



Un approccio graduale per l'ingegneria delle prestazioni nel Cloud AWS

# AWS Guida prescrittiva



---

# AWS Guida prescrittiva: Un approccio graduale per l'ingegneria delle prestazioni nel Cloud AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

---

# Table of Contents

Introduzione .....	1
Che cos'è l'ingegneria delle prestazioni? .....	1
Perché utilizzare l'ingegneria delle prestazioni? .....	1
I pilastri dell'ingegneria delle prestazioni .....	3
Generazione di dati di test .....	4
Strumenti per la generazione di dati di test .....	6
Verifica l'osservabilità .....	6
Registrazione dei log .....	8
Monitoraggio .....	12
Tracciamento .....	16
Automazione dei test .....	19
Strumenti di automazione dei test .....	20
Reportistica dei test .....	21
Registrazione standardizzata .....	22
Esempio di pilastri prestazionali .....	23
Risorse .....	25
Collaboratori .....	27
Cronologia dei documenti .....	28
Glossario .....	29
# .....	29
A .....	30
B .....	33
C .....	35
D .....	38
E .....	42
F .....	45
G .....	47
H .....	48
I .....	49
L .....	52
M .....	53
O .....	58
P .....	60
Q .....	63

---

R .....	64
S .....	67
T .....	71
U .....	73
V .....	73
W .....	74
Z .....	75
.....	lxxvi

# Un approccio graduale per l'ingegneria delle prestazioni nel Cloud AWS

Amazon Web Services ([collaboratori](#))

Aprile 2024 (cronologia dei [documenti](#))

Questa guida descrive le migliori pratiche per pianificare, creare e abilitare l'ingegneria delle prestazioni per i carichi di lavoro delle applicazioni in esecuzione su Amazon Web Services (AWS). Definisce quattro pilastri per l'ingegneria delle prestazioni e suggerisce diversi approcci per soddisfare i requisiti prestazionali delle applicazioni. Per ogni pilastro, questa guida elenca strumenti e soluzioni per configurare i test delle prestazioni e l'ambiente di test.

## Che cos'è l'ingegneria delle prestazioni?

L'ingegneria delle prestazioni comprende le tecniche applicate durante il ciclo di vita di sviluppo di un sistema per garantire il rispetto dei requisiti prestazionali non funzionali (come velocità effettiva, latenza o utilizzo della memoria).

Prima di iniziare il test delle prestazioni, è necessario configurare l'ambiente delle prestazioni. Un ambiente prestazionale tipico si basa sui seguenti pilastri:

- Generazione di dati di test
- Osservabilità del test
- Automazione dei test
- Reportistica dei test

## Perché utilizzare l'ingegneria delle prestazioni?

L'ingegneria delle prestazioni è il processo di ottimizzazione continua delle prestazioni delle applicazioni sin dall'inizio della fase di progettazione. Apporta grande valore all'azienda evitando la rilavorazione e il refactoring del codice in una fase successiva del ciclo di sviluppo. L'avvio dell'ingegneria delle prestazioni nella fase di progettazione porta a un'applicazione che offre prestazioni migliori perché le prestazioni possono essere prese in considerazione nella progettazione.

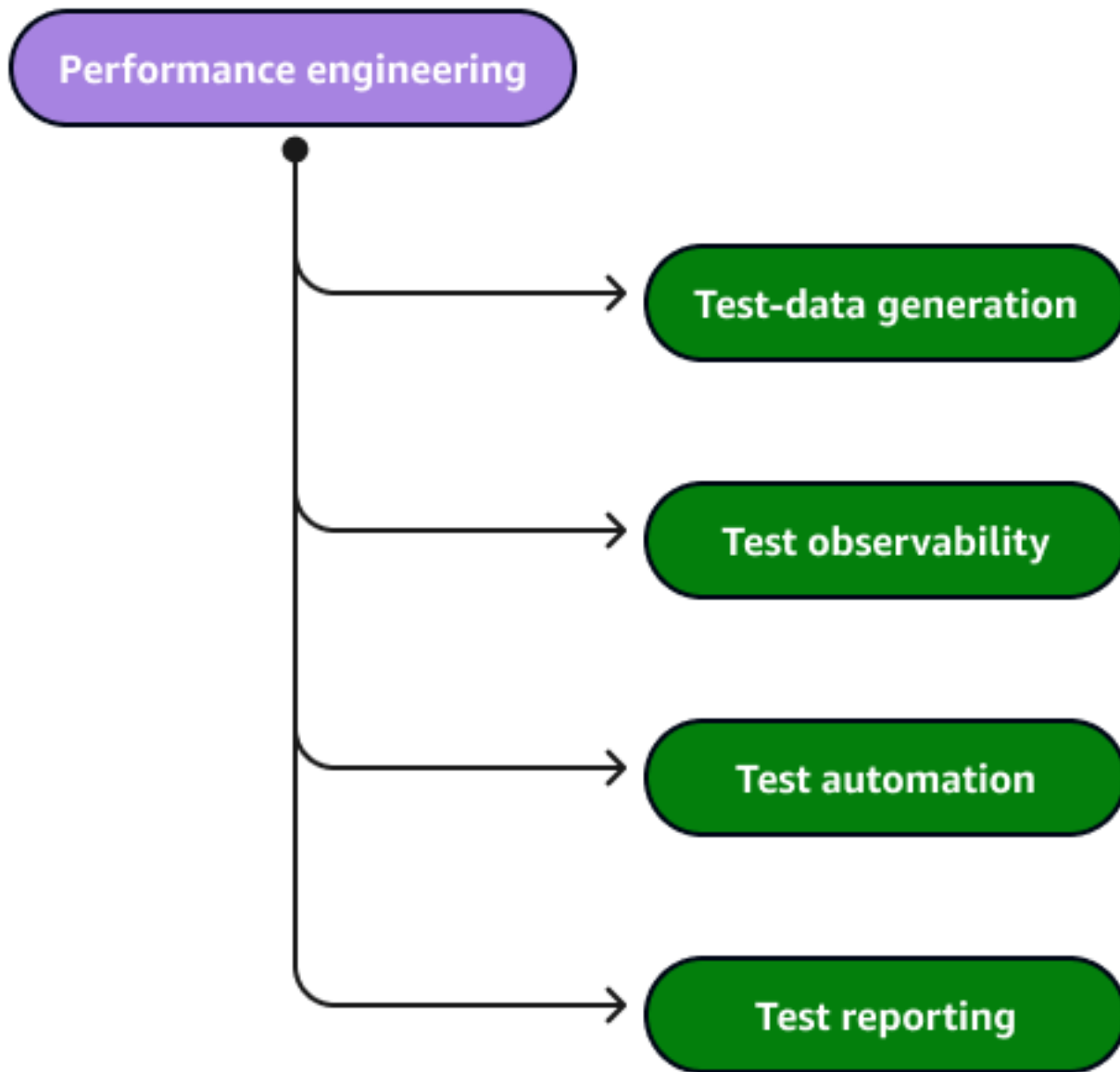
---

L'ingegneria delle prestazioni richiede la partecipazione attiva di architetti di sistema DevOps, sviluppatori e Quality Assurance.

# I pilastri dell'ingegneria delle prestazioni

Per favorire una mentalità ingegneristica delle prestazioni, è importante creare una solida base durante la configurazione dell'ingegneria delle prestazioni per l'applicazione. L'ingegneria delle prestazioni richiede la creazione di quattro pilastri principali:

- Generazione di dati di test: i tecnici delle prestazioni configurano strumenti per generare i dati di test.
- Osservabilità dei test: i tecnici delle prestazioni configurano l'ambiente di osservabilità per garantire che l'esecuzione delle prestazioni possa essere registrata e tracciata e che le risorse che gestiscono i carichi siano monitorate.
- Automazione dei test : [gli ingegneri delle prestazioni sviluppano test automatici che simulano il traffico degli utenti e il carico del sistema utilizzando strumenti come Apache o ghz. JMeter](#)
- Rapporti sui test: vengono raccolti dati sulla configurazione di ogni test eseguito insieme ai risultati delle prestazioni. I dati consentono di correlare le modifiche alla configurazione alle prestazioni e forniscono informazioni preziose.



L'integrazione di questi pilastri incoraggerà la mentalità prestazionale a partire dalle fasi iniziali della progettazione. Ciò contribuirà a evitare modifiche all'applicazione o all'ambiente nelle fasi successive di sviluppo e test.

## Generazione di dati di test

La generazione di dati di test implica la generazione e il mantenimento di una grande quantità di dati per l'esecuzione del test case delle prestazioni. Questi dati generati fungono da input per i test case in modo che l'applicazione possa essere testata su un set diversificato di dati.

Spesso, la generazione di dati di test è un processo complesso. Tuttavia, l'utilizzo di set di dati creati in modo inadeguato può portare a un comportamento imprevedibile delle applicazioni nell'ambiente di produzione. La generazione di dati di test per i test delle prestazioni è diversa dagli approcci tradizionali alla generazione di dati di test. Richiede scenari reali e la maggior parte dei clienti desidera testare i propri carichi di lavoro con dati simili ai dati di produzione effettivi. Inoltre, i dati di test generati di solito devono essere ripristinati o aggiornati allo stato originale dopo ogni esecuzione del test, il che aumenta il tempo e lo sforzo.

La generazione dei dati di test include le seguenti considerazioni principali:

- **Precisione:** l'accuratezza dei dati è importante in tutti gli aspetti dei test. Dati imprecisi generano risultati imprecisi. Ad esempio, quando viene generata una transazione con carta di credito, non dovrebbe riguardare una data futura.
- **Validità:** i dati devono essere validi per il caso d'uso. Ad esempio, durante il test delle transazioni con carta di credito, non è consigliabile generare 10.000 transazioni per utente al giorno, poiché ciò si discosta notevolmente dallo scenario di utilizzo valido.
- **Automazione:** l'automazione della generazione dei dati di test può portare vantaggi in termini di tempo e impegno. Porta anche a un'efficace automazione dei test. La generazione manuale dei dati di test può avere conseguenze in termini di qualità e tempo richiesti.

Esistono diversi meccanismi che è possibile adottare in base ai casi d'uso, come segue:

- **Basato su API:** in questo caso, lo sviluppatore fornisce un'API per la generazione di dati di test che il tester può utilizzare per generare dati. Utilizzando strumenti di test come [JMeter](#), ad esempio, i tester possono scalare la generazione di dati utilizzando un'API aziendale. Ad esempio, se disponi di un'API per aggiungere un utente, puoi utilizzare la stessa API per creare centinaia di utenti con profili diversi. Allo stesso modo, puoi eliminare gli utenti chiamando l'operazione di eliminazione dell'API. Per applicazioni con flussi di lavoro complessi, lo sviluppatore può fornire un'API composita in grado di generare set di dati su diversi componenti. Utilizzando questo approccio, i tester possono scrivere l'automazione per generare ed eliminare i set di dati in base alle loro esigenze.

Tuttavia, se il sistema è complesso o il tempo di risposta dell'API per chiamata è elevato, la configurazione e la rimozione dei dati potrebbero richiedere molto tempo.

- **Basato su istruzioni SQL:** un approccio alternativo consiste nell'utilizzare istruzioni SQL di backend per generare un grande volume di dati. Lo sviluppatore può fornire istruzioni SQL basate su modelli per la generazione di dati di test. I tester possono utilizzare le istruzioni per compilare i dati oppure possono creare script wrapper su queste istruzioni per automatizzare

la generazione dei dati di test. Utilizzando questo approccio, i tester possono compilare e rimuovere i dati molto rapidamente se è necessario ripristinarli dopo il completamento del test. Tuttavia, questo approccio richiede l'accesso diretto al database dell'applicazione, cosa che potrebbe non essere possibile in un tipico ambiente protetto. Inoltre, le interrogazioni non valide potrebbero comportare una compilazione errata dei dati, che può produrre risultati distorti. Gli sviluppatori devono inoltre aggiornare continuamente le istruzioni SQL nel codice dell'applicazione per riflettere le modifiche apportate all'applicazione nel tempo.

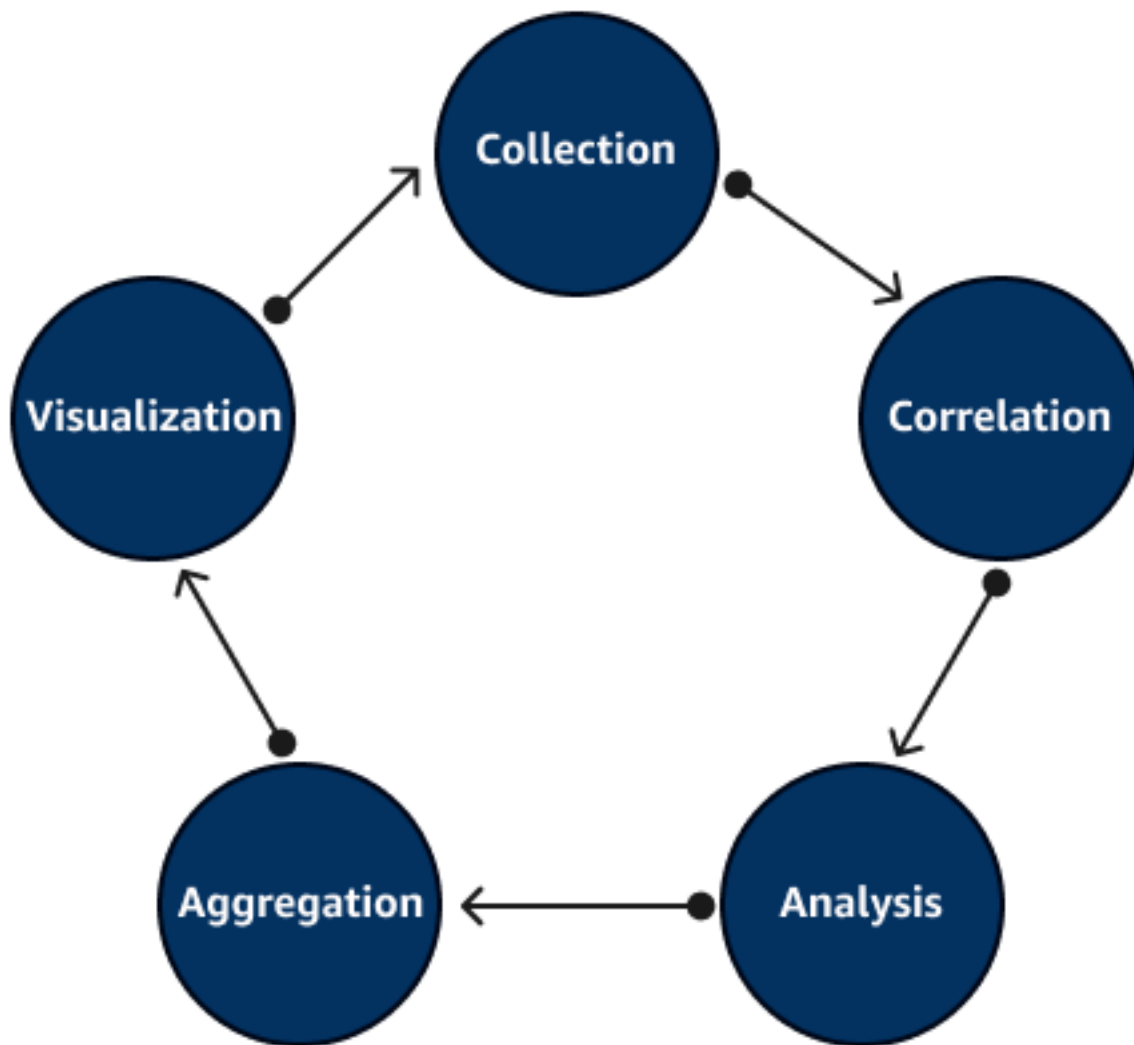
## Strumenti per la generazione di dati di test

