l' CloudWatch API per aggiungere, modificare o eliminare i valori di queste proprietà. Per informazioni sull'utilizzo dei tag, consulta quanto segue.

- TagResourcee ListTagsForResourcenell'Amazon CloudWatch API Reference UntagResource
- tag-resource, untag-resource e in Amazon CLI Reference list-tags-for-resource CloudWatch
- Utilizzo dell'editor di tag nella Guida per l'utente di Resource Groups

Convenzioni di denominazione e utilizzo dei tag

Le seguenti convenzioni di base di denominazione e utilizzo si applicano all'utilizzo dei tag con le risorse: CloudWatch

- Ogni risorsa può avere un massimo di 50 tag.
- Per ciascuna risorsa, ogni chiave del tag deve essere univoca e ogni chiave del tag può avere un solo valore.
- La lunghezza massima delle chiavi di tag è 128 caratteri Unicode in UTF-8.
- Il valore massimo dei tag è 256 caratteri Unicode in UTF-8.
- I caratteri consentiti sono lettere, numeri, spazi rappresentabili in formato UTF-8, oltre ai seguenti caratteri: . : + = @ _ / - (trattino).
- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole. Come best practice, è consigliabile definire una strategia per l'uso delle lettere maiuscole e minuscole nei tag e implementarla costantemente in tutti i tipi di risorse. Ad esempio, puoi decidere se utilizzare `Costcenter`, `costcenter` o `CostCenter` e utilizzare la stessa convenzione per tutti i tag. Non utilizzare tag simili con lettere maiuscole o minuscole incoerenti.
- Il `aws:` prefisso è vietato per i tag perché è riservato all'uso. AWS Non è possibile modificare né eliminare le chiavi o i valori di tag con tale prefisso. I tag con questo prefisso non vengono conteggiati per il limite del numero di tag per risorsa.

Integrazione Grafana

Puoi utilizzare Grafana versione 6.5.0 e successive per avanzare contestualmente nella CloudWatch console e interrogare un elenco dinamico di metriche utilizzando i caratteri jolly. In questo modo puoi monitorare i parametri relativi alle risorse AWS , ad esempio istanze o container di Amazon Elastic Compute Cloud. Quando vengono create nuove istanze come parte di un evento di Auto Scaling, verranno visualizzate automaticamente nel grafico senza la necessità di monitorare gli ID della nuova istanza. Le dashboard predefinite aiutano a semplificare l'esperienza iniziale per il monitoraggio di Amazon EC2, Amazon Elastic Block Store AWS Lambda e delle risorse.

È possibile utilizzare Grafana versione 7.0 e successive per eseguire query di CloudWatch Logs Insights sui gruppi di log in Logs. CloudWatch Puoi visualizzare i risultati della query in grafici a barre, a linee e in pila e in formato tabella. Per ulteriori informazioni su CloudWatch Logs Insights, consulta Analisi dei dati di log con Logs Insights. CloudWatch

Per ulteriori informazioni su come iniziare, consulta Using AWS CloudWatch in Grafana nella documentazione di Grafana Labs.

Console per più account e più regioni CloudWatch

Per ottenere la più completa esperienza di osservabilità e scoperta tra account per le tue metriche, i log e le tracce, ti consigliamo di utilizzare l'osservabilità tra account. CloudWatch Per ulteriori informazioni, consulta CloudWatch osservabilità tra più account.

CloudWatch offre anche una dashboard per più account e più regioni. CloudWatch Questa funzionalità offre visibilità su più account per pannelli di controllo, allarmi, parametri e pannelli di controllo automatici, ma non per log e tracce.

Se utilizzi anche l'osservabilità CloudWatch tra account, un caso d'uso di questa CloudWatch dashboard consiste nel consentire a uno dei tuoi account di origine dell'osservabilità tra account di visualizzazione delle CloudWatch metriche di un altro account di origine.

Il resto di questa sezione descrive il pannello di controllo su più account e regioni. Puoi utilizzarla per creare dashboard che riepilogano CloudWatch i dati di più AWS account e più regioni in un'unica dashboard. AWS È inoltre possibile creare un avviso in un account che osservi un parametro che si trova in un account diverso.

Molte organizzazioni hanno le proprie AWS risorse distribuite in più account, per fornire limiti di fatturazione e sicurezza. In questo caso, si consiglia di designare uno o più account come account di monitoraggio e creare in tali account pannelli di controllo su più account.

La funzionalità per più account è integrata con AWS Organizations, per aiutarti a creare in modo efficiente le tue dashboard per più account.

Funzionalità tra più regioni

La funzionalità tra più regioni è ora integrata automaticamente. Non è necessario eseguire alcuna procedura aggiuntiva per poter visualizzare i parametri di diverse regioni in un singolo account sullo stesso grafico o sullo stesso pannello di controllo. La funzionalità tra regioni non è supportata per gli allarmi, quindi non è possibile creare un allarme in una regione che osserva un parametro in un'altra regione.

Argomenti

Attivazione della funzionalità tra account in CloudWatch

Per configurare la funzionalità tra account nella CloudWatch console, utilizza la CloudWatch console per configurare gli account di condivisione e gli account di monitoraggio.

Configurazione di un account di condivisione

È necessario abilitare la condivisione in ogni account che renderà i dati disponibili per l'account di monitoraggio.

In questo modo verranno concesse autorizzazioni di sola lettura che hai scelto nel passaggio 5 a tutti gli utenti che visualizzano un pannello di controllo tra più account nell'account con cui è attiva la condivisione, se l'utente ha le autorizzazioni corrispondenti nell'account con è attiva la condivisione.

Per consentire al tuo account di condividere CloudWatch dati con altri account

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Impostazioni.
3. Per Condividi i tuoi CloudWatch dati, scegli Configura.
4. In Sharing (Condivisione), scegliere Specific accounts (Account specifici) e immettere gli ID degli account di cui si desidera condividere i dati.

   Tutti gli account che specifichi qui possono visualizzare i CloudWatch dati del tuo account. Specificare gli ID solo degli account conosciuti e attendibili.
5. Per Permissions (Autorizzazioni), specificare come condividere i dati con una delle seguenti opzioni:

   - Fornisci l'accesso in sola lettura a CloudWatch metriche, dashboard e allarmi. Questa opzione consente agli account di monitoraggio di creare dashboard per più account che includono widget contenenti i dati del tuo account. CloudWatch
   - Includi dashboard CloudWatch automatici. Se si seleziona questa opzione, gli utenti dell'account di monitoraggio possono anche visualizzare le informazioni nei pannelli di controllo automatici di questo account. Per ulteriori informazioni, consulta Guida introduttiva ad Amazon CloudWatch.
   - Include l'accesso in sola lettura a X-Ray per la mappa di tracciamento X-Ray. Se si seleziona questa opzione, gli utenti dell'account di monitoraggio possono anche visualizzare la mappa del servizio X-Ray e le informazioni di traccia di X-Ray in questo account. Per ulteriori informazioni, consulta Uso della mappa del tracciamento X-Ray.

- Full read-only access to everything in your account (Accesso completo in sola lettura a tutti di dati dell'account). Questa opzione consente agli account utilizzati per la condivisione di creare dashboard tra più account che includono widget contenenti CloudWatch i dati del tuo account. Consente inoltre a tali account di esaminare più a fondo l'account e visualizzarne i dati nelle console di altri servizi AWS .

6. Scegli Launch template. CloudFormation

   Nella schermata di conferma digitare **`Confirm`** e scegliere Launch template (Avvia modello).

7. Selezionare la casella di controllo I acknowledge... (Acconsento...) e scegliere Create stack (Crea stack).

Condivisione con un'intera organizzazione

Completando la procedura precedente viene creato un ruolo IAM che consente all'account di condividere i dati con un account. È possibile creare o modificare un ruolo IAM che condivide i dati con tutti gli account di un'organizzazione. Eseguire questa operazione solo se si conoscono e si considerano attendibili tutti gli account dell'organizzazione.

In questo modo verranno concesse autorizzazioni di sola lettura elencate nelle policy mostrate nel passaggio 5 a tutti gli utenti che visualizzano un pannello di controllo tra più account nell'account con cui è attiva la condivisione, se l'utente ha le autorizzazioni corrispondenti nell'account con è attiva la condivisione.

Per condividere i dati CloudWatch del tuo account con tutti gli account di un'organizzazione

1. Se non l'hai già fatto, completa la procedura precedente per condividere i dati con un solo AWS account.
2. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
3. Nel riquadro di navigazione, seleziona Ruoli.
4. Nell'elenco dei ruoli, scegli CloudWatch- CrossAccountSharingRole.
5. Scegliere Trust relationships (Relazioni di trust), quindi Edit trust relationship (Modifica relazione di trust).

   Viene visualizzata una policy come questa:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Modificare la policy nel modo seguente, sostituendo *org-id* con l'ID dell'organizzazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "org-id"
        }
      }

    }
  ]
}
```

7. Scegli Update Trust Policy (Aggiorna policy di trust).

Configurazione di un account di monitoraggio

Abilita ogni account di monitoraggio se desideri visualizzare i CloudWatch dati di più account.

Una volta completata la procedura seguente, CloudWatch crea un ruolo collegato al servizio da CloudWatch utilizzare nell'account di monitoraggio per accedere ai dati condivisi dagli altri account. Questo ruolo collegato al servizio viene chiamato. AWSServiceRoleForCloudWatchCrossAccount Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per CloudWatch.

Per consentire al tuo account di visualizzare i dati di più account CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione scegli Settings (Impostazioni), quindi nella sezione Cross-account cross-region (Su più account tra più regioni), scegli Configure (Configura).
3. Nella sezione Visualizza tra più account interregionali, scegli Abilita, quindi seleziona la casella di controllo Mostra selettore nella console per consentire la visualizzazione di un selettore di account nella CloudWatch console quando rappresenti graficamente una metrica o crei un allarme.
4. In View cross-account cross-region (Visualizzazione su più account tra più regioni), scegliere una delle seguenti opzioni:

   - Account Id Input (Input ID account). Questa opzione richiede di immettere manualmente un ID account ogni volta che si desidera passare a un altro account quando si visualizzano i dati su più account.
   - AWS Selettore dell'account dell'organizzazione. Questa opzione fa sì che vengano visualizzati gli account specificati al termine dell'integrazione su più account con Organizations visualizzato. La prossima volta che utilizzi la console, CloudWatch visualizza un elenco a discesa di questi account tra cui scegliere quando visualizzi i dati tra più account.

     A tale scopo, è necessario aver prima utilizzato l'account di gestione dell'organizzazione CloudWatch per consentire la visualizzazione di un elenco degli account dell'organizzazione. Per ulteriori informazioni, consulta (Facoltativo) Effettua l'integrazione con AWS Organizations.
   - Custom account selector (Selettore account personalizzato). Questa opzione richiede di immettere un elenco di ID account. La prossima volta che utilizzi la console, CloudWatch visualizza un elenco a discesa di questi account tra cui scegliere quando visualizzi i dati tra più account.

     È anche possibile immettere un'etichetta per ciascuno di questi account per identificarli quando si scelgono gli account da visualizzare.

     Le impostazioni del selettore account effettuate qui da un utente vengono mantenute solo per tale utente, non per tutti gli altri utenti nell'account di monitoraggio.
5. Scegli Abilita .

Dopo aver completato questa configurazione, è possibile creare pannelli di controllo su più account. Per ulteriori informazioni, consulta Pannelli di controllo su più account tra più regioni.

(Facoltativo) Effettua l'integrazione con AWS Organizations

Se si desidera integrare la funzionalità tra account AWS Organizations, è necessario creare un elenco di tutti gli account dell'organizzazione a disposizione degli account di monitoraggio.

Per abilitare la CloudWatch funzionalità tra account per accedere a un elenco di tutti gli account dell'organizzazione

1. Accedi all'account di gestione della tua organizzazione.
2. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
3. Nel riquadro di spostamento scegliere Settings (Impostazioni), quindi scegliere Configure (Configura).
4. Per Grant permission to view the list of accounts in the organization (Concedi l'autorizzazione a visualizzare l'elenco degli account nell'organizzazione), scegliere Specific accounts (Account specifici) per inserire un elenco di ID account. L'elenco degli account nell'organizzazione viene condiviso solo con gli account specificati qui.
5. Scegliere Share organization account list (Condividi elenco account organizzazione).
6. Scegli Launch CloudFormation template.

   Nella schermata di conferma digitare **`Confirm`** e scegliere Launch template (Avvia modello).

Risoluzione dei problemi relativi alla CloudWatch configurazione di più account

Questa sezione contiene suggerimenti per la risoluzione dei problemi relativi alla distribuzione tra account e console in. CloudWatch

Sto ricevendo errori di accesso negato di visualizzazioni di dati su più account

Verifica quanto segue:

- Il tuo account di monitoraggio dovrebbe avere un ruolo denominato AWSServiceRoleForCloudWatchCrossAccount. In caso contrario, è necessario creare questo ruolo. Per ulteriori informazioni, consulta Set Up a Monitoring Account.
- Ogni account di condivisione deve avere un ruolo denominato CloudWatch-CrossAccountSharingRole. In caso contrario, è necessario creare questo ruolo. Per ulteriori informazioni, consulta la pagina Set Up A Sharing Account.

- Il ruolo di condivisione deve considerare attendibile l'account di monitoraggio.

Per confermare che i ruoli siano configurati correttamente per la console con CloudWatch più account

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Nell'elenco dei ruoli, assicurarsi che esista il ruolo necessario. In un account di condivisione, cerca CloudWatch- CrossAccountSharingRole. In un account di monitoraggio, cerca AWSServiceRoleForCloudWatchCrossAccount.
4. Se hai un account di condivisione e CloudWatch- esiste CrossAccountSharingRole già, scegli CloudWatch- CrossAccountSharingRole.
5. Scegliere Trust relationships (Relazioni di trust), quindi Edit trust relationship (Modifica relazione di trust).
6. Verificare che nella policy sia elencato l'ID dell'account di monitoraggio o l'ID di un'organizzazione contenente l'account di monitoraggio.

Non viene visualizzato un elenco a discesa degli account nella console

Innanzitutto, verificare di aver creato i ruoli IAM corretti, come illustrato nella sezione relativa alla risoluzione dei problemi precedente. Se questi sono impostati correttamente, assicurarsi di aver abilitato l'account per visualizzare i dati su più account, come descritto in Enable Your Account to View Cross-Account Data.

Disabilitazione e pulizia dopo l'utilizzo di più account

Per disabilitare la funzionalità su più account per CloudWatch, segui questi passaggi.

Passaggio 1: rimuovere gli stack o i ruoli di più account

Il metodo migliore consiste nel rimuovere gli AWS CloudFormation stack utilizzati per abilitare la funzionalità tra più account.

- In ciascuno degli account di condivisione, rimuovi lo stack CloudWatch- CrossAccountSharingRole.

- Se prima abilitavi AWS Organizations la funzionalità tra account con tutti gli account di un'organizzazione, rimuovi lo CrossAccountListAccountsRole stack CloudWatch- nell'account di gestione dell'organizzazione.

Se non hai utilizzato gli AWS CloudFormation stack per abilitare la funzionalità tra più account, procedi come segue:

- In ciascuno degli account di condivisione, elimina il ruolo CloudWatch- CrossAccountSharingRole IAM.
- Se prima abilitavi AWS Organizations la funzionalità tra account con tutti gli account di un'organizzazione, elimina il ruolo CloudWatch- CrossAccountSharing - ListAccountsRole IAM nell'account di gestione dell'organizzazione.