AWS fornisce strumenti nativi personalizzati che puoi utilizzare per la generazione di dati di test:

- **Amazon Kinesis Data Generator:** Amazon Kinesis Data Generator (KDG) semplifica il compito di generare dati e inviarli ad Amazon Kinesis. Lo strumento fornisce un'interfaccia utente intuitiva che viene eseguita direttamente nel browser. Per ulteriori informazioni e un'implementazione di riferimento, consulta il post del blog [Test Your Streaming Data Solution with the New Amazon Kinesis Data Generator](#).
- **AWS Glue Test Data Generator:** il AWS Glue Test Data Generator fornisce un framework configurabile per la generazione di dati di test utilizzando processi serverless. AWS Glue PySpark La descrizione dei dati di test richiesta è completamente configurabile tramite un file di configurazione YAML. [Per ulteriori informazioni e un'implementazione di riferimento, consulta il repository Test Data Generator.AWS Glue](#) GitHub

## Verifica l'osservabilità

L'osservabilità dei test supporta la raccolta, la correlazione, l'aggregazione e l'analisi della telemetria nella rete, nell'infrastruttura e nelle applicazioni durante i test delle prestazioni. Ottieni informazioni complete sul comportamento, le prestazioni e lo stato del tuo sistema. Queste informazioni ti aiutano a rilevare, indagare e risolvere i problemi più rapidamente. Aggiungendo l'intelligenza artificiale e l'apprendimento automatico, puoi reagire in modo proattivo, prevedere e prevenire i problemi.



[L'osservabilità si basa sulla registrazione, il monitoraggio e il tracciamento.](#) La responsabilità di implementare con successo queste attività spetta ai team delle applicazioni e dell'infrastruttura.

All'inizio della fase di progettazione, i team applicativi devono comprendere lo stato attuale del loro stack di osservabilità, compresi la registrazione, il monitoraggio e il tracciamento. Possono quindi scegliere gli strumenti che si integrano più agevolmente nello stack di osservabilità.

Allo stesso modo, il team dell'infrastruttura è responsabile della gestione e della scalabilità dell'infrastruttura di osservabilità.

Considerate i seguenti aspetti per quanto riguarda l'osservabilità dei test:

- Disponibilità di registri e tracce delle applicazioni
- Correlazione di log e tracce

- Disponibilità di nodi, contenitori e metriche delle applicazioni
- Automazione per configurare e aggiornare l'infrastruttura di osservabilità su richiesta
- Capacità di visualizzare la telemetria
- Scalabilità dell'infrastruttura di osservabilità

## Registrazione dei log

La registrazione è il processo di conservazione dei dati sugli eventi che si verificano in un sistema. Il registro può includere problemi, errori o informazioni sull'operazione corrente. I log possono essere classificati in diversi tipi, come i seguenti:

- Registro degli eventi
- Registro del server
- Registro di sistema
- Registri di autorizzazione e accesso
- Audit logs

Uno sviluppatore può cercare nei log codici o modelli di errore specifici, filtrarli in base a campi specifici o archivarli in modo sicuro per analisi future. I log aiutano lo sviluppatore a eseguire l'analisi delle cause principali dei problemi di prestazioni e anche a correlare i componenti del sistema.

La creazione di una soluzione di registrazione efficace richiede uno stretto coordinamento tra i team dell'applicazione e dell'infrastruttura. I log delle applicazioni non sono utili a meno che non esista un'infrastruttura di registrazione scalabile che supporti casi d'uso come l'analisi, il filtraggio, il buffering e la correlazione dei log. I casi d'uso più comuni, come la generazione di un ID di correlazione, la registrazione del tempo di esecuzione per i metodi aziendali critici e la definizione dei modelli di registro, possono essere semplificati.

### Team addetto all'applicazione

Uno sviluppatore di applicazioni deve garantire che i log generati seguano le migliori pratiche di registrazione. Le migliori pratiche includono quanto segue:

- Generazione di correlazioni IDs per tenere traccia delle richieste uniche
- Registrazione del tempo impiegato dai metodi aziendali critici

- Registrazione a un livello di registro appropriato
- Condivisione di una libreria di registrazione comune

Quando progetti applicazioni che interagiscono con diversi microservizi, utilizza questi principi di progettazione della registrazione per semplificare il filtraggio e l'estrazione dei log sul backend.

#### Generazione di correlazioni IDs per tenere traccia delle richieste uniche

Quando l'applicazione riceve la richiesta, può verificare se un ID di correlazione è già presente nell'intestazione. Se non è presente un ID, l'applicazione deve generare un ID. Ad esempio, un Application Load Balancer aggiunge un'intestazione chiamata `X-Amzn-Trace-Id`. L'applicazione può utilizzare l'intestazione per correlare la richiesta proveniente dal load balancer all'applicazione. Allo stesso modo, l'applicazione dovrebbe iniettare `traceId` se chiama i microservizi dipendenti in modo che i log generati dai diversi componenti in un flusso di richieste siano correlati.

#### Registrazione del tempo impiegato dai metodi aziendali critici

Quando l'applicazione riceve una richiesta, interagisce con un componente diverso. L'applicazione deve registrare il tempo impiegato per i metodi aziendali critici secondo uno schema definito. Ciò può semplificare l'analisi dei log nel backend. Può anche aiutarti a generare informazioni utili dai log. È possibile utilizzare approcci come la programmazione orientata agli aspetti (AOP) per generare tali registri in modo da separare i problemi di registrazione dalla logica aziendale.

#### Registrazione a un livello di registro appropriato

L'applicazione dovrebbe scrivere registri che contengano una quantità utile di informazioni. Utilizzate i livelli di registro per classificare gli eventi in base alla loro gravità. Ad esempio, utilizzo `WARNING` e `ERROR` livelli per eventi importanti che richiedono un'indagine. Utilizza `INFO` e `DEBUG` per tracciamenti dettagliati ed eventi ad alto volume. Imposta i gestori di log per acquisire solo i livelli necessari per la produzione. Generare troppa registrazione a `INFO` livello di log non è utile e aumenta la pressione sull'infrastruttura di backend. `DEBUG` la registrazione può essere utile, ma deve essere usata con cautela. L'utilizzo `DEBUG` dei log può generare un grande volume di dati, quindi non è consigliato in ambienti di test delle prestazioni.

#### Condivisione di una libreria di registrazione comune

I team applicativi devono utilizzare una libreria di registrazione comune, ad esempio con uno schema di registrazione comune predefinito che gli sviluppatori possono utilizzare come dipendenze nel loro progetto. [AWS SDK per Java](#)

## Team di infrastruttura

DevOps gli ingegneri possono ridurre gli sforzi utilizzando i seguenti principi di progettazione della registrazione durante il filtraggio e l'estrazione dei log sul backend. Il team dell'infrastruttura deve configurare e supportare le seguenti risorse.

### Agente di registro

Un agente di registro (log shipper) è un programma che legge i log da una posizione e li invia a un'altra posizione. I log agent vengono utilizzati per leggere i file di registro memorizzati su un computer e caricare gli eventi di registro nel backend per la centralizzazione.

I log sono dati non strutturati che devono essere strutturati prima di poterli ricavare informazioni significative. Gli agenti di registro utilizzano i parser per leggere le istruzioni di registro ed estrarre i campi pertinenti come timestamp, livello di registro e nome del servizio, e strutturano tali dati in un formato JSON. Disporre di un log agent leggero all'edge è utile perché comporta un minore utilizzo delle risorse. Il log agent può inviare i dati direttamente al backend oppure può utilizzare un log forwarder intermedio che invia i dati al backend. L'uso di un log forwarder scarica il lavoro dagli agenti di registro all'origine.

### Parser di log

Un parser di log converte i log non strutturati in log strutturati. I parser di log agent arricchiscono inoltre i log aggiungendo metadati. L'analisi dei dati può essere eseguita all'origine (fine dell'applicazione) o centralmente. Lo schema per la memorizzazione dei log deve essere estensibile in modo da poter aggiungere nuovi campi. Si consiglia di utilizzare formati di registro standard come JSON. Tuttavia, in alcuni casi, i log devono essere trasformati in formati JSON per una migliore ricerca. La scrittura dell'espressione parser corretta consente una trasformazione efficiente.

### Registra il backend

Un servizio di backend per i registri raccoglie, inserisce e visualizza i dati di registro da varie fonti. L'agente di registro può scrivere direttamente sul backend o utilizzare un log forwarder intermedio. Durante i test delle prestazioni, assicurati di archiviare i log in modo che possano essere cercati in un secondo momento. Archivia i log nel backend separatamente per ogni applicazione. Ad esempio, utilizzate un indice dedicato per un'applicazione e utilizzate lo schema di indice per cercare i log distribuiti tra diverse applicazioni correlate. Consigliamo di salvare almeno 7 giorni di dati per la ricerca nei log. Tuttavia, l'archiviazione dei dati per un periodo più lungo può comportare costi di archiviazione non necessari. Poiché durante il test delle prestazioni viene generato un grande volume

di log, è importante che l'infrastruttura di registrazione ridimensioni e dimensioni correttamente il backend di registrazione.

## Visualizzazione dei log

Per ottenere informazioni significative e utilizzabili dai log delle applicazioni, utilizzate strumenti di visualizzazione dedicati per elaborare e trasformare i dati di registro non elaborati in rappresentazioni grafiche. Visualizzazioni come diagrammi, grafici e dashboard possono aiutare a scoprire tendenze, modelli e anomalie che potrebbero non essere immediatamente evidenti quando si esaminano i log non elaborati.

I vantaggi principali dell'utilizzo degli strumenti di visualizzazione includono la capacità di correlare i dati su più sistemi e applicazioni per identificare dipendenze e colli di bottiglia. I dashboard interattivi supportano l'analisi approfondita dei dati a diversi livelli di granularità per risolvere problemi o individuare tendenze di utilizzo. Le piattaforme specializzate di visualizzazione dei dati forniscono funzionalità come analisi, avvisi e condivisione dei dati che possono migliorare il monitoraggio e l'analisi.

Sfruttando la potenza della visualizzazione dei dati nei registri delle applicazioni, i team di sviluppo e operativi possono ottenere visibilità sulle prestazioni del sistema e delle applicazioni. Le informazioni ottenute possono essere utilizzate per una varietà di scopi, tra cui l'ottimizzazione dell'efficienza, il miglioramento dell'esperienza utente, il miglioramento della sicurezza e la pianificazione della capacità. Il risultato finale sono dashboard personalizzati per le varie parti interessate, che forniscono at-a-glance visualizzazioni che riassumono i dati di registro in informazioni utili e approfondite.

## Automatizzazione dell'infrastruttura di registrazione

Poiché applicazioni diverse hanno requisiti diversi, è importante automatizzare l'installazione e le operazioni dell'infrastruttura di registrazione. Utilizza gli strumenti Infrastructure as Code (IaC) per fornire il backend dell'infrastruttura di registrazione. È quindi possibile effettuare il provisioning dell'infrastruttura di registrazione come servizio condiviso o come implementazione indipendente su misura per una particolare applicazione.

Consigliamo agli sviluppatori di utilizzare pipeline di distribuzione continua (CD) per automatizzare quanto segue:

- Implementa l'infrastruttura di registrazione su richiesta e smontala quando non è necessaria.
- Implementa agenti di registro su destinazioni diverse.
- Implementa configurazioni di log parser e forwarder.

- Implementa i dashboard delle applicazioni.

## Strumenti di registrazione

AWS fornisce servizi nativi di registrazione, allarme e dashboard. Le seguenti sono le risorse più diffuse per Servizi AWS la registrazione:

- Amazon OpenSearch Service aiuta le organizzazioni a raccogliere, inserire e visualizzare i dati di log da varie fonti. Per ulteriori informazioni, consulta [Registrazione centralizzata](#) con OpenSearch
- [Amazon CloudWatch agent](#) e [AWS for Fluent Bit](#) sono i log agent più diffusi su AWS. Per informazioni sull'utilizzo dell' CloudWatch agente con [Amazon CloudWatch Logs Insights](#), consulta il post del blog [Simplifying Apache server logs with Amazon CloudWatch Logs Insights](#). AWS Per l'implementazione di riferimento di Fluent Bit, consulta il post del blog [Centralized Container Logging with Fluent Bit](#).

## Monitoraggio

Il monitoraggio è il processo di raccolta di diverse metriche, come CPU e memoria, e l'archiviazione in un database di serie temporali come Amazon Managed Service for Prometheus. Il sistema di monitoraggio può essere basato su push o pull. Nei sistemi basati su push, la fonte invia periodicamente le metriche al database delle serie temporali. Nei sistemi basati su pull, lo scraper raccoglie le metriche da varie fonti e le archivia nel database delle serie temporali. Gli sviluppatori possono analizzare le metriche, filtrarle e tracciarle nel tempo per visualizzare le prestazioni. L'implementazione corretta del monitoraggio può essere suddivisa in due grandi aree: applicazione e infrastruttura.

Per gli sviluppatori di applicazioni, le seguenti metriche sono fondamentali:

- Latenza: il tempo impiegato per ricevere una risposta
- Throughput delle richieste: il numero totale di richieste gestite al secondo
- Tasso di errore delle richieste: il numero totale di errori

Acquisisci l'utilizzo delle risorse, la saturazione e il conteggio degli errori per ogni risorsa (ad esempio il contenitore dell'applicazione, il database) coinvolta nella transazione commerciale. Ad esempio, quando si monitora l'utilizzo della CPU, è possibile tenere traccia dell'utilizzo medio della CPU, del carico medio e del carico di picco durante l'esecuzione del test delle prestazioni. Quando una

risorsa raggiunge la saturazione durante uno stress test, ma potrebbe non raggiungere la saturazione durante un'esecuzione delle prestazioni per un periodo di tempo più breve.