Passaggio 2: rimuovere il ruolo collegato ai servizi

Nell'account di monitoraggio, elimina il ruolo IAM AWSServiceRoleForCloudWatchCrossAccountcollegato al servizio.

CloudWatch quote di servizio

CloudWatch ha le seguenti quote per metriche, allarmi, richieste API e notifiche e-mail di allarme.

> **Note**
>
> Per alcuni AWS servizi CloudWatch, tra cui, puoi utilizzare le metriche di CloudWatch utilizzo per visualizzare l'utilizzo corrente del servizio su grafici e dashboard. CloudWatch È possibile utilizzare una funzione matematica CloudWatch metrica per visualizzare le quote di servizio per tali risorse sui grafici. È inoltre possibile configurare gli allarmi che avvisano quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni, consulta Visualizzazione delle quote di servizio e impostazione degli allarmi.

Risorsa	Quota predefinita
Operazione per gli allarmi	5 ad allarme. Questa quota non può essere modificata.
Periodo di valutazione dell'allarme	Il valore massimo, calcolato moltiplicando il periodo di allarme per il numero di periodi di valutazione utilizzati, è di un giorno (86.400 secondi). Questa quota non può essere modificata.
Allarmi	10 al mese a cliente gratuiti. Ulteriori allarmi sono soggetti a costi aggiuntivi. Nessun limite al numero totale di allarmi per account. Gli allarmi basati su espressioni matematiche per i parametri possono avere fino a 10 parametri. 200 allarmi Metrics Insights per regione. È possibile richiedere un aumento della quota.
Modelli di rilevamento delle anomalie	500 per regione per account.
Richieste API	1.000.000 al mese a cliente gratuiti.
Canary	200 per regione per account.

Risorsa	Quota predefinita
	È possibile richiedere un aumento della quota.
Richieste API di Contributor Insights	Le API seguenti dispongono di una quota di 20 transazioni al secondo (TPS) e per regione. • DescribeInsightRules La quota non può essere modificata. • GetInsightRuleReport È possibile richiedere un aumento della quota. Le API seguenti dispongono di una quota di 5 TPS per regione. Questa quota non può essere modificata. • DeleteInsightRules • PutInsightRule Le API seguenti dispongono di una quota di 1 TPS per regione. Questa quota non può essere modificata. • DisableInsightRules • EnableInsightRules
Regole Contributor Insights	100 regole per regione per account. È possibile richiedere un aumento della quota.
Parametri personalizzati	Nessuna quota.

Risorsa	Quota predefinita
Pannelli di controllo	Fino a 500 widget per pannello di controllo. Fino a 500 parametri per widget del pannello di controllo. Fino a 2500 parametri per pannello di controllo, in tutti i widget. Queste quote includono tutti i parametri recuperati per l'uso nelle funzioni matematiche dei parametri, anche se tali parametri non sono visualizzati nel grafico. Queste quote non possono essere modificate.
DescribeAlarms	9 transazioni al secondo (TPS) per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. È possibile richiedere un aumento della quota.
Richiesta DeleteAlarms Richiesta DescribeAlarmHistory Richiesta DisableAlarmActions Richiesta EnableAlarmActions Richiesta SetAlarmState	3 TPS per Regione per ciascuna di queste operazioni. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. Queste quote non possono essere modificate.
Richiesta DescribeAlarmsForMetric	9 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. Queste quote non possono essere modificate.
Richiesta DeleteDashboards Richiesta GetDashboard Richiesta ListDashboards Richiesta PutDashboard	10 TPS per Regione per ciascuna di queste operazioni. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. Queste quote non possono essere modificate.

Risorsa	Quota predefinita
PutAnomalyDetector DescribeAnomalyDetectors	10 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling.
DeleteAnomalyDetector	5 TPS per regione Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling.
Dimensioni	30/parametro. Questa quota non può essere modificata.

Risorsa	Quota predefinita
GetMetricData	10 TPS per regione per operazioni che includono query Metrics Insights. Per le operazioni che non includono query Metrics Insights, la quota è di 50 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a limitazione (della larghezza di banda della rete). È possibile richiedere un aumento della quota. Per operazioni `GetMetricData` che includono una query Metrics Insights, la quota è di 4.300.000 punti dati al secondo (DPS) per le 3 ore più recenti. Questo viene calcolato in base al numero totale di punti dati scansionati dalla query (che non può includere più di 10.000 parametri). 180.000 Datapoint Per Secondo (DPS) se `StartTime` utilizzato nella richiesta API è inferiore o uguale a tre ore dall'orario attuale. 396.000 DPS se `StartTime` è più di tre ore dall'orario attuale. Questo è il numero massimo di datapoint che è possibile richiedere al secondo utilizzan do una o più chiamate API senza essere sottoposti a throttling. Questa quota non può essere modificata. Il DPS viene calcolato in base ai punti dati stimati, non ai punti dati effettivi. La stima del punto dati viene calcolata utilizzando l'intervallo di tempo, il periodo e il periodo di conservazione richiesti. Ciò significa che se i punti dati effettivi nei parametri richiesti sono scarsi o vuoti, la limitazione si verifica ancora se i punti dati stimati superano la quota. La quota DPS è per regione.

Risorsa	Quota predefinita
GetMetricData	Una singola chiamata `GetMetricData` può includere quanto segue: • Fino a 500 strutture `MetricDataQuery` . • Fino a 100 funzioni `SERVICE_QUOTA()` . • Fino a 100 funzioni `SEARCH()`. • Fino a 5 funzioni `LAMBDA()`. Queste quote non possono essere modificate.
GetMetricStatistics	400 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. È possibile richiedere un aumento della quota.
GetMetricWidgetImage	Fino a 500 parametri per immagine. Questa quota non può essere modificata. 20 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. È possibile richiedere un aumento della quota.
ListMetrics	25 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. È possibile richiedere un aumento della quota.
Valori dei dati delle metriche	Il valore di un punto dati della metrica deve essere compreso nell'intervallo da -2^360 a 2^360. I valori speciali (ad esempio NaN, +Infinity, -Infinity) non sono supportati. Questa quota non può essere modificata.

Risorsa	Quota predefinita
MetricDatumarticoli	PutMetricData1000/richiesta. Un MetricDatumoggetto può contenere un singolo valore o un StatisticSetoggetto che rappresenta molti valori. Questa quota non può essere modificata.
Metriche	10 al mese a cliente gratuiti.
Query di Informazioni dettagliate sui parametri	Una singola query può elaborare non più di 10.000 parametri. Ciò significa che se le clausole SELEZIONA RE,DA, eDOVEcorrispondono a più di 10.000 parametri , solo i primi 10.000 di questi parametri trovati verranno elaborati dalla query. Una singola query non può restituire più di 500 serie temporali. Puoi eseguire query solo per le ultime tre ore di dati
Frequenze di richiesta dell'API Observability Access Manager (OAM).	1 TPS per regione per. PutSinkPolicy 10 TPS per regione per ogni altra API CloudWatch OAM. Queste quote riflettono il numero massimo di richieste di operazioni al secondo che è possibile effettuare al secondo senza essere sottoposti a limitazione. Queste quote non possono essere modificate.
Collegamenti agli account di origine OAM	Ogni account di origine può essere collegato a un massimo di cinque account di monitoraggio Questa quota non può essere modificata.
Sink OAM	1 sink per regione per account Questa quota non può essere modificata.

Risorsa	Quota predefinita
Richiesta PutCompositeAlarm	3 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. È possibile richiedere un aumento della quota.
Richiesta PutMetricAlarm	3 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. È possibile richiedere un aumento della quota.
Richiesta PutMetricData	1 MB per le richieste POST HTTP. PutMetricDataè in grado di gestire 500 transazioni al secondo (TPS), ovvero il numero massimo di richieste operative che è possibile effettuare al secondo senza subire limitazioni. `PutMetricData` è in grado di gestire 1.000 parametri per richiesta. È possibile richiedere un aumento della quota.
E-mail di notifica Amazon SNS	1.000 al mese a cliente gratuite.
Gruppi Synthetics	20 per account. Questa quota non può essere modificata.
TagResource	20 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. Questa quota non può essere modificata.
UntagResource	20 TPS per regione. Il numero massimo di richieste di operazioni al secondo che è possibile effettuare senza essere sottoposti a throttling. Questa quota non può essere modificata.

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti in ogni versione della Amazon CloudWatch User Guide, a partire da giugno 2018. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
CloudWatch La mappa dei servizi di Application Signals supporta i client Canary, RUM e i raggruppamenti AWS di dipendenze dei servizi.	La versione di anteprima di Application Signals ha aggiunto raggruppamenti predefiniti nella mappa dei servizi per canaries, client RUM e dipendenze di AWS servizio dello stesso tipo. Questa modifica riduce il numero di icone nella visualizz azione predefinita della mappa dei servizi per semplificare la visualizzazione e la navigazio ne.	21 maggio 2024
CloudWatchReadOnlyAccess Politica IAM aggiornata	CloudWatch ha modificato l'ambito di un'autorizzazione in CloudWatchReadOnly Access. L'ambito della politica ha aggiunto le `applicati on-signals:BatchGe t* application-signal s:List*` azioni necessari e per consentire agli utenti di utilizzare CloudWatch Applicati on Signals per visualizzare, esaminare e diagnosticare problemi relativi allo stato dei propri servizi. `applicati`	17 maggio 2024

	`on-signals:Get*` CloudWatchha inoltre aggiunto un'`iam:GetRole` azione in modo che gli utenti possano verificare se Application Signals è configurato.	
CloudWatchFullAccessPolitica IAM V2 aggiornata	CloudWatch ha modificato l'ambito di un'autorizzazione in CloudWatchFullAccessV2. L'ambito della politica è stato aggiunto `application-signals:*` in modo che gli utenti possano utilizzar e CloudWatch Application Signals per visualizzare, esaminare e diagnosticare problemi relativi allo stato dei propri servizi.	17 maggio 2024
Lambda Insights supporta AWS GovCloud (Stati Uniti orientali) e AWS GovCloud (Stati Uniti occidentali)	CloudWatch Lambda Insights ha aggiunto il supporto per le regioni AWS GovCloud (Stati Uniti orientali) e AWS GovCloud (Stati Uniti occidentali).	29 aprile 2024
CloudWatch l'osservabilità tra account supporta i filtri delle risorse	Ora puoi creare filtri per specificare quali namespace delle metriche e gruppi di log sono condivisi dall'account di origine all'account di monitorag gio, quando crei il collegame nto tra gli account.	26 aprile 2024

CloudWatch Aggiornamenti di Application Signals	La versione di anteprima di Application Signals ha aggiunto tre funzionalità. Application Signals ora supporta le applicazioni Python. Offre un processo di abilitazione più semplice per le applicazioni su architetture Amazon EKS. Inoltre, include nuove configurazioni che puoi utilizzare per gestire la cardinalità delle metriche raccolte.	26 aprile 2024
CloudWatch Container Insights con osservabilità migliorata per Amazon EKS può raccogliere metriche AWS Elastic Fabric Adapter (EFA)	Ora puoi utilizzare CloudWatch Container Insights con una migliore osservabilità per Amazon EKS e raccogliere metriche AWS Elastic Fabric Adapter (EFA) dai cluster Amazon EKS.	23 aprile 2024

Politica IAM aggiornata	CloudWatch ha aggiornato la CloudWatchApplicationSignalsServiceRolePolicypolitica. L'ambito `logs:StartQuery` e le `logs:GetQueryResults` autorizzazioni di questa politica sono stati modificati per aggiungere `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` e `"arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"` abilitare Application Signals su più architetture. Questa policy è associata al ruolo collegato al servizio. AWSServiceRoleForCloudWatchApplicationSignals	18 aprile 2024
Internet Monitor fornisce una mappa meteorologica globale di Internet ai clienti autenticati AWS	Amazon CloudWatch Internet Monitor ora mostra una mappa meteorologica globale di Internet disponibile nella console per tutti i AWS clienti autenticati. Per visualizzare la mappa, nella CloudWatch console Amazon, accedi a Internet Monitor.	16 aprile 2024
CloudWatch Container Insights con osservabilità migliorata per Amazon EKS può raccogliere metriche AWS Neuron	Ora puoi utilizzare CloudWatch Container Insights con osservabilità migliorata per Amazon EKS e raccogliere metriche AWS Neuron dai cluster Amazon EKS.	16 aprile 2024

CloudWatch Application Signals aggiunge una scheda di panoramica dei servizi e altre metriche per facilitare la diagnostica	Una nuova scheda Panoramica dei servizi mostra una panoramica del servizio, incluso il numero di operazioni, le dipendenze, i materiali sintetici e le pagine dei client. La scheda mostra le metriche chiave per l'intero servizio e le principali operazioni e dipendenze. È inoltre possibile visualizzare le tracce a raggi X correlate a problemi quali guasti, errori e problemi di latenza.	16 aprile 2024
CloudWatch Container Insights con osservabilità migliorata per Amazon EKS aggiunge il supporto per Windows	Ora puoi utilizzare CloudWatch Container Insights con osservabilità migliorata per Amazon EKS per raccogliere metriche dai nodi di lavoro Windows sui cluster Amazon EKS.	10 aprile 2024
CloudWatchApplicationSignalsServiceRolePolicyPolitica IAM aggiornata	CloudWatch ha modificato l'ambito di un'autorizzazione in CloudWatchApplicationSignalsServiceRolePolicy. L'ambito dell'`cloudwatch:GetMetricData` autorizzazione è stato modificato `*` in modo che Application Signals possa recuperare le metriche dalle fonti negli account collegati.	8 aprile 2024

Amazon CloudWatch Internet Monitor ora supporta l'osservabilità tra più account	Ora puoi utilizzare l'osservabilità tra più account di Internet Monitor per monitorare le tue applicazioni che si estendono su più account all'interno di una singola. Account AWS Regione AWS	29 marzo 2024
CloudWatchAgentServerPolicy e CloudWatchAgentAdminPolicypolitiche aggiornate	CloudWatch ha aggiunto autorizzazioni a entrambe le `CloudWatchAgentAdminPolicy` policy `CloudWatchAgentServerPolicy` e per consentire all' CloudWatch agente di pubblicare tracce X-Ray e modificare i periodi di conservazione dei gruppi di log. In entrambe le politiche, sono state `xray:PutTraceSegments` aggiunte le`xray:PutTelemetryRecords` ,`xray:GetSamplingRules` ,`xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` e le `logs:PutRetentionPolicy` autorizzazioni	12 febbraio 2024

Nuovo ruolo collegato al servizio e policy IAM per CloudWatch Network Monitor	CloudWatch ha aggiunto un nuovo ruolo collegato al servizio, chiamato. AWSServiceRoleForNetworkMonitor CloudWatch ha aggiunto questo nuovo ruolo collegato ai servizi per consentire di creare monitor per recuperare le metriche di rete tra le sottoreti di origine e gli indirizzi IP di destinazione. La nuova policy CloudWatchNetworkMonitorServiceRolePolicyIAM è associata a questo ruolo e la policy concede l'autorizzazione a recuperare le metriche di rete per CloudWatch tuo conto.	22 dicembre 2023
CloudWatch rilascia Amazon CloudWatch Network Monitor	CloudWatch ha rilasciato una nuova funzionalità, Amazon CloudWatch Network Monitor. Si tratta di un nuovo servizio di monitoraggio attivo della rete che identifica se esistono problemi di rete all'interno della AWS rete o della rete aziendale.	22 dicembre 2023