## Metriche

Le applicazioni possono utilizzare diversi attuatori, come gli attuatori Spring Boot, per monitorare le proprie applicazioni. Queste librerie di livello di produzione generalmente espongono un endpoint REST per il monitoraggio delle informazioni sulle applicazioni in esecuzione. Le librerie possono monitorare l'infrastruttura sottostante, le piattaforme applicative e altre risorse. Se una qualsiasi delle metriche predefinite non soddisfa i requisiti, lo sviluppatore deve implementare metriche personalizzate. Le metriche personalizzate possono aiutare a tenere traccia degli indicatori chiave di performance aziendali (KPIs) che non possono essere tracciati attraverso i dati delle implementazioni predefinite. Ad esempio, potresti voler tenere traccia di un'operazione aziendale come la latenza di integrazione delle API di terze parti o il numero totale di transazioni completate.

## Cardinalità

La cardinalità si riferisce al numero di serie temporali uniche di una metrica. Le metriche sono etichettate per fornire informazioni aggiuntive. Ad esempio, un'applicazione basata su REST che tiene traccia del numero di richieste per una particolare API indica una cardinalità pari a 1. Se aggiungi un'etichetta utente per identificare il numero di richieste per utente, la cardinalità aumenta proporzionalmente al numero di utenti. Aggiungendo etichette che creano cardinalità, puoi suddividere le metriche in base a vari gruppi. È importante utilizzare le etichette giuste per il caso d'uso corretto, perché la cardinalità aumenta il numero di serie di metriche nel database di monitoraggio delle serie temporali di backend.

## Risoluzione

In una configurazione di monitoraggio tipica, l'applicazione di monitoraggio è configurata per acquisire periodicamente le metriche dall'applicazione. La periodicità dello scraping definisce la granularità dei dati di monitoraggio. Le metriche raccolte a intervalli più brevi tendono a fornire una visione più accurata delle prestazioni perché sono disponibili più punti dati. Tuttavia, il carico sul database delle serie temporali aumenta man mano che vengono archiviate più voci. In genere una granularità di 60 secondi è una risoluzione standard e 1 secondo è una risoluzione elevata.

## DevOps squadra

Gli sviluppatori di applicazioni spesso chiedono agli DevOps ingegneri di configurare un ambiente di monitoraggio per visualizzare le metriche dell'infrastruttura e delle applicazioni. L' DevOps ingegnere

deve configurare un ambiente scalabile e che supporti gli strumenti di visualizzazione dei dati utilizzati dallo sviluppatore dell'applicazione. Ciò comporta l'acquisizione dei dati di monitoraggio da diverse fonti e l'invio dei dati a un database centrale di serie temporali come [Amazon Managed Service for Prometheus](#).

## Backend di monitoraggio

Un servizio di backend di monitoraggio supporta la raccolta, l'archiviazione, l'interrogazione e la visualizzazione dei dati delle metriche. In genere è un database di serie temporali come Amazon Managed Service for Prometheus o InfluxDB. InfluxData Utilizzando un meccanismo di rilevamento dei servizi, il raccoglitore di monitoraggio può raccogliere metriche da diverse fonti e archivarle. Durante i test delle prestazioni, è importante archiviare i dati delle metriche in modo che possano essere ricercati in un secondo momento. Ti consigliamo di salvare almeno 15 giorni di dati per le metriche. Tuttavia, l'archiviazione delle metriche per un periodo più lungo non aggiunge vantaggi significativi e comporta costi di archiviazione inutili. Poiché il test delle prestazioni può generare un grande volume di metriche, è importante che l'infrastruttura delle metriche sia scalabile e fornisca al contempo prestazioni di query rapide. Il servizio di backend di monitoraggio fornisce un linguaggio di interrogazione che può essere utilizzato per visualizzare i dati delle metriche.

## Visualizzazione

Fornisci strumenti di visualizzazione in grado di visualizzare i dati dell'applicazione per fornire informazioni significative. L' DevOps ingegnere e lo sviluppatore dell'applicazione devono imparare il linguaggio di interrogazione per il backend di monitoraggio e lavorare a stretto contatto per generare un modello di dashboard che possa essere riutilizzato. Nelle dashboard, includi la latenza e gli errori, visualizzando allo stesso tempo l'utilizzo e la saturazione delle risorse nell'infrastruttura e nelle risorse applicative.

## Automatizzazione dell'infrastruttura di monitoraggio

Analogamente alla registrazione, è importante automatizzare l'installazione e il funzionamento dell'infrastruttura di monitoraggio in modo da soddisfare i diversi requisiti delle diverse applicazioni. Utilizza gli strumenti IaC per fornire il backend dell'infrastruttura di monitoraggio. È quindi possibile effettuare il provisioning dell'infrastruttura di monitoraggio come servizio condiviso o come implementazione indipendente su misura per una particolare applicazione.

Utilizza le pipeline CD per automatizzare quanto segue:

- Implementa l'infrastruttura di monitoraggio su richiesta e smontala quando non è necessaria.

- Aggiorna la configurazione di monitoraggio per filtrare o aggregare le metriche.
- Implementa i dashboard delle applicazioni.

## Strumenti di monitoraggio

Amazon Managed Service for Prometheus è [un](#) servizio di monitoraggio compatibile con Prometheus per l'infrastruttura dei container e i parametri applicativi per i container che puoi utilizzare per monitorare in modo sicuro gli ambienti di container su larga scala. Per ulteriori informazioni, consulta il post del blog [Getting Started with Amazon Managed Service for Prometheus](#).

Amazon CloudWatch offre il monitoraggio completo su. AWS CloudWatch supporta soluzioni AWS native e open source in modo che tu possa capire cosa sta succedendo nel tuo stack tecnologico in qualsiasi momento.

AWS Gli strumenti nativi includono quanto segue:

- [Pannelli di CloudWatch controllo Amazon](#)
- [CloudWatch](#) Container Insights
- [CloudWatch metriche](#)
- [CloudWatch allarmi](#)

Amazon CloudWatch offre funzionalità appositamente progettate per casi d'uso specifici come il monitoraggio dei container tramite CloudWatch Container Insights. Queste funzionalità sono integrate in CloudWatch modo da poter configurare log, raccolta di metriche e monitoraggio.

Per le applicazioni e i microservizi containerizzati, utilizza Container Insights per raccogliere, aggregare e riepilogare metriche e log. Container Insights è disponibile per le piattaforme Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) e Kubernetes su Amazon Elastic Compute Cloud (Amazon EC2). [Container Insights raccoglie i dati sotto forma di eventi di registro delle prestazioni nel formato metrico incorporato](#). Queste voci degli eventi del registro delle prestazioni utilizzano uno schema JSON strutturato che supporta l'inserimento e l'archiviazione di dati ad alta cardinalità su larga scala.

Per informazioni sull'implementazione di Container Insights con Amazon EKS, consulta il post del blog [Introduzione ad Amazon CloudWatch Container Insights per Amazon EKS Fargate using AWS Distro](#) for. OpenTelemetry

## Tracciamento

Il tracciamento implica l'uso specializzato della registrazione delle informazioni sui processi di un programma. Le informazioni ricavate dai log possono aiutare gli ingegneri a eseguire il debug delle singole transazioni e a identificare i punti deboli. Il tracciamento può essere abilitato automaticamente o utilizzando la strumentazione manuale.

Poiché un'applicazione si integra con diversi servizi, è importante identificare le prestazioni dell'applicazione e dei servizi sottostanti. Il tracciamento funziona con tracce e intervalli. Una traccia è il processo di richiesta completo e ogni traccia è composta da intervalli. Un intervallo è un intervallo di tempo contrassegnato e rappresenta l'attività all'interno dei singoli componenti o servizi di un sistema. Le tracce forniscono un quadro generale di ciò che accade quando viene effettuata una richiesta a un'applicazione.

### Team di candidatura

Gli sviluppatori di applicazioni strumentalizzano le proprie applicazioni inviando dati di traccia per le richieste in entrata e in uscita e altri eventi all'interno dell'applicazione, insieme ai metadati relativi a ciascuna richiesta. Per generare tracce, un'applicazione deve essere dotata della strumentazione necessaria per generare tracce. La strumentazione può essere automatica o manuale.

### Strumentazione automatica

È possibile raccogliere dati di telemetria da un'applicazione utilizzando la [strumentazione automatica](#) senza dover modificare il codice sorgente. Gli agenti di strumentazione automatica possono generare tracce applicative di un'applicazione o di un servizio. In genere, si utilizzano le modifiche alla configurazione per aggiungere l'agente o un altro meccanismo.

La strumentazione della libreria implica apportare modifiche minime al codice dell'applicazione per aggiungere strumentazione precostruita. La strumentazione si rivolge a librerie o framework specifici, come AWS SDK, client HTTP Apache o client SQL.

### Strumentazione manuale

In questo approccio, gli sviluppatori di applicazioni aggiungono codice di strumentazione all'applicazione in ogni posizione in cui desiderano raccogliere informazioni di traccia. Ad esempio, utilizzate la programmazione orientata agli aspetti (AOP) per raccogliere dati di tracciamento. AWS X-Ray Gli sviluppatori possono SDKs utilizzarla per strumentare le proprie applicazioni.

## Campionamento

I dati di tracciamento vengono spesso generati in grandi volumi. È importante disporre di un meccanismo per determinare se i dati di traccia devono essere esportati o meno. Il campionamento è il processo per determinare quali dati devono essere esportati. Questo viene generalmente fatto per ridurre i costi. Personalizzando le regole di campionamento, è possibile controllare la quantità di dati registrati. È inoltre possibile modificare il comportamento di campionamento senza modificare e ridistribuire il codice. È importante controllare la frequenza di campionamento per generare la giusta quantità di tracce.

Gli sviluppatori di applicazioni possono annotare le tracce aggiungendo metadati come coppie chiave-valore. Le annotazioni arricchiscono le tracce e aiutano a perfezionare il filtraggio nel backend.

## DevOps squadra

DevOps agli ingegneri viene spesso chiesto di configurare un ambiente di tracciamento per consentire allo sviluppatore di applicazioni di visualizzare le tracce per l'infrastruttura e le applicazioni. La configurazione dell'ambiente di tracciamento prevede la raccolta di dati di traccia da diverse fonti e l'invio a un archivio centrale per la visualizzazione.

## Backend di tracciamento

Un backend di tracciamento è un servizio come AWS X-Ray quello che raccoglie dati sulle richieste servite dall'applicazione. Fornisce strumenti che è possibile utilizzare per visualizzare, filtrare e ottenere informazioni dettagliate su tali dati per identificare problemi e opportunità di ottimizzazione. Per ogni richiesta tracciata all'applicazione, è possibile visualizzare informazioni dettagliate sulla richiesta e sulla risposta e sulle altre chiamate effettuate dall'applicazione a AWS risorse a valle, microservizi, database e Web. APIs

## Tracciamento automatico

Poiché applicazioni diverse hanno requisiti di tracciamento diversi, è importante automatizzare la configurazione e il funzionamento dell'infrastruttura di tracciamento. Utilizza gli strumenti IaC per fornire il backend dell'infrastruttura di tracciamento.

Utilizza le pipeline CD per automatizzare quanto segue:

- Implementa l'infrastruttura di tracciamento su richiesta e smontala quando non è necessaria.
- Implementa la configurazione di tracciamento tra le applicazioni.

## Strumenti di tracciamento

AWS fornisce i seguenti servizi per il tracciamento e la visualizzazione associata:

- AWS X-Ray riceve tracce dall'applicazione, oltre alle tracce dai AWS servizi utilizzati dall'applicazione che sono già integrati con X-Ray. Esistono diversi SDKs agenti e strumenti che possono essere utilizzati per strumentare l'applicazione per il tracciamento a raggi X.. Per ulteriori informazioni, consulta la [documentazione relativa ad AWS X-Ray](#).

Gli sviluppatori possono anche utilizzare AWS X-Ray SDKs per inviare tracce a X-Ray. AWS X-Ray SDKs prevedeGo, Node.js JavaPython, .NET eRuby. Ogni X-Ray SDK offre quanto segue:

- Moduli di intercettazione da aggiungere al tuo codice per il tracciamento delle richieste HTTP in ingresso
- Da gestori client a client AWS SDK di strumentazione utilizzati dall'applicazione per chiamare altri servizi AWS
- Un client HTTP per effettuare chiamate strumentali ad altri servizi Web HTTP interni ed esterni

X-Ray supporta SDKs anche le chiamate strumentali ai database SQL, la strumentazione client AWS SDK automatica e altre funzionalità. Invece di inviare i dati di traccia direttamente a X-Ray, l'SDK invia i documenti del segmento JSON a un processo demone che ascolta il traffico UDP. Il [demone X-Ray memorizza i segmenti in una coda e li carica su X-Ray in batch](#). [Per ulteriori informazioni sulla strumentazione dell'applicazione utilizzando un SDK X-Ray, consultate la documentazione X-Ray](#).

- Amazon OpenSearch Service è un servizio AWS gestito per l'esecuzione e il ridimensionamento OpenSearch dei cluster, che può essere utilizzato per archiviare centralmente log, metriche e tracce. Il plug-in Observability offre un'esperienza unificata per la raccolta e il monitoraggio di parametri, registri e tracce provenienti da origini dati comuni. La raccolta e il monitoraggio dei dati in un unico posto offrono l'osservabilità completa dell'intera infrastruttura. end-to-end [Per informazioni sull'implementazione, consulta la documentazione del OpenSearch servizio](#).
- AWS Distro for OpenTelemetry (ADOT) è una AWS distribuzione basata sul progetto Cloud Native Computing Foundation (CNCF). OpenTelemetry [ADOT attualmente include il supporto per la strumentazione automatica per Java e Python](#). [Inoltre, ADOT supporta la strumentazione automatica delle AWS Lambda funzioni e delle relative richieste downstream utilizzando Java Node.js e i Python runtime tramite ADOT Managed Lambda Layers](#). Gli sviluppatori possono utilizzare il collettore ADOT per inviare tracce a diversi backend, tra cui AWS X-Ray Amazon Service. OpenSearch

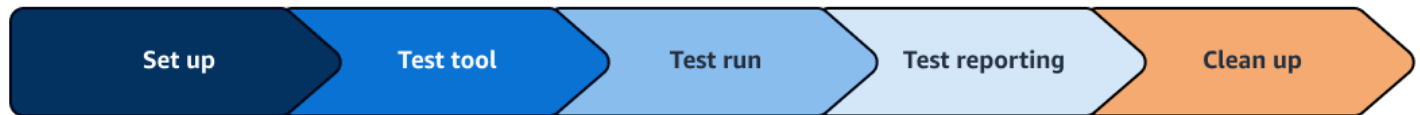
[Per un esempio di riferimento su come strumentare la tua applicazione utilizzando l'SDK ADOT, consulta la documentazione.](#) Per un esempio di riferimento su come utilizzare l'SDK ADOT per inviare dati ad Amazon OpenSearch Service, consulta la documentazione del [OpenSearch servizio](#).