CloudWatchReadOnly Accesspolitica aggiornata	CloudWatch ha aggiunto autorizzazioni di sola lettura esistenti per CloudWatch Synthetics, X-Ray CloudWatch e RUM e nuove autorizzazioni di sola lettura CloudWatch per Application Signals in modo che gli utenti con questa politica possano valutare e diagnosticare i problemi di integrità del servizio come riportato CloudWatchReadOnlyAccessda Application Signals. CloudWatch L'`cloudwatch:GenerateQuery` autorizzazione è stata aggiunta in modo che gli utenti con questa politica possano generare una stringa di query di Metrics Insights da un prompt in linguaggio naturale. CloudWatch	5 dicembre 2023

CloudWatchFullAccessCriterioV2 aggiornato	CloudWatch ha aggiunto le autorizzazioni esistenti a CloudWatchFullAccessV2 per Synthetics CloudWatch , X-Ray e CloudWatch RUM e ha aggiunto nuove autorizzazioni per CloudWatch Application Signals in modo che gli utenti con questa politica possano gestire completamente Application Signals per valutare e diagnosticare i problemi relativi allo stato del servizio.	5 dicembre 2023

Nuovo ruolo collegato al servizio e nuova policy IAM	CloudWatch ha aggiunto un nuovo ruolo collegato al servizio, chiamato. AWSServiceRoleForCloudWatchApplicationSignals CloudWatch ha aggiunto questo nuovo ruolo collegato al servizio per consentire ad CloudWatch Application Signals di raccogliere dati di CloudWatch log, dati di traccia a raggi X, dati di CloudWatch metrica e dati di etichettatura dalle applicazioni abilitate per Application Signals. CloudWatch La nuova policy CloudWatchApplicationSignalsServiceRolePolicyIAM è associata a questo ruolo e la policy concede il permesso ad CloudWatch Application Signals di raccogliere dati di monitoraggio e tagging da altri servizi pertinenti. AWS	30 novembre 2023

CloudWatch lancia la versione di anteprima di Application Signals	CloudWatch Application Signals è in anteprima. Utilizzate Application Signals per strumentare le vostre applicazioni AWS in modo da monitorare lo stato attuale delle applicazioni, creare obiettivi di livello di servizio (SLO) e monitorare le prestazioni delle applicazioni a lungo termine rispetto agli obiettivi aziendali. Per ulteriori informazioni, consulta Application Signals.	30 novembre 2023
CloudWatch aggiunge il supporto per l'interrogazione di altre fonti di dati	Puoi utilizzarlo CloudWatch per interrogare, visualizzare e creare allarmi per metriche provenienti da altre fonti di dati. Per ulteriori informazioni, consulta Interrogazione di metriche da altre fonti di dati.	26 novembre 2023
CloudWatch Metrics Insights supporta la generazione di query in linguaggio naturale	CloudWatch Metrics Insights supporta le query in linguaggio naturale per generare e aggiornare le query. Per ulteriori informazioni, consulta Utilizzare il linguaggio naturale per generare e aggiornare le query di CloudWatch Metric Insights.	26 novembre 2023

CloudWatch rilascia Container Insights con osservabilità migliorata per Amazon EKS	CloudWatch ha rilasciato una nuova versione di Container Insights. Questa versione supporta l'osservabilità migliorata per i cluster Amazon EKS e può raccogliere parametri più dettagliati dai cluster in esecuzione su Amazon EKS. Dopo l'installazione, raccoglie automaticamente la telemetria dettagliata dell'infrastruttura e i log dei container per i cluster Amazon EKS. Puoi quindi impiegare pannelli di controllo accurati e immediatamente utilizzabili per approfondire la telemetria delle applicazioni e dell'infrastruttura.	6 novembre 2023
CloudWatch metric streams aggiunge una configurazione rapida dei partner	CloudWatch metric streams ora offre un'opzione di configurazione rapida per i partner, che puoi utilizzare per configurare rapidamente un flusso di metriche per alcuni provider di terze parti.	17 ottobre 2023

CloudWatch rilascia raccomandazioni sugli allarmi	CloudWatch Synthetics ora fornisce consigli sugli allarmi per le metriche di altri servizi. AWS Queste raccomandazioni possono aiutarti a identificare i parametri per i quali attivare gli allarmi, in linea con le best practice per il monitoraggio di questi servizi.	16 ottobre 2023
CloudWatch Synthetics rilascia il runtime -6.0 syn-nodejs-puppeteer	CloudWatch Synthetics ha rilasciato il runtime. `syn-nodejs-puppeteer-6.0`	26 settembre 2023
Aggiunge il supporto di Amazon CloudWatch Application Insights per applicazioni cross-account	Ora puoi condividere le applicazioni CloudWatch Application Insights oltre i confini degli account.	26 settembre 2023

Nuovo ruolo collegato al servizio e nuova policy IAM	CloudWatch ha aggiunto un nuovo ruolo collegato al servizio, chiamato. AWSServiceRoleForCloudWatchMetrics_DbPerfInsights CloudWatch ha aggiunto questo nuovo ruolo collegato ai servizi per consentire di recuperare le metriche CloudWatch di Performance Insights per allarmi, rilevamento di anomalie e snapshot. La nuova policy AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicyIAM è associata a questo ruolo e la policy concede l'autorizzazione CloudWatch a recuperare le metriche di Performance Insights per tuo conto.	20 settembre 2023
Aggiunge una nuova funzione matematica con parametri	CloudWatch ha aggiunto una nuova funzione matematica delle metriche`DB_PERF_INSIGHTS` , che puoi utilizzare per recuperare le metriche di Performance Insights dai servizi di AWS database per allarmi, rilevamento di anomalie e snapshot.	20 settembre 2023

CloudWatchReadOnly Accesspolitica aggiornata	CloudWatch ha aggiunto l'`application-autoscaling:DescribeScalingPolicies` autorizzazione a CloudWatchReadOnly Accessin modo che gli utenti con questo criterio possano accedere alle informazioni sulle politiche di Application Auto Scaling.	14 settembre 2023
CloudWatch l'agente ha aggiunto il supporto per AL2023	L' CloudWatch agente supporta AL2023.	8 agosto 2023
Nuova politica IAM gestita, CloudWatchFullAccess V2	CloudWatch ha aggiunto una nuova politica CloudWatchFullAccessV2. Questa politica garantisce l'accesso completo alle CloudWatch azioni e alle risorse, definendo al contempo meglio le autorizzazioni concesse ad altri servizi come Amazon SNS e. Amazon EC2 Auto Scaling	1° agosto 2023
Ruolo collegato al servizio aggiornato per Amazon CloudWatch Internet Monitor: aggiornamento a una policy esistente	Aggiunge nuove autorizzazioni, `elasticloadbalancing:DescribeLoadBalancers` e `ec2:DescribeNetworkInterfaces`, al ruolo collegato al servizio per Monitor Internet, per supportare il monitoraggio del traffico per risorse specifiche di Network Load Balancer.	25 luglio 2023

È stato aggiunto il supporto per le risorse Network Load Balancer in Amazon Internet Monitor CloudWatch	Aggiunge il supporto per la creazione di un monitor in Monitor Internet con risorse specifiche di Network Load Balancer, per fornire livelli di osservabilità più granulari per l'applicazione.	25 luglio 2023
Funzionalità delle variabili del pannello di controllo	CloudWatch ha rilasciato variabili di dashboard, che puoi utilizzare per creare dashboard flessibili in grado di visualizzare rapidamente contenuti diversi a seconda di come imposti un campo di input all'interno della dashboard. Ad esempio, puoi creare una dashboard in grado di passare rapidamen te tra diverse funzioni Lambda o ID di istanza Amazon EC2, oppure una che può passare a regioni diverse. AWS Per ulteriori informazi oni, consulta Creazione di pannelli di controllo flessibili con le variabili del pannello di controllo.	28 giugno 2023

Monitor Internet ora supporta la personalizzazione della soglia per gli eventi di integrità	Monitor Internet ha aggiunto la possibilità di personalizzare la soglia per quando un punteggio di prestazione globale o un punteggio di disponibilità attivano un evento di integrità. Per ulteriori informazioni, consulta Monitoraggio delle prestazioni e della disponibilità in tempo reale in Amazon CloudWatch Internet Monitor.	26 giugno 2023