Per un esempio di riferimento su come strumentare la tua applicazione in esecuzione su Amazon EKS, consulta il post del blog [Raccolta di metriche e tracce utilizzando i componenti aggiuntivi di Amazon EKS per AWS Distro for. OpenTelemetry](#)

## Automazione dei test

I test automatizzati con un framework e strumenti specializzati possono ridurre l'intervento umano e massimizzare la qualità. I test automatici delle prestazioni non sono diversi dai test di automazione come i test di unità e i test di integrazione.

Utilizza DevOps le pipeline nelle diverse fasi per i test delle prestazioni.



Le cinque fasi della pipeline di automazione dei test sono:

1. Configurazione: utilizza gli approcci relativi ai dati di test descritti nella sezione [Generazione dei dati di test](#) per questa fase. La generazione di dati di test realistici è fondamentale per ottenere risultati di test validi. È necessario creare con cura dati di test diversi che coprano un'ampia gamma di casi d'uso e corrispondano strettamente ai dati di produzione in tempo reale. Prima di eseguire test delle prestazioni su vasta scala, potrebbe essere necessario eseguire test di prova iniziali per convalidare gli script di test, gli ambienti e gli strumenti di monitoraggio.
2. Strumento di test: per eseguire il test delle prestazioni, seleziona uno strumento di test di carico appropriato, ad esempio o ghz. JMeter Considerate la soluzione migliore per le vostre esigenze aziendali in termini di simulazione di carichi utente reali.
3. Esecuzione dei test: con gli strumenti e gli ambienti di test stabiliti, esegui test end-to-end delle prestazioni su una gamma di carichi utente e durate previsti. Durante tutto il test, monitora attentamente lo stato del sistema sottoposto a test. Si tratta in genere di una fase di lunga durata. Monitora i tassi di errore per l'invalidazione automatica dei test e interrompi il test se ci sono troppi errori.

Lo strumento di test del carico fornisce informazioni sull'utilizzo delle risorse, sui tempi di risposta e sui potenziali colli di bottiglia.

4. **Rapporti sui test:** raccogli i risultati dei test insieme all'applicazione e alla configurazione del test. Automatizza la raccolta della configurazione delle applicazioni, della configurazione dei test e dei risultati, il che aiuta a registrare i dati relativi ai test delle prestazioni e ad archivarli centralmente. La gestione centralizzata dei dati sulle prestazioni aiuta a fornire informazioni approfondite e supporta la definizione programmatica dei criteri di successo per l'azienda.
5. **Pulizia:** dopo aver completato un test delle prestazioni, ripristina l'ambiente e i dati di test per prepararti alle esecuzioni successive. Innanzitutto, ripristina tutte le modifiche apportate ai dati del test durante l'esecuzione. È necessario ripristinare i database e gli altri archivi di dati allo stato originale, ripristinando tutti i record nuovi, aggiornati o eliminati generati durante il test.

È possibile riutilizzare la pipeline per ripetere il test più volte finché i risultati non riflettono le prestazioni desiderate. Puoi anche utilizzare la pipeline per verificare che le modifiche al codice non compromettano le prestazioni. È possibile eseguire test di convalida del codice in orari non lavorativi e utilizzare i dati di test e osservabilità disponibili per la risoluzione dei problemi.

Le migliori pratiche includono quanto segue:

- Registra l'ora di inizio e di fine e genera automaticamente URL per la registrazione. Questo ti aiuta a filtrare i dati di osservabilità nella finestra temporale appropriata. Sistemi di monitoraggio e tracciamento.
- Inserisci gli identificatori dei test nell'intestazione mentre richiami i test. Gli sviluppatori di applicazioni possono arricchire i propri dati di registrazione, monitoraggio e tracciamento utilizzando l'identificatore come filtro nel backend.
- Limita la pipeline a una sola esecuzione alla volta. L'esecuzione di test simultanei genera rumori che possono causare confusione durante la risoluzione dei problemi. È inoltre importante eseguire il test in un ambiente prestazionale dedicato.

## Strumenti di automazione dei test

Gli strumenti di test svolgono un ruolo importante in qualsiasi automazione dei test. Le scelte più comuni per gli strumenti di test open source includono quanto segue:

- [Apache JMeter](#) è il cavallo da corsa più esperto. Nel corso degli anni, Apache JMeter è diventato più affidabile e ha aggiunto funzionalità. Con l'interfaccia grafica, è possibile creare test complessi senza conoscere alcun linguaggio di programmazione. Aziende come Apache BlazeMeter supportano Apache JMeter.
- [K6](#) è uno strumento gratuito che offre supporto, hosting della sorgente di carico e un'interfaccia web integrata per organizzare, eseguire e analizzare i test di carico.
- Il test di carico [Vegeta](#) segue un concetto diverso. Invece di definire la simultaneità o sovraccaricare il sistema, viene definita una determinata frequenza. Lo strumento crea quindi quel carico indipendentemente dai tempi di risposta del sistema.
- [Hey](#) e [ab](#), lo strumento di benchmarking del server HTTP Apache, sono strumenti di base che è possibile utilizzare dalla riga di comando per eseguire il carico specificato su un singolo endpoint. Questo è il modo più veloce per generare carico se si dispone di un server su cui eseguire gli strumenti. Funzionerà anche un laptop locale, anche se potrebbe non essere abbastanza potente da produrre un carico elevato.
- [ghz](#) è un'utilità da riga di comando e un pacchetto [Go](#) per il test di carico e il benchmarking dei servizi [gRPC](#).

AWS fornisce il Distributed Load Testing sulla AWS soluzione. La soluzione crea e simula migliaia di utenti connessi che generano record transazionali a un ritmo costante senza la necessità di fornire server. [Per ulteriori informazioni, consulta la Solutions Library.AWS](#)

È possibile utilizzarla AWS CodePipeline per automatizzare la pipeline di test delle prestazioni. [Per ulteriori informazioni sull'automazione dei test delle API mediante l'utilizzo CodePipeline, consulta il AWS DevOps blog e la documentazione.AWS](#)

## Reportistica dei test

I report sui test si riferiscono alla raccolta, all'analisi e alla presentazione di dati relativi alle prestazioni di sistemi, applicazioni, servizi o processi. Implica la misurazione di varie metriche e indicatori per valutare l'efficienza, la reattività, l'affidabilità e l'efficacia complessiva di un particolare sistema o componente.

Il reporting basato sui test delle prestazioni implica la scelta di metriche pertinenti in base al contesto e agli obiettivi dell'analisi. Le metriche prestazionali più comuni includono tempi di risposta, velocità effettiva, tassi di errore, utilizzo delle risorse (CPU, memoria, disco) e latenza di rete.

Una volta raccolti, i dati relativi alle prestazioni devono essere archiviati in un archivio centrale. I risultati di questi test potrebbero provenire da ambienti, applicazioni e strumenti di test diversi. Quando sono in esecuzione più carichi di lavoro in ambienti diversi, è difficile raccogliere dati relativi alle prestazioni e correlarli tra questi dati per trarre conclusioni informate. Consigliamo di definire un metodo standard per la raccolta dei dati relativi alle metriche delle prestazioni utilizzando un repository centrale per l'archiviazione e la visualizzazione dei dati.

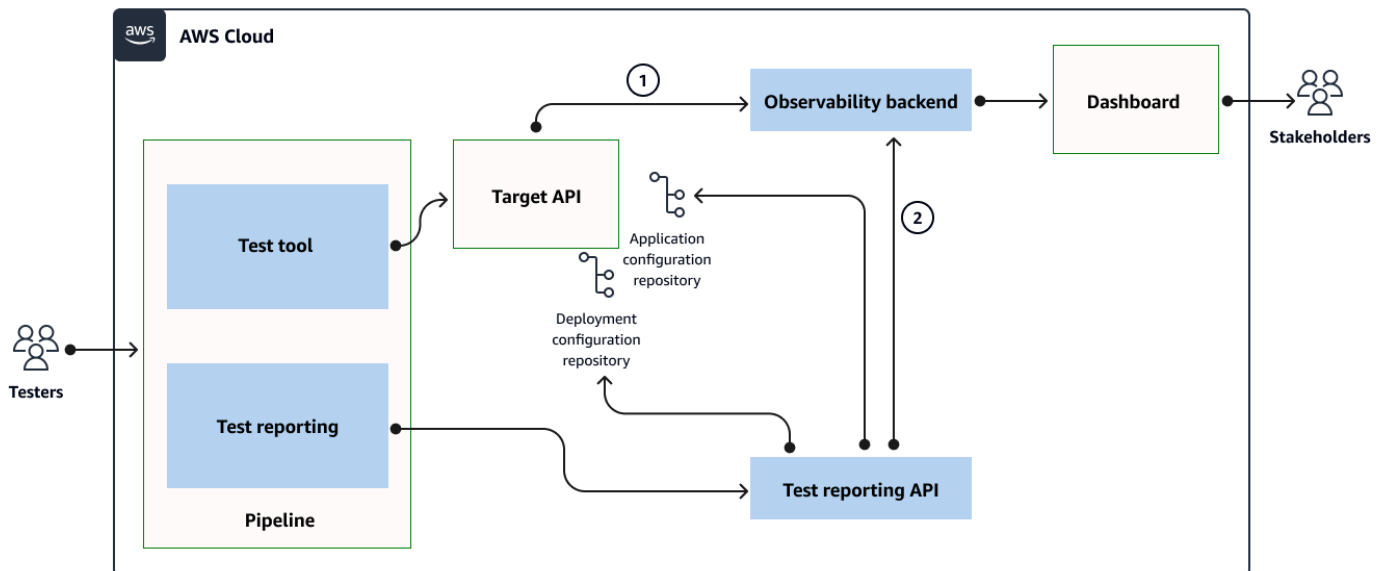
## Registrazione standardizzata

Consigliamo di standardizzare il modo in cui le diverse parti interessate eseguono i test delle prestazioni e scrivono i dati risultanti in un archivio centrale. Ad esempio, ciò potrebbe assumere la forma di un'API che accetta i risultati e li archivia in una soluzione di archiviazione persistente. In situazioni in cui è necessario recuperare i dati da fonti come GitOps Amazon Managed Service for Prometheus, l'API può estrarre tali dettagli direttamente dalle fonti specificate sulla base di file di schema che descrivono come estrarre i campi dalle specifiche di implementazione e dalle specifiche di Kubernetes. [I file di schema possono utilizzare JSONPath espressioni o Prometheus Query Language \(PromQL\)](#). Come accennato in precedenza, le metriche raccolte devono essere pertinenti al contesto e agli obiettivi dell'analisi delle prestazioni.

I dati trasmessi all'API possono includere dettagli e tag relativi all'applicazione e all'ambiente per cui è stato eseguito il test. Questo aiuta a eseguire analisi sui dati dei test delle prestazioni.

# I pilastri dell'ingegneria delle prestazioni in azione

La seguente architettura di riferimento illustra i pilastri dell'ingegneria delle prestazioni per testare un'API specifica.



1. I dati di registrazione, monitoraggio e tracciamento vengono inviati dall'API di destinazione al backend.
2. Quando viene richiamata, l'API di segnalazione dei test invia i risultati e le informazioni di configurazione al backend.

Il componente principale è l'API o l'applicazione di destinazione in fase di test. L'API di destinazione si sincronizza con l'archivio di configurazione dell'applicazione e l'archivio di configurazione della distribuzione in GitOps modo da ottenere le configurazioni più recenti dell'applicazione e dell'infrastruttura. Questa sincronizzazione consente l'esecuzione dei test automatici sullo stato attuale desiderato dell'applicazione e della sua infrastruttura di supporto, come definito nei repository Git.

La pipeline di automazione dei test automatizza la generazione dei dati di test, l'esecuzione del test e la segnalazione dei risultati del test per l'API di destinazione.

L'API di destinazione genera informazioni sulle prestazioni (metriche, log e tracce), utilizzando le [migliori pratiche di osservabilità](#), e trasmette i dati delle metriche al backend di osservabilità.

---

L'API di reporting dei test raccoglie tutti i dati di reporting relativi ai test (configurazione e risultati dei test) e li archivia nel backend di osservabilità.

L'aggregazione di informazioni sulle prestazioni e dati di reporting (configurazione, risultati dei test) consente di interrogare i dati relativi alle prestazioni per l'API di destinazione. Ad esempio, potresti chiedere quanto segue:

- Quali sono le prime dieci transazioni più lente?
- Qual è il P99, P90, il numero medio di ogni test?
- Come si confrontano le configurazioni dei due cicli di test?

La correlazione dei casi di test con risultati, configurazioni e metriche su un periodo di tempo aiuta a identificare la configurazione migliore e i risultati prestazionali.

Utilizzando i risultati di questi test, è possibile prendere decisioni più precise e basate sui dati per l'API e avere fiducia nel portare l'API in produzione.

# Risorse

## Servizi AWS

- [Amazon CloudWatch](#)
- [AWS CodePipeline](#)
- [AWS Distro per OpenTelemetry](#)
- [OpenSearch Servizio Amazon](#)
- [AWS X-Ray](#)

## Implementazioni

- [amazon-kinesis-data-generator](#)
- [AWS Glue Generatore di dati di test](#)
- [Test di carico distribuito su AWS](#)

## Post di blog

- [Registrazione centralizzata dei container con Fluent Bit](#)
- [Testa la tua soluzione di streaming di dati con il nuovo Amazon Kinesis Data Generator](#)
- [Presentazione di Amazon CloudWatch Container Insights per Amazon EKS Fargate con AWS Distro per OpenTelemetry](#)
- [Tracciamento delle applicazioni su Kubernetes con AWS X-Ray](#)
- [Raccolta di metriche e tracce con componenti aggiuntivi Amazon EKS per Distro per AWS OpenTelemetry](#)
- [Guida introduttiva ad Amazon Managed Service for Prometheus](#)

## Workshop

- [Introduzione all' AWS osservabilità](#)

## AWS Guida prescrittiva

- [Applicazioni per i test di carico](#) (guida)

---

## Applicazioni di terze parti

- [Apache JMeter](#)
- [K6](#)
- [Vegeta](#)
- [Ehi](#) e [ab](#)
- [GHZ](#)

# Collaboratori

Hanno collaborato alla stesura del presente documento:

- Varun Sharma, consulente principale senior, AWS
- Akash Kumar, consulente capo senior, AWS
- Archana Bhatnagar, responsabile dello studio, AWS
- Pratik Sharma, Servizi professionali II, AWS

## Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un [feed RSS](#).

Modifica	Descrizione	Data
<a href="#">Pubblicazione iniziale</a>	—	24 aprile 2024

# AWS Glossario delle linee guida prescrittive

I seguenti sono termini di uso comune nelle strategie, nelle guide e nei modelli forniti da AWS Prescriptive Guidance. Per suggerire voci, utilizza il link [Fornisci feedback](#) alla fine del glossario.