Monitor Internet ora supporta tutte le Regioni commerciali	Internet Monitor ne ha aggiunti sette nuovi Regioni AWS e ora supporta tutte le regioni commerciali.	19 giugno 2023
Nuove versioni dell'estensione Lambda Insights	CloudWatch ha aggiunto la versione 1.0.229.0 dell'estensione Lambda Insights per piattaforme x86-64 e piattaforme ARM64. Per ulteriori informazioni, consulta Versioni disponibili dell'estensione Lambda Insights.	12 giugno 2023

CloudWatchReadOnly Accesspolitica aggiornata	CloudWatch autorizzazioni aggiunte a. CloudWatchReadOnlyAccess Le `logs:StopLiveTail` autorizzazioni `logs:StartLiveTail` e sono state aggiunte in modo che gli utenti con questo criterio possano utilizzare la console per avviare e interrompere le sessioni live tail di CloudWatch Logs. Per ulteriori informazioni, consulta Use live tail to view logs in near real time.	6 giugno 2023
CloudWatch RUM aggiunge il supporto per metriche personalizzate	Puoi utilizzare i monitor delle app CloudWatch RUM per creare metriche personalizzate e inviarle a CloudWatch ed Evidently. CloudWatch Questa funzionalità include un aggiornamento della politica IAM ServiceRolePolicy gestita da AmazonCloudWatchRUM. In quella policy, una chiave di condizione è stata modificata in modo che CloudWatch RUM possa inviare metriche personalizzate a namespace di metriche personalizzate.	9 febbraio 2023

Policy gestite nuove e aggiornate per CloudWatch	Per supportare l'osservabilità CloudWatch tra più account, le `CloudWatchReadOnlyAccess` politiche `CloudWatchFullAccess` e sono state aggiornate e sono state aggiunte le seguenti nuove politiche gestite:`CloudWatchCrossAccountSharingConfiguration` ,, `OAMFullAccess` e. `OAMReadOnlyAccess` Per ulteriori informazioni, consulta CloudWatch gli aggiornamenti alle politiche AWS gestite.	7 febbraio 2023
CloudWatch Aggiornamenti delle policy relative ai ruoli collegati ai servizi di Application Insights: aggiornamento a una policy esistente.	CloudWatch Application Insights ha aggiornato una politica esistente relativa ai ruoli AWS collegati ai servizi.	19 dicembre 2022
Supporto Amazon CloudWatch Application Insights per applicazioni e microservizi containerizzati dalla console Container Insights.	Puoi visualizzare i problemi rilevati da CloudWatch Application Insights per Amazon ECS e Amazon EKS sulla dashboard di Container Insights.	17 novembre 2021
Monitoraggio di Amazon CloudWatch Application Insights per database SAP HANA.	È possibile monitorare i database SAP HANA con Application Insights.	15 novembre 2021

Supporto Amazon CloudWatch Application Insights per il monitoraggio di tutte le risorse di un account.	È possibile effettuare l'accesso e il monitoraggio di tutte le risorse di un account.	15 settembre 2021
Supporto di Amazon CloudWatch Application Insights per Amazon FSx.	Puoi monitorare i parametri recuperati da Amazon FSx.	31 agosto 2021
SDK Metrics non è più supportato.	CloudWatch SDK Metrics non è più supportato.	25 agosto 2021
Supporto di Amazon CloudWatch Application Insights per la configurazione del monitoraggio dei container.	Puoi monitorare i contenitori utilizzando le best practice con Amazon CloudWatch Application Insights.	18 maggio 2021
I flussi di parametri sono disponibili al pubblico	Puoi utilizzare i flussi metrici per trasmettere continuamente i CloudWatch parametri verso una destinazione di tua scelta. Per ulteriori informazioni, consulta Metric streams nella Amazon CloudWatch User Guide.	31 marzo 2021
Monitoraggio di Amazon CloudWatch Application Insights per database Oracle su Amazon RDS e Amazon EC2.	Puoi monitorare le metriche e i log recuperati da Oracle con Amazon CloudWatch Application Insights.	16 gennaio 2021

Lambda Insights è disponibile al pubblico	CloudWatch Lambda Insights è una soluzione di monitoraggio e risoluzione dei problemi per applicazioni serverless in esecuzione su. AWS Lambda Per ulteriori informazioni, consulta Using Lambda Insights nella Amazon CloudWatch User Guide.	3 dicembre 2020
Monitoraggio di Amazon CloudWatch Application Insights per le metriche degli esportatori Prometheus JMX.	Puoi monitorare le metriche recuperate dall'esportatore Prometheus JMX con Amazon Application Insights. CloudWatch	20 novembre 2020
CloudWatch Synthetics rilascia una nuova versione di runtime	CloudWatch Synthetics ha rilasciato una nuova versione di runtime. Per ulteriori informazioni, consulta Canary Runtime Versions nella Amazon CloudWatch User Guide.	11 settembre 2020
Monitoraggio di Amazon CloudWatch Application Insights per Postgre SQL su Amazon RDS e Amazon EC2.	Puoi monitorare le applicazioni create con PostgreSQL in esecuzione su Amazon RDS o Amazon EC2.	11 settembre 2020

CloudWatch supporta la condivisione di dashboard	Ora puoi condividere le CloudWatch dashboard con persone esterne alla tua organizzazione e al tuo AWS account. Per ulteriori informazioni, consulta Sharing CloudWatch Dashboards nella Amazon CloudWatch User Guide.	10 settembre 2020
Configura i monitor per le applicazioni.NET utilizzando SQL Server sul backend con Application Insights CloudWatch	Puoi utilizzare il tutorial della documentazione per aiutarti a configurare i monitor per le applicazioni.NET utilizzando SQL Server sul backend con Application Insights. CloudWatch	19 agosto 2020
AWS CloudFormation supporto per le applicazioni Amazon CloudWatch Application Insights.	Puoi aggiungere il monitoraggio di CloudWatch Application Insights, comprese le metriche chiave e la telemetria, all'applicazione, al database e al server Web, direttamente dai modelli. AWS CloudFormation	30 luglio 2020
Monitoraggio di Amazon CloudWatch Application Insights per Aurora per cluster di database MySQL.	Puoi monitorare i cluster di database Aurora for MySQL (RDS Aurora) con Amazon Application Insights. CloudWatch	2 luglio 2020

CloudWatch Disponibilità generale di Contributor Insights	CloudWatch Contributor Insights è ora disponibile a tutti. Consente di analizzare i dati di log registro e creare serie temporali che visualizzino i dati dei collaboratori. Puoi visualizzare i parametri relative ai primi N collaboratori, al numero totale di collaboratori univoci e al loro utilizzo. Per ulteriori informazioni, consulta Using Contributor Insights per analizzare dati ad alta cardinalità nella Amazon CloudWatch User Guide.	2 aprile 2020
CloudWatch Anteprima pubblica di Synthetics	CloudWatch Synthetics è ora disponibile in anteprima pubblica. Consente di creare Canary per monitorare gli endpoint e le API. Per ulteriori informazioni, consulta Using Canaries nella Amazon CloudWatch User Guide.	25 novembre 2019

CloudWatch Anteprima pubblica di Contributor Insights	CloudWatch Contributor Insights è ora disponibile in anteprima pubblica. Consente di analizzare i dati di log registro e creare serie temporali che visualizzino i dati dei collaboratori. Puoi visualizzare i parametri relative ai primi N collaboratori, al numero totale di collaboratori univoci e al loro utilizzo. Per ulteriori informazioni, consulta Using Contributor Insights per analizzare dati ad alta cardinalità nella Amazon CloudWatch User Guide.	25 novembre 2019
CloudWatch lancia una funzionalità ServiceLens	ServiceLens migliora l'osservabilità dei servizi e delle applicazioni consentendoti di integrare tracce, metriche, registri e allarmi in un unico posto. ServiceLens si integra CloudWatch con AWS X-Ray per fornire una visualizzazione dell'applicazione. end-to-end	21 novembre 2019

Utilizzatelo CloudWatch per gestire in modo proattivo le AWS quote di servizio	È possibile utilizzarlo CloudWatch per gestire in modo proattivo le AWS quote di servizio. CloudWatch le metriche di utilizzo forniscono visibilità sull'utilizzo delle risorse e sulle operazioni API da parte dell'account. Per ulteriori informazioni, consulta Service Quotas Integration and Usage Metrics nella Amazon CloudWatch User Guide.	19 novembre 2019
CloudWatch invia eventi quando gli allarmi cambiano stato	CloudWatch ora invia un evento ad Amazon EventBridge quando un CloudWatch allarme cambia stato. Per ulteriori informazioni, consulta Alarm Events e EventBridge nella Amazon CloudWatch User Guide.	8 ottobre 2019
Container Insights	CloudWatch Container Insights è ora disponibile al pubblico. Consente di raccogliere, aggregare e riepilogare parametri e log dalle applicazioni e dai microservizi containerizzati. Per ulteriori informazioni, consulta Using Container Insights nella Amazon CloudWatch User Guide.	30 agosto 2019

Aggiornamenti per i parametri di anteprima di Container Insights su Amazon EKS e Kubernetes	L'anteprima pubblica di Container Insights su Amazon EKS e Kubernetes è stata aggiornata. InstanceId è ora inclusa come dimensione nelle istanze EC2 del cluster. Ciò consente agli allarmi che sono stati creati su questi parametri di attivare le seguenti operazioni EC2: Stop (Arresto), Terminate (Termina) , Reboot (Riavvia) o Recover (Ripristina). Inoltre, i parametri relativi a pod e parametri sono ora segnalati dallo spazio dei nomi Kubernetes per semplific are il monitoraggio e gli allarmi sui parametri in base allo spazio dei nomi.	19 agosto 2019
Aggiornamenti per l'integra zione AWS Systems Manager OpsCenter	Aggiornamenti su come CloudWatch Application Insights si integra con Systems Manager OpsCenter.	7 agosto 2019
CloudWatch metriche di utilizzo	CloudWatch le metriche di utilizzo ti aiutano a tenere traccia dell'utilizzo delle tue CloudWatch risorse e a rimanere entro i limiti del servizio. Per ulteriori informazioni, consulta https:// docs.aws.amazon.com/ AmazonCloudWatch/latest/ monitoring/CloudWatch-Us age-Metrics.html.	6 agosto 2019

CloudWatch anteprima pubblica di Container Insights	CloudWatch Container Insights è ora disponibile in anteprima pubblica. Consente di raccogliere, aggregare e riepilogare parametri e log dalle applicazioni e dai microservizi containerizzati. Per ulteriori informazioni, consulta Using Container Insights nella Amazon CloudWatch User Guide.	9 luglio 2019
CloudWatch Anteprima pubblica di Anomaly Detection	CloudWatch il rilevamento delle anomalie è ora disponibi le in anteprima pubblica. CloudWatch applica algoritmi di apprendimento automatico ai dati passati di una metrica per creare un modello dei valori previsti della metrica. Puoi utilizzare questo modello per la visualizzazione e l'impostazione di allarmi. Per ulteriori informazioni, consulta Using CloudWatch Anomaly Detection nella Amazon CloudWatch User Guide.	9 luglio 2019

CloudWatch Application Insights per.NET e SQL Server	CloudWatch Application Insights for .NET e SQL Server facilita l'osservabilità per le applicazioni.NET e SQL Server. Può essere utile per configurare i migliori monitor per le risorse dell'applicazione, per analizzare i dati in modo continuo per rilevare problemi con le applicazioni.	21 giugno 2019
CloudWatch riorganizzazione della sezione degli agenti	La documentazione dell' CloudWatch agente è stata riscritta per migliorarne la chiarezza, in particolare per i clienti che utilizzano la riga di comando per installar e e configurare l'agente. Per ulteriori informazioni, consulta la sezione Raccolta di metriche e log da istanze Amazon EC2 e server locali con l' CloudWatch agente nella Amazon User Guide. CloudWatch	28 marzo 2019

Aggiunta della funzione RICERCA alle espressioni matematiche per i parametri	È ora possibile utilizzare una funzione RICERCA nelle espressioni matematiche per i parametri. Ciò consente di creare pannelli di controllo che si aggiornano automatic amente quando vengono create nuove risorse che corrispondono alla query di ricerca. Per ulteriori informazi oni, consulta Using Search Expressions in Graphs nella Amazon CloudWatch User Guide.	21 marzo 2019
AWS Metriche SDK per Enterprise Support	SDK Metrics ti aiuta a valutare lo stato dei tuoi AWS servizi e a diagnosticare la latenza causata dal raggiungimento dei limiti di utilizzo dell'acco unt o da un'interruzione del servizio. Per ulteriori informazi oni, consulta Monitorare le applicazioni utilizzando i parametri AWS SDK nella Amazon CloudWatch User Guide.	11 dicembre 2018
Allarmi basati su espressioni matematiche	CloudWatch supporta la creazione di allarmi basati su espressioni matematic he metriche. Per ulteriori informazioni, consulta Alarms on Math Expressions nella Amazon CloudWatch User Guide.	20 novembre 2018

Nuova home page CloudWatch della console	Amazon ha creato una nuova home page nella CloudWatch console, che mostra automaticamente le metriche e gli allarmi chiave per tutti i AWS servizi che stai utilizzando. Per ulteriori informazioni, consulta la sezione Getting Started with Amazon CloudWatch nella Amazon CloudWatch User Guide.	19 novembre 2018
AWS CloudFormation modelli per l' CloudWatch agente	Amazon ha caricato AWS CloudFormation modelli che puoi utilizzare per installare e aggiornare l' CloudWatchagente. Per ulteriori informazioni, consulta Install the CloudWatch Agent on New Instances Using AWS CloudFormation nella Amazon CloudWatch User Guide.	9 novembre 2018
Miglioramenti apportati all'agente CloudWatch	L' CloudWatch agente è stato aggiornato per funzionare con entrambi i protocolli StatsD e collectd. Ha migliorato anche il supporto tra account. Per ulteriori informazioni, consulta Recuperare metriche personalizzate con StatsD, Recuperare metriche personalizzate con collectd e Invio di metriche e log a un altro account nella Amazon User Guide. AWS CloudWatch	28 settembre 2018

Supporto per gli endpoint Amazon VPC	Ora puoi stabilire una connessione privata tra il tuo VPC e. CloudWatch Per ulteriori informazioni, consulta Using CloudWatch with Interface VPC Endpoints nella Amazon CloudWatch User Guide.	28 giugno 2018

La tabella seguente descrive importanti modifiche alla Amazon CloudWatch User Guide prima di giugno 2018.

Modifica	Descrizione	Data di rilascio
Matematica dei parametri	Ora puoi eseguire espressioni matematiche sulle CloudWatch metriche, producendo nuove serie temporali da aggiungere ai grafici della dashboard . Per ulteriori informazioni, consulta Utilizzare la matematica dei parametri.	4 aprile 2018
"M su N" allarmi	Ora è possibile configurare un allarme per il trigger basato su "M su N" punti di dati in un qualsiasi intervallo di valutazione dell'allarme. Per ulteriori informazioni, consulta Valutazione di un allarme.	8 dicembre 2017
CloudWatch agente	È stato rilasciato un nuovo CloudWatch agente unificato. Puoi utilizzare l'agente multipiattaforma unificato per raccogliere sia parametri di sistema che file di log personalizzati da istanze Amazon EC2 e da server locali. Il nuovo agente supporta Windows e Linux e consente di personalizzare i parametri raccolti, compresi i parametri delle sottorisorse, ad esempio i core per CPU. Per ulteriori informazioni, consulta Raccogli metriche, log e tracce con l'agente CloudWatch .	7 settembre 2017

Modifica	Descrizione	Data di rilascio
Parametri di gateway NAT	Aggiunti parametri per il gateway NAT di Amazon VPC	7 settembre 2017
Parametri ad alta risoluzione	Puoi ora configurare facoltativamente parametri personalizzati come parametri ad alta risoluzione, con una granularità di un secondo. Per ulteriori informazioni, consulta Parametri ad alta risoluzione.	26 luglio 2017
API del pannello di controllo	Ora puoi creare, modificare ed eliminare pannelli di controllo tramite API e AWS CLI. Per ulteriori informazioni, consulta Creazione di un pannello di controllo CloudWatch .	6 luglio 2017
AWS Direct Connect metriche	Sono state aggiunte metriche per. AWS Direct Connect	29 giugno 2017
Parametri VPN Amazon VPC	Aggiunti parametri per VPN di Amazon VPC	15 maggio 2017
AppStream Metriche 2.0	Aggiunte metriche per AppStream la versione 2.0.	8 marzo 2017
CloudWatch selettore di colori per console	Puoi ora scegliere il colore di ogni parametro sui widget del pannello di controllo. Per ulteriori informazioni, consulta Modificare un grafico su una dashboard CloudWatch .	27 febbraio 2017