## Numeri

### 7 R

Sette strategie di migrazione comuni per trasferire le applicazioni sul cloud. Queste strategie si basano sulle 5 R identificate da Gartner nel 2011 e sono le seguenti:

- **Refactor/re-architect** — Sposta un'applicazione e modificala sfruttando appieno le funzionalità native del cloud per migliorare l'agilità, le prestazioni e la scalabilità. Ciò comporta in genere la portabilità del sistema operativo e del database. Esempio: migra il tuo database Oracle locale all'edizione Amazon PostgreSQL-Compatible Aurora.
- **Ridefinire la piattaforma (lift and reshape)**: trasferisci un'applicazione nel cloud e introduci un certo livello di ottimizzazione per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale ad Amazon Relational Database Service (Amazon RDS) per Oracle in Cloud AWS
- **Riacquistare (drop and shop)**: passa a un prodotto diverso, in genere effettuando la transizione da una licenza tradizionale a un modello SaaS. Esempio: migra il tuo sistema di gestione delle relazioni con i clienti (CRM) su Salesforce.com
- **Eseguire il rehosting (lift and shift)**: trasferisci un'applicazione sul cloud senza apportare modifiche per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale su Oracle su un'istanza EC2 in Cloud AWS
- **Trasferire (eseguire il rehosting a livello hypervisor)**: trasferisci l'infrastruttura sul cloud senza acquistare nuovo hardware, riscrivere le applicazioni o modificare le operazioni esistenti. Esegui la migrazione dei server da una piattaforma locale a un servizio cloud per la stessa piattaforma. Esempio: migra un'applicazione su Microsoft Hyper-V. AWS
- **Riesaminare (mantenere)**: mantieni le applicazioni nell'ambiente di origine. Queste potrebbero includere applicazioni che richiedono una rifattorizzazione significativa che desideri rimandare a un momento successivo e applicazioni legacy che desideri mantenere, perché non vi è alcuna giustificazione aziendale per effettuarne la migrazione.
- **Ritirare**: disattiva o rimuovi le applicazioni che non sono più necessarie nell'ambiente di origine.

# A

## A2A () Agent-to-Agent

Un protocollo statico per la collaborazione tra agenti che supporta la delega delle attività e il trasferimento dello stato.

## ABAC

[Vedi controllo degli accessi basato sugli attributi.](#)

## servizi astratti

Vedi [servizi gestiti](#).

## ACIDO

Vedi [atomicità, consistenza, isolamento, durata](#).

## migrazione attiva-attiva

Un metodo di migrazione del database in cui i database di origine e di destinazione vengono mantenuti sincronizzati (utilizzando uno strumento di replica bidirezionale o operazioni di doppia scrittura) ed entrambi i database gestiscono le transazioni provenienti dalle applicazioni di connessione durante la migrazione. Questo metodo supporta la migrazione in piccoli batch controllati anziché richiedere una conversione una tantum. È più flessibile ma richiede più lavoro rispetto alla migrazione [attiva-passiva](#).

## migrazione attiva-passiva

Un metodo di migrazione del database in cui i database di origine e di destinazione vengono mantenuti sincronizzati, ma solo il database di origine gestisce le transazioni provenienti dalle applicazioni di connessione mentre i dati vengono replicati nel database di destinazione. Il database di destinazione non accetta alcuna transazione durante la migrazione.

## Agente

Un sistema di intelligenza artificiale in grado di ragionare, pianificare e intraprendere azioni in modo autonomo utilizzando strumenti per raggiungere gli obiettivi.

## Agente Ops

Pratiche operative per la creazione, il test, l'implementazione e l'esecuzione di agenti di intelligenza artificiale in produzione su larga scala.

## funzione aggregata

Una funzione SQL che opera su un gruppo di righe e calcola un singolo valore restituito per il gruppo. Esempi di funzioni aggregate includono SUM e MAX.

## Intelligenza artificiale

Vedi [intelligenza artificiale](#).

## AIOps

Guarda le [operazioni di intelligenza artificiale](#).

## anonimizzazione

Il processo di eliminazione permanente delle informazioni personali in un set di dati.

L'anonimizzazione può aiutare a proteggere la privacy personale. I dati anonimi non sono più considerati dati personali.

## anti-modello

Una soluzione utilizzata frequentemente per un problema ricorrente in cui la soluzione è controproducente, inefficace o meno efficace di un'alternativa.

## controllo delle applicazioni

Un approccio alla sicurezza che consente l'uso solo di applicazioni approvate per proteggere un sistema dal malware.

## portfolio di applicazioni

Una raccolta di informazioni dettagliate su ogni applicazione utilizzata da un'organizzazione, compresi i costi di creazione e manutenzione dell'applicazione e il relativo valore aziendale. Queste informazioni sono fondamentali per [il processo di scoperta e analisi del portfolio](#) e aiutano a identificare e ad assegnare la priorità alle applicazioni da migrare, modernizzare e ottimizzare.

## intelligenza artificiale (IA)

Il campo dell'informatica dedicato all'uso delle tecnologie informatiche per svolgere funzioni cognitive tipicamente associate agli esseri umani, come l'apprendimento, la risoluzione di problemi e il riconoscimento di schemi. Per ulteriori informazioni, consulta la sezione [Che cos'è l'intelligenza artificiale?](#)

## operazioni di intelligenza artificiale (AIOps)

Il processo di utilizzo delle tecniche di machine learning per risolvere problemi operativi, ridurre gli incidenti operativi e l'intervento umano e aumentare la qualità del servizio. Per ulteriori

informazioni su come viene utilizzato AIOps nella strategia di migrazione AWS , consulta la [guida all'integrazione delle operazioni](#).

### crittografia asimmetrica

Un algoritmo di crittografia che utilizza una coppia di chiavi, una chiave pubblica per la crittografia e una chiave privata per la decrittografia. Puoi condividere la chiave pubblica perché non viene utilizzata per la decrittografia, ma l'accesso alla chiave privata deve essere altamente limitato.

### atomicità, consistenza, isolamento, durabilità (ACID)

Un insieme di proprietà del software che garantiscono la validità dei dati e l'affidabilità operativa di un database, anche in caso di errori, interruzioni di corrente o altri problemi.

### Controllo degli accessi basato su attributi (ABAC)

La pratica di creare autorizzazioni dettagliate basate su attributi utente, come reparto, ruolo professionale e nome del team. Per ulteriori informazioni, consulta [ABAC for AWS](#) nella documentazione AWS Identity and Access Management (IAM).

### fonte di dati autorevole

Una posizione in cui è archiviata la versione principale dei dati, considerata la fonte di informazioni più affidabile. È possibile copiare i dati dalla fonte di dati autorevole in altre posizioni allo scopo di elaborarli o modificarli, ad esempio anonimizzandoli, oscurandoli o pseudonimizzandoli.

### Zona di disponibilità

Una posizione distinta all'interno di un edificio Regione AWS che è isolata dai guasti in altre zone di disponibilità e offre una connettività di rete economica e a bassa latenza verso altre zone di disponibilità nella stessa regione.

### AWS Cloud Adoption Framework (CAF)AWS

Un framework di linee guida e best practice AWS per aiutare le organizzazioni a sviluppare un piano efficiente ed efficace per passare con successo al cloud. AWS CAF organizza le linee guida in sei aree di interesse chiamate prospettive: business, persone, governance, piattaforma, sicurezza e operazioni. Le prospettive relative ad azienda, persone e governance si concentrano sulle competenze e sui processi aziendali; le prospettive relative alla piattaforma, alla sicurezza e alle operazioni si concentrano sulle competenze e sui processi tecnici. Ad esempio, la prospettiva relativa alle persone si rivolge alle parti interessate che gestiscono le risorse umane (HR), le funzioni del personale e la gestione del personale. In questa prospettiva, AWS CAF fornisce linee

guida per lo sviluppo delle persone, la formazione e le comunicazioni per aiutare a preparare l'organizzazione all'adozione del cloud di successo. Per ulteriori informazioni, consulta il [sito web di AWS CAF](#) e il [white paper AWS CAF](#).

## AWS Workload Qualification Framework (WQF)AWS

Uno strumento che valuta i carichi di lavoro di migrazione dei database, consiglia strategie di migrazione e fornisce stime del lavoro. AWS WQF è incluso in (). AWS Schema Conversion Tool AWS SCT Analizza gli schemi di database e gli oggetti di codice, il codice dell'applicazione, le dipendenze e le caratteristiche delle prestazioni e fornisce report di valutazione.

## B

### bot difettoso

Un [bot](#) che ha lo scopo di disturbare o causare danni a individui o organizzazioni.

### BCP

Vedi la [pianificazione della continuità operativa](#).

### grafico comportamentale

Una vista unificata, interattiva dei comportamenti delle risorse e delle interazioni nel tempo. Puoi utilizzare un grafico comportamentale con Amazon Detective per esaminare tentativi di accesso non riusciti, chiamate API sospette e azioni simili. Per ulteriori informazioni, consulta [Dati in un grafico comportamentale](#) nella documentazione di Detective.

### sistema big-endian

Un sistema che memorizza per primo il byte più importante. Vedi anche [endianness](#).

### Classificazione binaria

Un processo che prevede un risultato binario (una delle due classi possibili). Ad esempio, il modello di machine learning potrebbe dover prevedere problemi come "Questa e-mail è spam o non è spam?" o "Questo prodotto è un libro o un'auto?"

### filtro Bloom

Una struttura di dati probabilistica ed efficiente in termini di memoria che viene utilizzata per verificare se un elemento fa parte di un set.

## blue/green dispiegamento

Una strategia di implementazione in cui si creano due ambienti separati ma identici. La versione corrente dell'applicazione viene eseguita in un ambiente (blu) e la nuova versione dell'applicazione nell'altro ambiente (verde). Questa strategia consente di ripristinare rapidamente il sistema con un impatto minimo.

## bot

Un'applicazione software che esegue attività automatizzate su Internet e simula l'attività o l'interazione umana. Alcuni bot sono utili o utili, come i web crawler che indicizzano le informazioni su Internet. Alcuni altri bot, noti come bot dannosi, hanno lo scopo di disturbare o causare danni a individui o organizzazioni.

## botnet

Reti di [bot](#) infettate da [malware](#) e controllate da un'unica parte, nota come bot herder o bot operator. Le botnet sono il meccanismo più noto per scalare i bot e il loro impatto.

## ramo

Un'area contenuta di un repository di codice. Il primo ramo creato in un repository è il ramo principale. È possibile creare un nuovo ramo a partire da un ramo esistente e quindi sviluppare funzionalità o correggere bug al suo interno. Un ramo creato per sviluppare una funzionalità viene comunemente detto ramo di funzionalità. Quando la funzionalità è pronta per il rilascio, il ramo di funzionalità viene ricongiunto al ramo principale. Per ulteriori informazioni, consulta [Informazioni sulle filiali](#) (documentazione). GitHub

## accesso break-glass

In circostanze eccezionali e tramite una procedura approvata, un mezzo rapido per consentire a un utente di accedere a un sito a Account AWS cui in genere non dispone delle autorizzazioni necessarie. Per ulteriori informazioni, consulta l'indicatore [Implementare le procedure break-glass](#) nella guida. AWS Well-Architected

## strategia brownfield

L'infrastruttura esistente nell'ambiente. Quando si adotta una strategia brownfield per un'architettura di sistema, si progetta l'architettura in base ai vincoli dei sistemi e dell'infrastruttura attuali. Per l'espansione dell'infrastruttura esistente, è possibile combinare strategie brownfield e [greenfield](#).

## cache del buffer

L'area di memoria in cui sono archiviati i dati a cui si accede con maggiore frequenza.

## capacità di business

Azioni intraprese da un'azienda per generare valore (ad esempio vendite, assistenza clienti o marketing). Le architetture dei microservizi e le decisioni di sviluppo possono essere guidate dalle capacità aziendali. Per ulteriori informazioni, consulta la sezione [Organizzazione in base alle funzionalità aziendali](#) del whitepaper [Esecuzione di microservizi containerizzati su AWS](#).

## pianificazione della continuità operativa (BCP)

Un piano che affronta il potenziale impatto di un evento che comporta l'interruzione dell'attività, come una migrazione su larga scala, sulle operazioni e consente a un'azienda di riprendere rapidamente le operazioni.

# C

## CAF

Vedi [AWS Cloud Adoption Framework](#).

## implementazione canaria

Il rilascio lento e incrementale di una versione agli utenti finali. Quando sei sicuro, distribuisce la nuova versione e sostituisci la versione corrente nella sua interezza.

## CoE

Vedi [Cloud Center of Excellence](#).

## CDC

Vedi [Change Data Capture](#).

## Change Data Capture (CDC)

Il processo di tracciamento delle modifiche a un'origine dati, ad esempio una tabella di database, e di registrazione dei metadati relativi alla modifica. È possibile utilizzare CDC per vari scopi, ad esempio il controllo o la replica delle modifiche in un sistema di destinazione per mantenere la sincronizzazione.

## ingegneria del caos

Introduzione intenzionale di guasti o eventi dirompenti per testare la resilienza di un sistema. Puoi usare [AWS Fault Injection Service \(AWS FIS\)](#) per eseguire esperimenti che stressano i tuoi AWS carichi di lavoro e valutarne la risposta.

## CI/CD

Vedi [integrazione continua e distribuzione continua](#).

## classificazione

Un processo di categorizzazione che aiuta a generare previsioni. I modelli di ML per problemi di classificazione prevedono un valore discreto. I valori discreti sono sempre distinti l'uno dall'altro. Ad esempio, un modello potrebbe dover valutare se in un'immagine è presente o meno un'auto.

## Sviluppatore cittadino

Un utente aziendale che crea applicazioni di intelligenza artificiale utilizzando piattaforme senza code/low codice senza competenze tecniche specializzate.

## crittografia lato client

Crittografia dei dati localmente, prima che il bersaglio li Servizio AWS riceva.

## centro di eccellenza del cloud (CCoE)

Un team multidisciplinare che guida le iniziative di adozione del cloud in tutta l'organizzazione, tra cui lo sviluppo di best practice per il cloud, la mobilitazione delle risorse, la definizione delle tempistiche di migrazione e la guida dell'organizzazione attraverso trasformazioni su larga scala. Per ulteriori informazioni, consulta i [post di CCoE](#) sull' Cloud AWS Enterprise Strategy Blog.

## cloud computing

La tecnologia cloud generalmente utilizzata per l'archiviazione remota di dati e la gestione dei dispositivi IoT. Il cloud computing è generalmente collegato alla tecnologia di [edge computing](#).

## modello operativo cloud

In un'organizzazione IT, il modello operativo utilizzato per creare, maturare e ottimizzare uno o più ambienti cloud. Per ulteriori informazioni, consulta [Building your Cloud Operating Model](#).

## fasi di adozione del cloud

Le quattro fasi che le organizzazioni in genere attraversano quando migrano verso Cloud AWS:

- Progetto: esecuzione di alcuni progetti relativi al cloud per scopi di dimostrazione e apprendimento
- Fondamento: effettuare investimenti fondamentali per dimensionare l'adozione del cloud (ad esempio, creazione di una zona di destinazione, definizione di un CCoE, definizione di un modello operativo)
- Migrazione: migrazione di singole applicazioni
- Re-invention — Ottimizzazione di prodotti e servizi e innovazione nel cloud

Queste fasi sono state definite da Stephen Orban nel post del blog [The Journey Toward Cloud-First & the Stages of Adoption](#) sul blog Enterprise Strategy. Cloud AWS Per informazioni sulla loro relazione con la strategia di AWS migrazione, consulta la guida alla [preparazione alla migrazione](#).

## CMDB

Vedi [database di gestione della configurazione](#).

## repository di codice

Una posizione in cui il codice di origine e altri asset, come documentazione, esempi e script, vengono archiviati e aggiornati attraverso processi di controllo delle versioni. Gli archivi cloud più comuni includono GitHub o Bitbucket Cloud. Ogni versione del codice è denominata ramo. In una struttura a microservizi, ogni repository è dedicato a una singola funzionalità. Una singola CI/CD pipeline può utilizzare più repository.

## cache fredda

Una cache del buffer vuota, non ben popolata o contenente dati obsoleti o irrilevanti. Ciò influisce sulle prestazioni perché l'istanza di database deve leggere dalla memoria o dal disco principale, il che richiede più tempo rispetto alla lettura dalla cache del buffer.

## dati freddi

Dati a cui si accede raramente e che in genere sono storici. Quando si eseguono interrogazioni di questo tipo di dati, le interrogazioni lente sono in genere accettabili. Lo spostamento di questi dati su livelli o classi di storage meno costosi e con prestazioni inferiori può ridurre i costi.

## visione artificiale (CV)

Un campo dell'[intelligenza artificiale](#) che utilizza l'apprendimento automatico per analizzare ed estrarre informazioni da formati visivi come immagini e video digitali. Ad esempio, Amazon SageMaker AI fornisce algoritmi di elaborazione delle immagini per CV.

## deriva della configurazione

Per un carico di lavoro, una modifica della configurazione rispetto allo stato previsto. Potrebbe causare la non conformità del carico di lavoro e in genere è graduale e involontaria.

## database di gestione della configurazione (CMDB)

Un repository che archivia e gestisce le informazioni su un database e il relativo ambiente IT, inclusi i componenti hardware e software e le relative configurazioni. In genere si utilizzano i dati di un CMDB nella fase di individuazione e analisi del portafoglio della migrazione.

## Pacchetto di conformità

Una raccolta di AWS Config regole e azioni correttive che puoi assemblare per personalizzare i controlli di conformità e sicurezza. È possibile distribuire un pacchetto di conformità come singola entità in una regione Account AWS and o all'interno di un'organizzazione utilizzando un modello YAML. Per ulteriori informazioni, consulta i [Conformance](#) Pack nella documentazione. AWS Config

## integrazione e distribuzione continue ( ) CI/CD

Il processo di automazione delle fasi di origine, compilazione, test, gestione temporanea e produzione del processo di rilascio del software. CI/CD viene comunemente descritto come una pipeline. CI/CD può aiutarvi ad automatizzare i processi, migliorare la produttività, migliorare la qualità del codice e velocizzare le consegne. Per ulteriori informazioni, consulta [Vantaggi della distribuzione continua](#). CD può anche significare continuous deployment (implementazione continua). Per ulteriori informazioni, consulta [Distribuzione continua e implementazione continua a confronto](#).

## CV

Vedi [visione artificiale](#).

## D

### dati a riposo

Dati stazionari nella rete, ad esempio i dati archiviati.

### classificazione dei dati

Un processo per identificare e classificare i dati nella rete in base alla loro criticità e sensibilità. È un componente fondamentale di qualsiasi strategia di gestione dei rischi di sicurezza informatica

perché consente di determinare i controlli di protezione e conservazione appropriati per i dati. La classificazione dei dati è un componente del pilastro della sicurezza nel AWS Well-Architected Framework. Per ulteriori informazioni, consulta [Classificazione dei dati](#).

#### deriva dei dati

Una variazione significativa tra i dati di produzione e i dati utilizzati per addestrare un modello di machine learning o una modifica significativa dei dati di input nel tempo. La deriva dei dati può ridurre la qualità, l'accuratezza e l'equità complessive nelle previsioni dei modelli ML.

#### dati in transito

Dati che si spostano attivamente attraverso la rete, ad esempio tra le risorse di rete.

#### rete di dati

Un framework architettonico che fornisce la proprietà distribuita e decentralizzata dei dati con gestione e governance centralizzate.

#### riduzione al minimo dei dati

Il principio della raccolta e del trattamento dei soli dati strettamente necessari. Praticare la riduzione al minimo dei dati in the Cloud AWS può ridurre i rischi per la privacy, i costi e l'impronta di carbonio delle analisi.

#### perimetro dei dati

Una serie di barriere preventive nell' AWS ambiente che aiutano a garantire che solo le identità attendibili accedano alle risorse attendibili delle reti previste. Per ulteriori informazioni, consulta [Building a data perimeter](#) on. AWS

#### pre-elaborazione dei dati

Trasformare i dati grezzi in un formato che possa essere facilmente analizzato dal modello di ML. La pre-elaborazione dei dati può comportare la rimozione di determinate colonne o righe e l'eliminazione di valori mancanti, incoerenti o duplicati.

#### provenienza dei dati

Il processo di tracciamento dell'origine e della cronologia dei dati durante il loro ciclo di vita, ad esempio il modo in cui i dati sono stati generati, trasmessi e archiviati.

#### soggetto dei dati

Un individuo i cui dati vengono raccolti ed elaborati.

## data warehouse

Un sistema di gestione dei dati che supporta la business intelligence, come l'analisi. I data warehouse contengono in genere grandi quantità di dati storici e vengono generalmente utilizzati per interrogazioni e analisi.

## linguaggio di definizione del database (DDL)

Istruzioni o comandi per creare o modificare la struttura di tabelle e oggetti in un database.

## linguaggio di manipolazione del database (DML)

Istruzioni o comandi per modificare (inserire, aggiornare ed eliminare) informazioni in un database.

## DDL

Vedi linguaggio di [definizione del database](#).

## deep ensemble

Combinare più modelli di deep learning per la previsione. È possibile utilizzare i deep ensemble per ottenere una previsione più accurata o per stimare l'incertezza nelle previsioni.

## deep learning

Un sottocampo del ML che utilizza più livelli di reti neurali artificiali per identificare la mappatura tra i dati di input e le variabili target di interesse.

## difesa in profondità

Un approccio alla sicurezza delle informazioni in cui una serie di meccanismi e controlli di sicurezza sono accuratamente stratificati su una rete di computer per proteggere la riservatezza, l'integrità e la disponibilità della rete e dei dati al suo interno. Quando si adotta questa strategia AWS, si aggiungono più controlli a diversi livelli della AWS Organizations struttura per proteggere le risorse. Ad esempio, un approccio di difesa approfondita potrebbe combinare autenticazione a più fattori, segmentazione della rete e crittografia.

## amministratore delegato

In AWS Organizations, un servizio compatibile può registrare un account AWS membro per amministrare gli account dell'organizzazione e gestire le autorizzazioni per quel servizio. Questo account è denominato amministratore delegato per quel servizio specifico. Per ulteriori informazioni e un elenco di servizi compatibili, consulta [Servizi che funzionano con AWS Organizations](#) nella documentazione di AWS Organizations .

## implementazione

Il processo di creazione di un'applicazione, di nuove funzionalità o di correzioni di codice disponibili nell'ambiente di destinazione. L'implementazione prevede l'applicazione di modifiche in una base di codice, seguita dalla creazione e dall'esecuzione di tale base di codice negli ambienti applicativi.

### Ambiente di sviluppo

[Vedi ambiente.](#)

### controllo di rilevamento

Un controllo di sicurezza progettato per rilevare, registrare e avvisare dopo che si è verificato un evento. Questi controlli rappresentano una seconda linea di difesa e avvisano l'utente in caso di eventi di sicurezza che aggirano i controlli preventivi in vigore. Per ulteriori informazioni, consulta [Controlli di rilevamento](#) in Implementazione dei controlli di sicurezza in AWS.

### mappatura del flusso di valore dello sviluppo (DVSM)

Un processo utilizzato per identificare e dare priorità ai vincoli che influiscono negativamente sulla velocità e sulla qualità nel ciclo di vita dello sviluppo del software. DVSM estende il processo di mappatura del flusso di valore originariamente progettato per pratiche di produzione snella. Si concentra sulle fasi e sui team necessari per creare e trasferire valore attraverso il processo di sviluppo del software.

### gemello digitale

Una rappresentazione virtuale di un sistema reale, ad esempio un edificio, una fabbrica, un'attrezzatura industriale o una linea di produzione. I gemelli digitali supportano la manutenzione predittiva, il monitoraggio remoto e l'ottimizzazione della produzione.

### tabella delle dimensioni

In uno [schema a stella](#), una tabella più piccola che contiene gli attributi dei dati quantitativi in una tabella dei fatti. Gli attributi della tabella delle dimensioni sono in genere campi di testo o numeri discreti che si comportano come testo. Questi attributi vengono comunemente utilizzati per il vincolo delle query, il filtraggio e l'etichettatura dei set di risultati.

### disastro

Un evento che impedisce a un carico di lavoro o a un sistema di raggiungere gli obiettivi aziendali nella sua sede principale di implementazione. Questi eventi possono essere disastri naturali,

guasti tecnici o il risultato di azioni umane, come errori di configurazione involontari o attacchi di malware.

## disaster recovery (DR)

La strategia e il processo utilizzati per ridurre al minimo i tempi di inattività e la perdita di dati causati da un [disastro](#). Per ulteriori informazioni, consulta [Disaster Recovery of Workload su AWS: Recovery in the Cloud in the AWS Well-Architected Framework](#).

## DML

Vedi linguaggio di [manipolazione del database](#).

## progettazione basata sul dominio

Un approccio allo sviluppo di un sistema software complesso collegandone i componenti a domini in evoluzione, o obiettivi aziendali principali, perseguiti da ciascun componente. Questo concetto è stato introdotto da Eric Evans nel suo libro *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). Per informazioni su come utilizzare la progettazione basata sul dominio con lo strangler fig pattern, consulta [Modernizzare i servizi Web Microsoft ASP.NET \(ASMX\) legacy in modo incrementale utilizzando contenitori e Amazon API Gateway](#).

## DOTT.

Vedi [disaster recovery](#).

## rilevamento della deriva

Tracciamento delle deviazioni da una configurazione di base. Ad esempio, puoi utilizzarlo AWS CloudFormation per [rilevare la deriva nelle risorse di sistema](#) oppure puoi usarlo AWS Control Tower per [rilevare cambiamenti nella tua landing zone](#) che potrebbero influire sulla conformità ai requisiti di governance.

## DVSM

Vedi la [mappatura del flusso di valore dello sviluppo](#).

## E

### EDA

Vedi [analisi esplorativa dei dati](#).

## MODIFICA

Vedi [scambio elettronico di dati](#).

### edge computing

La tecnologia che aumenta la potenza di calcolo per i dispositivi intelligenti all'edge di una rete IoT. Rispetto al [cloud computing](#), [l'edge computing](#) può ridurre la latenza di comunicazione e migliorare i tempi di risposta.

### scambio elettronico di dati (EDI)

Lo scambio automatizzato di documenti aziendali tra organizzazioni. Per ulteriori informazioni, vedere [Cos'è lo scambio elettronico di dati](#).

### crittografia

Un processo di elaborazione che trasforma i dati in chiaro, leggibili dall'uomo, in testo cifrato.

### chiave crittografica

Una stringa crittografica di bit randomizzati generata da un algoritmo di crittografia. Le chiavi possono variare di lunghezza e ogni chiave è progettata per essere imprevedibile e univoca.

### endianità

L'ordine in cui i byte vengono archiviati nella memoria del computer. Big-endian i sistemi memorizzano per primi il byte più importante. Little-endian i sistemi memorizzano per primi il byte meno importante.

### endpoint

Vedi [service endpoint](#).

### servizio endpoint

Un servizio che puoi ospitare in un cloud privato virtuale (VPC) da condividere con altri utenti. Puoi creare un servizio endpoint con AWS PrivateLink e concedere autorizzazioni ad altri Account AWS o a AWS Identity and Access Management (IAM) principali. Questi account o principali possono connettersi al servizio endpoint in privato creando endpoint VPC di interfaccia. Per ulteriori informazioni, consulta [Creazione di un servizio endpoint](#) nella documentazione di Amazon Virtual Private Cloud (Amazon VPC).

### pianificazione delle risorse aziendali (ERP)

Un sistema che automatizza e gestisce i processi aziendali chiave (come contabilità, [MES](#) e gestione dei progetti) per un'azienda.

## crittografia envelope

Il processo di crittografia di una chiave di crittografia con un'altra chiave di crittografia. Per ulteriori informazioni, vedete [Envelope encryption](#) nella documentazione AWS Key Management Service (AWS KMS).

## ambiente

Un'istanza di un'applicazione in esecuzione. Di seguito sono riportati i tipi di ambiente più comuni nel cloud computing:

- ambiente di sviluppo: un'istanza di un'applicazione in esecuzione disponibile solo per il team principale responsabile della manutenzione dell'applicazione. Gli ambienti di sviluppo vengono utilizzati per testare le modifiche prima di promuoverle negli ambienti superiori. Questo tipo di ambiente viene talvolta definito ambiente di test.
- ambienti inferiori: tutti gli ambienti di sviluppo di un'applicazione, ad esempio quelli utilizzati per le build e i test iniziali.
- ambiente di produzione: un'istanza di un'applicazione in esecuzione a cui gli utenti finali possono accedere. In una CI/CD pipeline, l'ambiente di produzione è l'ultimo ambiente di distribuzione.
- ambienti superiori: tutti gli ambienti a cui possono accedere utenti diversi dal team di sviluppo principale. Si può trattare di un ambiente di produzione, ambienti di preproduzione e ambienti per i test di accettazione da parte degli utenti.

## epica

Nelle metodologie agili, categorie funzionali che aiutano a organizzare e dare priorità al lavoro. Le epiche forniscono una descrizione di alto livello dei requisiti e delle attività di implementazione. Ad esempio, le epiche della sicurezza AWS CAF includono la gestione delle identità e degli accessi, i controlli investigativi, la sicurezza dell'infrastruttura, la protezione dei dati e la risposta agli incidenti. Per ulteriori informazioni sulle epiche, consulta la strategia di migrazione AWS , consulta la [guida all'implementazione del programma](#).

## ERP

Vedi [pianificazione delle risorse aziendali](#).

## analisi esplorativa dei dati (EDA)

Il processo di analisi di un set di dati per comprenderne le caratteristiche principali. Si raccolgono o si aggregano dati e quindi si eseguono indagini iniziali per trovare modelli, rilevare anomalie

e verificare ipotesi. L'EDA viene eseguita calcolando statistiche di riepilogo e creando visualizzazioni di dati.