Allarmi sui pannelli di controllo	Puoi ora aggiungere allarmi ai pannelli di controllo . Per ulteriori informazioni, consulta Aggiungere o rimuovere un widget di allarme da una CloudWatch dashboard .	15 febbraio 2017
Aggiunta di parametri per Amazon Polly	Aggiunti parametri per Amazon Polly.	1° dicembre 2016

Modifica	Descrizione	Data di rilascio
Aggiunti parametri per il servizio gestito da Amazon per Apache Flink	Aggiunti parametri per il servizio gestito da Amazon per Apache Flink.	1° dicembre 2016
Aggiunto supporto per le statistiche dei percentili	Puoi specificare qualsiasi percentile, utilizzando fino a due decimali (ad esempio, p95,45). Per ulteriori informazioni, consulta Percentili.	17 Novembre 2016
Aggiunta di parametri per Amazon Simple Email Service	Aggiunta di parametri per Amazon Simple Email Service.	2 Novembre 2016
Conservazione dei parametri aggiornata	Amazon CloudWatch ora conserva i dati delle metriche per 15 mesi anziché 14 giorni.	1 Novembre 2016
Aggiornata interfaccia della console dei parametri	La CloudWatch console viene aggiornata con miglioramenti alle funzionalità esistenti e nuove funzionalità.	1 Novembre 2016
Aggiunta di parametri per Amazon Elastic Transcoder	Aggiunta di parametri per Amazon Elastic Transcoder.	20 settembre 2016
Aggiunta di parametri per Amazon API Gateway	Aggiunta di parametri per Amazon API Gateway.	9 settembre 2016

Modifica	Descrizione	Data di rilascio
Sono state aggiunte metriche per AWS Key Management Service	Sono state aggiunte metriche per. AWS Key Management Service	9 settembre 2016
Aggiunti parametri per i nuovi servizi di bilanciamento del carico delle applicazioni supportati da Elastic Load Balancing	Aggiunti parametri per l'Application Load Balancer	11 agosto 2016
Aggiunti nuovi parametri NetworkPacketsIn e NetworkPacketsOut parametri per Amazon EC2	Aggiunti nuovi parametri NetworkPacketsIn e NetworkPacketsOut parametri per Amazon EC2.	23 marzo 2016
Aggiunti nuovi parametri per il parco istanze Spot di Amazon EC2	Aggiunti nuovi parametri per il parco istanze Spot di Amazon EC2.	21 marzo 2016

Modifica	Descrizione	Data di rilascio
Sono state aggiunte nuove metriche relative CloudWatch ai log	Sono state aggiunte nuove metriche di CloudWatch Logs.	10 marzo 2016
OpenSearch Servizio Amazon e AWS WAF metriche e dimensioni aggiunti	Sono stati aggiunti Amazon OpenSearch Service e AWS WAF metriche e dimensioni.	14 ottobre 2015
È stato aggiunto il supporto per i dashboard CloudWatch	Le dashboard sono home page personalizzabili della CloudWatch console che puoi utilizzare per monitorare le risorse in un'unica visualizzazione, anche quelle distribuite in diverse regioni. Per ulteriori informazioni, consulta Utilizzo delle CloudWatch dashboard di Amazon.	8 ottobre 2015
AWS Lambda Metriche e dimensioni aggiunte	AWS Lambda Metriche e dimensioni aggiunte.	4 settembre 2015
Aggiunti dimensioni e parametri e Amazon Elastic Container Service	Aggiunti dimensioni e parametri e Amazon Elastic Container Service.	17 agosto 2015

Modifica	Descrizione	Data di rilascio
Aggiunti dimensioni e parametri e Amazon Simple Storage Service	Aggiunti dimensioni e parametri e Amazon Simple Storage Service.	26 luglio 2015
Nuova caratteristica: operazione di allarme di riavvio	Sono stati aggiunti l'operazione di allarme di riavvio e un nuovo ruolo IAM per l'uso con operazioni di allarme. Per ulteriori informazioni, consulta Creazione di allarmi per arrestare, terminare, riavviare o recuperare un'istanza EC2.	23 luglio 2015
Aggiunte WorkSpaces metriche e dimensioni di Amazon	Aggiunte WorkSpaces metriche e dimensioni di Amazon.	30 Aprile 2015
Aggiunti parametri e dimensioni di Amazon Machine Learning	Aggiunti parametri e dimensioni di Amazon Machine Learning.	9 aprile 2015
Nuova caratteristica: operazioni di allarme di recupero istanze Amazon EC2	Operazioni di allarme di aggiornamento per includere l'operazione di recupero delle istanze EC2. Per ulteriori informazioni, consulta Creazione di allarmi per arrestare, terminare, riavviare o recuperare un'istanza EC2.	12 marzo 2015

Modifica	Descrizione	Data di rilascio
Aggiunte CloudSearch metriche CloudFront e dimensioni di Amazon e Amazon	Aggiunte CloudSearch metriche CloudFront e dimensioni di Amazon e Amazon.	6 marzo 2015
Aggiunti dimensioni e parametri e Amazon Simple Workflow Service	Aggiunti dimensioni e parametri e Amazon Simple Workflow Service.	9 maggio 2014
Guida aggiornata per aggiungere supporto per AWS CloudTrail	È stato aggiunto un nuovo argomento per spiegare come utilizzare AWS CloudTrail per registrare le attività in Amazon CloudWatch. Per ulteriori informazioni, consulta Registrazione delle chiamate CloudWatch API Amazon con AWS CloudTrail.	30 aprile 2014
Guida aggiornata all'uso del nuovo AWS Command Line Interface (AWS CLI)	La AWS CLI è una CLI multiservizio con un'installazione semplificata, una configurazione unificata e una sintassi della riga di comando coerente. La AWS CLI è supportata su Linux/Unix, Windows e Mac. Gli esempi di CLI in questa guida sono stati aggiornati per utilizzare la nuova AWS CLI. Per informazioni su come installare e configurare la nuova AWS CLI, consulta Getting Set Up with the AWS CLI Interface nella Guida per l'AWS Command Line Interface utente.	21 febbraio 2014

Modifica	Descrizione	Data di rilascio
Aggiunti Amazon Redshift e AWS OpsWorks metriche e dimensioni	Aggiunti Amazon Redshift e AWS OpsWorks metriche e dimensioni.	16 luglio 2013
Aggiunti parametri e dimensioni di Amazon Route 53	Aggiunti parametri e dimensioni di Amazon Route 53.	26 giugno 2013
Nuova funzionalità: Amazon CloudWatch Alarm Actions	È stata aggiunta una nuova sezione per documentare le azioni di CloudWatch allarme di Amazon, che puoi utilizzare per interrompere o terminare un'istanza Amazon Elastic Compute Cloud. Per ulteriori informazioni, consulta Creazione di allarmi per arrestare, terminare, riavviare o recuperare un'istanza EC2.	8 gennaio 2013
Aggiornati parametri di EBS	Aggiornati i parametri di EBS per includere due nuovi parametri per i volumi di Provisioned IOPS.	20 Novembre 2012
Nuovi avvisi di fatturazione	Ora puoi monitorare i tuoi AWS addebiti utilizzando i CloudWatch parametri di Amazon e creare allarmi per avvisarti quando hai superato la soglia specificata. Per ulteriori informazioni, consulta Crea un allarme di fatturazione per monitorare gli addebiti stimati AWS.	10 maggio 2012
Nuovi parametri	Puoi ora accedere a sei nuovi parametri Elastic Load Balancing che forniscono conteggi di vari codici di risposta HTTP.	19 ottobre 2011
Nuova caratteristica	Puoi ora accedere ai parametri da Amazon EMR.	30 giugno 2011

Modifica	Descrizione	Data di rilascio
Nuova caratteristica	Puoi ora accedere a parametri da Amazon Simple Notification Service e Amazon Simple Queue Service.	14 luglio 2011
Nuova caratteristica	Aggiunte informazioni sull'utilizzo dell'API `PutMetricData` per pubblicare parametri personalizzati. Per ulteriori informazioni, consulta Pubblicare i parametri personalizzati di .	10 maggio 2011
Conservazione dei parametri aggiornata	Amazon CloudWatch ora conserva la cronologia di un allarme per due settimane anziché sei settimane. Con questa modifica, il periodo di conservazione degli allarmi corrisponde al periodo di conservazione di dati dei parametri.	7 Aprile 2011
Nuova caratteristica	Aggiunta della possibilità di inviare notifiche Amazon Simple Notification Service o Auto Scaling al superamento di una soglia di un parametro. Per ulteriori informazioni, consulta Allarmi.	2 dicembre 2010
Nuova caratteristica	Una serie di CloudWatch azioni ora include i NextToken parametri MaxRecords and, che consentono di controllare le pagine di risultati da visualizzare.	2 dicembre 2010
Nuova caratteristica	Questo servizio ora si integra con AWS Identity and Access Management (IAM).	2 dicembre 2010

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.