## F

### tabella dei fatti

Il tavolo centrale con [schema a stella](#). Memorizza dati quantitativi sulle operazioni aziendali. In genere, una tabella dei fatti contiene due tipi di colonne: quelle che contengono misure e quelle che contengono una chiave esterna per una tabella di dimensioni.

### fallire velocemente

Una filosofia che utilizza test frequenti e incrementali per ridurre il ciclo di vita dello sviluppo. È una parte fondamentale di un approccio agile.

### limite di isolamento dei guasti

Nel Cloud AWS, un limite come una zona di disponibilità Regione AWS, un piano di controllo o un piano dati che limita l'effetto di un errore e aiuta a migliorare la resilienza dei carichi di lavoro. Per ulteriori informazioni, consulta [AWS Fault Isolation Boundaries](#).

### ramo di funzionalità

Vedi [filiale](#).

### caratteristiche

I dati di input che usi per fare una previsione. Ad esempio, in un contesto di produzione, le caratteristiche potrebbero essere immagini acquisite periodicamente dalla linea di produzione.

### importanza delle caratteristiche

Quanto è importante una caratteristica per le previsioni di un modello. Di solito viene espresso come punteggio numerico che può essere calcolato con varie tecniche, come Shapley Additive Explanations (SHAP) e gradienti integrati. Per ulteriori informazioni, consulta [Interpretabilità del modello di machine learning con AWS](#).

### trasformazione delle funzionalità

Per ottimizzare i dati per il processo di machine learning, incluso l'arricchimento dei dati con fonti aggiuntive, il dimensionamento dei valori o l'estrazione di più set di informazioni da un singolo campo di dati. Ciò consente al modello di ML di trarre vantaggio dai dati. Ad esempio, se suddividi

la data "2021-05-27 00:15:37" in "2021", "maggio", "giovedì" e "15", puoi aiutare l'algoritmo di apprendimento ad apprendere modelli sfumati associati a diversi componenti dei dati.

prompt con pochi scatti

Fornire a un [LLM](#) un numero limitato di esempi che dimostrino l'attività e il risultato desiderato prima di chiedergli di eseguire un'attività simile. Questa tecnica è un'applicazione dell'apprendimento contestuale, in cui i modelli imparano da esempi (immagini) incorporati nei prompt. Few-shot i suggerimenti possono essere efficaci per attività che richiedono una formattazione, un ragionamento o una conoscenza del dominio specifici. [Vedi anche zero-shot prompting.](#)

FGAC

Vedi il controllo [granulare degli accessi](#).

controllo granulare degli accessi (FGAC)

L'uso di più condizioni per consentire o rifiutare una richiesta di accesso.

migrazione flash-cut

Un metodo di migrazione del database che utilizza la replica continua dei dati tramite [l'acquisizione dei dati delle modifiche](#) per migrare i dati nel più breve tempo possibile, anziché utilizzare un approccio graduale. L'obiettivo è ridurre al minimo i tempi di inattività.

FM

[Vedi il modello di base.](#)

modello di fondazione (FM)

Una grande rete neurale di deep learning che si è addestrata su enormi set di dati generalizzati e non etichettati. Le FM sono in grado di eseguire un'ampia varietà di attività generali, come comprendere il linguaggio, generare testo e immagini e conversare in linguaggio naturale. Per ulteriori informazioni, consulta [Cosa sono i modelli Foundation](#).

Gateway FM

[Un intermediario centralizzato che controlla e normalizza l'accesso ai modelli di base.](#) Conosciuto anche come gateway LLM.

# G

## IA generativa

Un sottoinsieme di modelli di [intelligenza artificiale](#) che sono stati addestrati su grandi quantità di dati e che possono utilizzare un semplice messaggio di testo per creare nuovi contenuti e artefatti, come immagini, video, testo e audio. Per ulteriori informazioni, consulta [Cos'è l'IA generativa](#).

## blocco geografico

Vedi [restrizioni geografiche](#).

## limitazioni geografiche (blocco geografico)

In Amazon CloudFront, un'opzione per impedire agli utenti di determinati paesi di accedere alle distribuzioni di contenuti. Puoi utilizzare un elenco consentito o un elenco di blocco per specificare i paesi approvati e vietati. Per ulteriori informazioni, consulta [Limitare la distribuzione geografica dei contenuti](#) nella CloudFront documentazione.

## Flusso di lavoro di GitFlow

Un approccio in cui gli ambienti inferiori e superiori utilizzano rami diversi in un repository di codice di origine. Il flusso di lavoro Gitflow è considerato obsoleto e il flusso di lavoro [basato su trunk è l'approccio moderno e preferito](#).

## immagine dorata

Un'istantanea di un sistema o di un software che viene utilizzata come modello per distribuire nuove istanze di quel sistema o software. Ad esempio, nella produzione, un'immagine dorata può essere utilizzata per fornire software su più dispositivi e contribuire a migliorare la velocità, la scalabilità e la produttività nelle operazioni di produzione dei dispositivi.

## strategia greenfield

L'assenza di infrastrutture esistenti in un nuovo ambiente. Quando si adotta una strategia greenfield per un'architettura di sistema, è possibile selezionare tutte le nuove tecnologie senza il vincolo della compatibilità con l'infrastruttura esistente, nota anche come [brownfield](#). Per l'espansione dell'infrastruttura esistente, è possibile combinare strategie brownfield e greenfield.

## guardrail

Una regola di livello elevato che consente di governare risorse, policy e conformità tra le unità organizzative (OU). I guardrail preventivi applicano le policy per garantire l'allineamento agli

standard di conformità. Vengono implementati utilizzando le policy di controllo dei servizi e i limiti delle autorizzazioni IAM. I guardrail di rilevamento rilevano le violazioni delle policy e i problemi di conformità e generano avvisi per porvi rimedio. Sono implementati utilizzando Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector e controlli personalizzati AWS Lambda .

## guardrail (AI)

Meccanismi di sicurezza che filtrano, convalidano e limitano gli input e gli output degli [agenti](#) per contribuire a garantire un comportamento dell'IA responsabile e sicuro.

# H

## AH

Vedi [disponibilità elevata](#).

## migrazione di database eterogenea

Migrazione del database di origine in un database di destinazione che utilizza un motore di database diverso (ad esempio, da Oracle ad Amazon Aurora). La migrazione eterogenea fa in genere parte di uno sforzo di riprogettazione e la conversione dello schema può essere un'attività complessa. [AWS offre AWS SCT](#) che aiuta con le conversioni dello schema.

## alta disponibilità (HA)

La capacità di un carico di lavoro di funzionare in modo continuo, senza intervento, in caso di sfide o disastri. I sistemi HA sono progettati per il failover automatico, fornire costantemente prestazioni di alta qualità e gestire carichi e guasti diversi con un impatto minimo sulle prestazioni.

## modernizzazione storica

Un approccio utilizzato per modernizzare e aggiornare i sistemi di tecnologia operativa (OT) per soddisfare meglio le esigenze dell'industria manifatturiera. Uno storico è un tipo di database utilizzato per raccogliere e archiviare dati da varie fonti in una fabbrica.

## dati di esclusione

[Una parte di dati storici etichettati che viene trattenuta da un set di dati utilizzata per addestrare un modello di apprendimento automatico.](#) È possibile utilizzare i dati di holdout per valutare le prestazioni del modello confrontando le previsioni del modello con i dati di holdout.

## human-in-the-loop (HITL)

Un modello di flusso di lavoro in cui l'esecuzione degli [agenti](#) viene sospesa per la revisione e l'approvazione umana nei punti decisionali critici.

## migrazione di database omogenea

Migrazione del database di origine in un database di destinazione che condivide lo stesso motore di database (ad esempio, da Microsoft SQL Server ad Amazon RDS per SQL Server). La migrazione omogenea fa in genere parte di un'operazione di rehosting o ridefinizione della piattaforma. Per migrare lo schema è possibile utilizzare le utilità native del database.

## dati caldi

Dati a cui si accede frequentemente, ad esempio dati in tempo reale o dati di traduzione recenti. Questi dati richiedono in genere un livello o una classe di storage ad alte prestazioni per fornire risposte rapide alle query.

## hotfix

Una soluzione urgente per un problema critico in un ambiente di produzione. A causa della sua urgenza, un hotfix viene in genere creato al di fuori del tipico DevOps flusso di lavoro di rilascio.

## periodo di hypercare

Subito dopo la conversione, il periodo di tempo in cui un team di migrazione gestisce e monitora le applicazioni migrate nel cloud per risolvere eventuali problemi. In genere, questo periodo dura da 1 a 4 giorni. Al termine del periodo di hypercare, il team addetto alla migrazione in genere trasferisce la responsabilità delle applicazioni al team addetto alle operazioni cloud.

I

## IaC

Vedi l'[infrastruttura come codice](#).

## Policy basata su identità

Una policy associata a uno o più principi IAM che definisce le relative autorizzazioni all'interno dell'Cloud AWS ambiente.

I

## applicazione inattiva

Un'applicazione che prevede un uso di CPU e memoria medio compreso tra il 5% e il 20% in un periodo di 90 giorni. In un progetto di migrazione, è normale ritirare queste applicazioni o mantenerle on-premise.

## IloT

Vedi [Industrial Internet of Things](#).

## infrastruttura immutabile

Un modello che implementa una nuova infrastruttura per i carichi di lavoro di produzione anziché aggiornare, applicare patch o modificare l'infrastruttura esistente. [Le infrastrutture immutabili sono intrinsecamente più coerenti, affidabili e prevedibili delle infrastrutture mutabili](#). Per ulteriori informazioni, consulta la best practice [Deploy using immutable](#) infrastrutture nel Framework. AWS Well-Architected

## VPC in ingresso (ingress)

In un'architettura AWS multi-account, un VPC che accetta, ispeziona e indirizza le connessioni di rete dall'esterno di un'applicazione. Nel documento [Architettura di riferimento per la sicurezza di AWS](#) si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

## migrazione incrementale

Una strategia di conversione in cui si esegue la migrazione dell'applicazione in piccole parti anziché eseguire una conversione singola e completa. Ad esempio, inizialmente potresti spostare solo alcuni microservizi o utenti nel nuovo sistema. Dopo aver verificato che tutto funzioni correttamente, puoi spostare in modo incrementale microservizi o utenti aggiuntivi fino alla disattivazione del sistema legacy. Questa strategia riduce i rischi associati alle migrazioni di grandi dimensioni.

## Industria 4.0

Un termine introdotto da [Klaus Schwab](#) nel 2016 per riferirsi alla modernizzazione dei processi di produzione attraverso progressi in termini di connettività, dati in tempo reale, automazione, analisi e. AI/ML

## infrastruttura

Tutte le risorse e gli asset contenuti nell'ambiente di un'applicazione.

## infrastruttura come codice (IaC)

Il processo di provisioning e gestione dell'infrastruttura di un'applicazione tramite un insieme di file di configurazione. Il processo IaC è progettato per aiutarti a centralizzare la gestione dell'infrastruttura, a standardizzare le risorse e a dimensionare rapidamente, in modo che i nuovi ambienti siano ripetibili, affidabili e coerenti.

## Internet delle cose industriale (IIoT)

L'uso di sensori e dispositivi connessi a Internet nei settori industriali, come quello manifatturiero, energetico, automobilistico, sanitario, delle scienze della vita e dell'agricoltura. Per ulteriori informazioni, consulta [Creazione di una strategia di trasformazione digitale dell'Internet delle cose industriale \(IIoT\)](#).

## VPC di ispezione

In un'architettura AWS multi-account, un VPC centralizzato che gestisce le ispezioni del traffico di rete tra VPC (uguali o diversi Regioni AWS), Internet e reti locali. Nel documento [Architettura di riferimento per la sicurezza di AWS](#) si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

## Internet of Things (IoT)

La rete di oggetti fisici connessi con sensori o processori incorporati che comunicano con altri dispositivi e sistemi tramite Internet o una rete di comunicazione locale. Per ulteriori informazioni, consulta [Cos'è l'IoT?](#)

## interpretabilità

Una caratteristica di un modello di machine learning che descrive il grado in cui un essere umano è in grado di comprendere in che modo le previsioni del modello dipendono dai suoi input. [Per ulteriori informazioni, consulta Interpretabilità del modello di machine learning con. AWS](#)

## IoT

Vedi [Internet of Things](#).

## libreria di informazioni IT (ITIL)

Una serie di best practice per offrire servizi IT e allinearli ai requisiti aziendali. ITIL fornisce le basi per ITSM.

## gestione dei servizi IT (ITSM)

Attività associate alla progettazione, implementazione, gestione e supporto dei servizi IT per un'organizzazione. Per informazioni sull'integrazione delle operazioni cloud con gli strumenti ITSM, consulta la [guida all'integrazione delle operazioni](#).

## ITIL

Vedi la [libreria di informazioni IT](#).

## ITSM

Vedi [Gestione dei servizi IT](#).

## L

### controllo degli accessi basato su etichette (LBAC)

Un'implementazione del controllo di accesso obbligatorio (MAC) in cui agli utenti e ai dati stessi viene assegnato esplicitamente un valore di etichetta di sicurezza. L'intersezione tra l'etichetta di sicurezza utente e l'etichetta di sicurezza dei dati determina quali righe e colonne possono essere visualizzate dall'utente.

### zona di destinazione

Una landing zone è un AWS ambiente multi-account ben progettato, scalabile e sicuro. Questo è un punto di partenza dal quale le organizzazioni possono avviare e distribuire rapidamente carichi di lavoro e applicazioni con fiducia nel loro ambiente di sicurezza e infrastruttura. Per ulteriori informazioni sulle zone di destinazione, consulta la sezione [Configurazione di un ambiente AWS multi-account sicuro e scalabile](#).

### modello linguistico di grandi dimensioni (LLM)

Un modello di [intelligenza artificiale](#) di deep learning preaddestrato su una grande quantità di dati. Un LLM può svolgere più attività, come rispondere a domande, riepilogare documenti, tradurre testo in altre lingue e completare frasi. [Per ulteriori informazioni, consulta Cosa sono gli LLM](#).

### migrazione su larga scala

Una migrazione di 300 o più server.

## BIANCO

Vedi controllo degli accessi [basato su etichette](#).

## Privilegio minimo

La best practice di sicurezza per la concessione delle autorizzazioni minime richieste per eseguire un'attività. Per ulteriori informazioni, consulta [Applicazione delle autorizzazioni del privilegio minimo](#) nella documentazione di IAM.

eseguire il rehosting (lift and shift)

Vedi [7 R](#).

sistema little-endian

Un sistema che memorizza per primo il byte meno importante. Vedi anche [endianità](#).

LLM

Vedi modello [linguistico di grandi dimensioni](#).

ambienti inferiori

Vedi [ambiente](#).

## M

machine learning (ML)

Un tipo di intelligenza artificiale che utilizza algoritmi e tecniche per il riconoscimento e l'apprendimento di schemi. Il machine learning analizza e apprende dai dati registrati, come i dati dell'Internet delle cose (IoT), per generare un modello statistico basato su modelli. Per ulteriori informazioni, consulta la sezione [Machine learning](#).

ramo principale

Vedi [filiale](#).

malware

Software progettato per compromettere la sicurezza o la privacy del computer. Il malware potrebbe interrompere i sistemi informatici, divulgare informazioni sensibili o ottenere accessi non autorizzati. Esempi di malware includono virus, worm, ransomware, trojan horse, spyware e keylogger.

## servizi gestiti

Servizi AWS per cui AWS gestisce il livello di infrastruttura, il sistema operativo e le piattaforme e si accede agli endpoint per archiviare e recuperare i dati. Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) e Amazon DynamoDB sono esempi di servizi gestiti. Questi sono noti anche come servizi astratti.

## sistema di esecuzione della produzione (MES)

Un sistema software per tracciare, monitorare, documentare e controllare i processi di produzione che convertono le materie prime in prodotti finiti in officina.

## MAP

Vedi [Migration Acceleration Program](#).

## MCP

Vedi [Model Context Protocol](#).

## Model Context Protocol (MCP)

[Un protocollo stateless per la comunicazione tra agenti e strumenti](#).

## Server MCP

Un servizio che espone uno o più [strumenti](#) tramite il [Model Context](#) Protocol.

## meccanismo

Un processo completo in cui si crea uno strumento, si promuove l'adozione dello strumento e quindi si esaminano i risultati per apportare le modifiche. Un meccanismo è un ciclo che si rafforza e si migliora man mano che funziona. Per ulteriori informazioni, vedete [Creazione di meccanismi](#) nel AWS Well-Architected Framework.

## account membro

Tutti gli account Account AWS diversi dall'account di gestione che fanno parte di un'organizzazione in AWS Organizations. Un account può essere membro di una sola organizzazione alla volta.

## MEH

Vedi [sistema di esecuzione della produzione](#).

## Message Queuing Telemetry Transport (MQTT)

[Un protocollo di comunicazione da macchina a macchina \(M2M\) leggero, basato sul publish/subscribe modello, per dispositivi IoT con risorse limitate.](#)

### microservizio

Un piccolo servizio indipendente che comunica tramite API ben definite ed è in genere di proprietà di piccoli team autonomi. Ad esempio, un sistema assicurativo potrebbe includere microservizi che si riferiscono a funzionalità aziendali, come vendite o marketing, o sottodomini, come acquisti, reclami o analisi. I vantaggi dei microservizi includono agilità, dimensionamento flessibile, facilità di implementazione, codice riutilizzabile e resilienza. [Per ulteriori informazioni, consulta Integrazione dei microservizi utilizzando servizi serverless. AWS](#)

### architettura di microservizi

Un approccio alla creazione di un'applicazione con componenti indipendenti che eseguono ogni processo applicativo come microservizio. Questi microservizi comunicano tramite un'interfaccia ben definita utilizzando API leggere. Ogni microservizio in questa architettura può essere aggiornato, distribuito e dimensionato per soddisfare la richiesta di funzioni specifiche di un'applicazione. Per ulteriori informazioni, vedere [Implementazione](#) dei microservizi su AWS

### Programma di accelerazione della migrazione (MAP)

Un AWS programma che fornisce consulenza, supporto, formazione e servizi per aiutare le organizzazioni a costruire una solida base operativa per il passaggio al cloud e per contribuire a compensare il costo iniziale delle migrazioni. MAP include una metodologia di migrazione per eseguire le migrazioni precedenti in modo metodico e un set di strumenti per automatizzare e accelerare gli scenari di migrazione comuni.

### migrazione su larga scala

Il processo di trasferimento della maggior parte del portfolio di applicazioni sul cloud avviene a ondate, con più applicazioni trasferite a una velocità maggiore in ogni ondata. Questa fase utilizza le migliori pratiche e le lezioni apprese nelle fasi precedenti per implementare una fabbrica di migrazione di team, strumenti e processi per semplificare la migrazione dei carichi di lavoro attraverso l'automazione e la distribuzione agile. Questa è la terza fase della [strategia di migrazione AWS](#).

### fabbrica di migrazione

Cross-functional team che semplificano la migrazione dei carichi di lavoro attraverso approcci automatizzati e agili. I team di Migration Factory includono in genere operazioni, analisti e

proprietari aziendali, ingegneri addetti alla migrazione, sviluppatori e DevOps professionisti che lavorano nell'ambito degli sprint. Tra il 20% e il 50% di un portfolio di applicazioni aziendali è costituito da schemi ripetuti che possono essere ottimizzati con un approccio di fabbrica. Per ulteriori informazioni, consulta la [discussione sulle fabbriche di migrazione](#) e la [Guida alla fabbrica di migrazione al cloud](#) in questo set di contenuti.

#### metadati di migrazione

Le informazioni sull'applicazione e sul server necessarie per completare la migrazione. Ogni modello di migrazione richiede un set diverso di metadati di migrazione. Esempi di metadati di migrazione includono la sottorete, il gruppo di sicurezza e l'account di destinazione. AWS

#### modello di migrazione

Un'attività di migrazione ripetibile che descrive in dettaglio la strategia di migrazione, la destinazione della migrazione e l'applicazione o il servizio di migrazione utilizzati. Esempio: riorganizza la migrazione su Amazon EC2 AWS con Application Migration Service.

#### Valutazione del portfolio di migrazione (MPA)

Uno strumento online che fornisce informazioni per la convalida del business case per la migrazione a. Cloud AWS MPA offre una valutazione dettagliata del portfolio (dimensionamento corretto dei server, prezzi, confronto del TCO, analisi dei costi di migrazione) e pianificazione della migrazione (analisi e raccolta dei dati delle applicazioni, raggruppamento delle applicazioni, prioritizzazione delle migrazioni e pianificazione delle ondate). [Lo strumento MPA](#) (richiede l'accesso) è disponibile gratuitamente per tutti i AWS consulenti e i consulenti dei partner APN.

#### valutazione della preparazione alla migrazione (MRA)

Il processo di acquisizione di informazioni sullo stato di preparazione al cloud di un'organizzazione, l'identificazione dei punti di forza e di debolezza e la creazione di un piano d'azione per colmare le lacune identificate, utilizzando il CAF. AWS Per ulteriori informazioni, consulta la [guida di preparazione alla migrazione](#). MRA è la prima fase della [strategia di migrazione AWS](#).

#### strategia di migrazione

L'approccio utilizzato per migrare un carico di lavoro verso. Cloud AWS Per ulteriori informazioni, consulta la voce [7 R](#) in questo glossario e consulta [Mobilita la tua organizzazione per](#) accelerare le migrazioni su larga scala.

#### ML

[Vedi machine learning.](#)

## modernizzazione

Trasformazione di un'applicazione obsoleta (legacy o monolitica) e della relativa infrastruttura in un sistema agile, elastico e altamente disponibile nel cloud per ridurre i costi, aumentare l'efficienza e sfruttare le innovazioni. Per ulteriori informazioni, vedere [Strategia per la modernizzazione delle applicazioni in](#). Cloud AWS

## valutazione della preparazione alla modernizzazione

Una valutazione che aiuta a determinare la preparazione alla modernizzazione delle applicazioni di un'organizzazione, identifica vantaggi, rischi e dipendenze e determina in che misura l'organizzazione può supportare lo stato futuro di tali applicazioni. Il risultato della valutazione è uno schema dell'architettura di destinazione, una tabella di marcia che descrive in dettaglio le fasi di sviluppo e le tappe fondamentali del processo di modernizzazione e un piano d'azione per colmare le lacune identificate. Per ulteriori informazioni, vedere [Valutazione della preparazione alla modernizzazione per](#) le applicazioni in. Cloud AWS

## applicazioni monolitiche (monoliti)

Applicazioni eseguite come un unico servizio con processi strettamente collegati. Le applicazioni monolitiche presentano diversi inconvenienti. Se una funzionalità dell'applicazione registra un picco di domanda, l'intera architettura deve essere dimensionata. L'aggiunta o il miglioramento delle funzionalità di un'applicazione monolitica diventa inoltre più complessa man mano che la base di codice cresce. Per risolvere questi problemi, puoi utilizzare un'architettura di microservizi. Per ulteriori informazioni, consulta la sezione [Scomposizione dei monoliti in microservizi](#).

## MAPPA

Vedi [Migration Portfolio Assessment](#).

## MQTT

Vedi [Message Queuing Telemetry Transport](#).

## classificazione multiclasse

Un processo che aiuta a generare previsioni per più classi (prevedendo uno o più di due risultati). Ad esempio, un modello di machine learning potrebbe chiedere "Questo prodotto è un libro, un'auto o un telefono?" oppure "Quale categoria di prodotti è più interessante per questo cliente?"

## infrastruttura mutabile

Un modello che aggiorna e modifica l'infrastruttura esistente per i carichi di lavoro di produzione. Per migliorare la coerenza, l'affidabilità e la prevedibilità, il AWS Well-Architected Framework consiglia l'uso di un'infrastruttura [immutabile](#) come best practice.

## O

### OAC

Vedi [Origin Access Control](#).

### QUERCIA

Vedi [Origin Access Identity](#).

### OCM

Vedi [gestione delle modifiche organizzative](#).

## migrazione offline

Un metodo di migrazione in cui il carico di lavoro di origine viene eliminato durante il processo di migrazione. Questo metodo prevede tempi di inattività prolungati e viene in genere utilizzato per carichi di lavoro piccoli e non critici.

## OI

Vedi [l'integrazione delle operazioni](#).

### OLA

Vedi accordo a [livello operativo](#).

## migrazione online

Un metodo di migrazione in cui il carico di lavoro di origine viene copiato sul sistema di destinazione senza essere messo offline. Le applicazioni connesse al carico di lavoro possono continuare a funzionare durante la migrazione. Questo metodo comporta tempi di inattività pari a zero o comunque minimi e viene in genere utilizzato per carichi di lavoro di produzione critici.

### OPC-UA

Vedi [Open Process Communications](#) - Unified Architecture.

## Comunicazioni a processo aperto - Architettura unificata () OPC-UA

Un protocollo di comunicazione da macchina a macchina (M2M) per l'automazione industriale. OPC-UA fornisce uno standard di interoperabilità con schemi di crittografia, autenticazione e autorizzazione dei dati.

### accordo a livello operativo (OLA)

Un accordo che chiarisce quali sono gli impegni reciproci tra i gruppi IT funzionali, a supporto di un accordo sul livello di servizio (SLA).

### revisione della prontezza operativa (ORR)

Un elenco di domande e best practice associate che aiutano a comprendere, valutare, prevenire o ridurre la portata degli incidenti e dei possibili guasti. Per ulteriori informazioni, vedere [Operational Readiness Reviews \(ORR\)](#) nel Framework. AWS Well-Architected

### tecnologia operativa (OT)

Sistemi hardware e software che interagiscono con l'ambiente fisico per controllare operazioni, apparecchiature e infrastrutture industriali. Nella produzione, l'integrazione di sistemi OT e di tecnologia dell'informazione (IT) è un obiettivo chiave per le trasformazioni [dell'Industria 4.0](#).

### integrazione delle operazioni (OI)

Il processo di modernizzazione delle operazioni nel cloud, che prevede la pianificazione, l'automazione e l'integrazione della disponibilità. Per ulteriori informazioni, consulta la [guida all'integrazione delle operazioni](#).

### trail organizzativo

Un percorso creato da noi AWS CloudTrail che registra tutti gli eventi di un'organizzazione per tutti Account AWS . AWS Organizations Questo percorso viene creato in ogni Account AWS che fa parte dell'organizzazione e tiene traccia dell'attività in ogni account. Per ulteriori informazioni, consulta [Creazione di un percorso per un'organizzazione](#) nella CloudTrail documentazione.

### gestione del cambiamento organizzativo (OCM)

Un framework per la gestione di trasformazioni aziendali importanti e che comportano l'interruzione delle attività dal punto di vista delle persone, della cultura e della leadership. OCM aiuta le organizzazioni a prepararsi e passare a nuovi sistemi e strategie accelerando l'adozione del cambiamento, affrontando i problemi di transizione e promuovendo cambiamenti culturali e organizzativi. Nella strategia di AWS migrazione, questo framework si chiama accelerazione delle

persone, a causa della velocità di cambiamento richiesta nei progetti di adozione del cloud. Per ulteriori informazioni, consultare la [Guida OCM](#).

#### controllo dell'accesso all'origine (OAC)

In CloudFront, un'opzione avanzata per limitare l'accesso per proteggere i contenuti di Amazon Simple Storage Service (Amazon S3). OAC supporta in tutto tutti i bucket S3 Regioni AWS, la crittografia lato server con AWS KMS (SSE-KMS) e le richieste dinamiche PUT e dirette al bucket S3. DELETE

#### identità di accesso origine (OAI)

Nel CloudFront, un'opzione per limitare l'accesso per proteggere i tuoi contenuti Amazon S3. Quando usi OAI, CloudFront crea un principale con cui Amazon S3 può autenticarsi. I principali autenticati possono accedere ai contenuti in un bucket S3 solo tramite una distribuzione specifica. CloudFront Vedi anche [OAC](#), che fornisce un controllo degli accessi più granulare e avanzato.

#### ORR

[Vedi la revisione della prontezza operativa.](#)

#### - NON

Vedi la [tecnologia operativa](#).

#### VPC in uscita (egress)

In un'architettura AWS multi-account, un VPC che gestisce le connessioni di rete avviate dall'interno di un'applicazione. Nel documento [Architettura di riferimento per la sicurezza di AWS](#) si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

## P

#### limite delle autorizzazioni

Una policy di gestione IAM collegata ai principali IAM per impostare le autorizzazioni massime che l'utente o il ruolo possono avere. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni](#) nella documentazione di IAM.

## informazioni di identificazione personale (PII)

Informazioni che, se visualizzate direttamente o abbinate ad altri dati correlati, possono essere utilizzate per dedurre ragionevolmente l'identità di un individuo. Esempi di informazioni personali includono nomi, indirizzi e informazioni di contatto.

Informazioni che consentono l'identificazione personale degli utenti

Visualizza le [informazioni di identificazione personale](#).

## playbook

Una serie di passaggi predefiniti che raccolgono il lavoro associato alle migrazioni, come l'erogazione delle funzioni operative principali nel cloud. Un playbook può assumere la forma di script, runbook automatici o un riepilogo dei processi o dei passaggi necessari per gestire un ambiente modernizzato.

## PLC

Vedi [controllore logico programmabile](#).

## PLM

Vedi la gestione [del ciclo di vita del prodotto](#).

## policy

[Un oggetto in grado di definire le autorizzazioni \(vedi politica basata sull'identità\), specificare le condizioni di accesso \(vedi politicabasata sulle risorse\) o definire le autorizzazioni massime per tutti gli account di un'organizzazione in \(vedi politica di controllo dei servizi\). AWS Organizations](#)

## persistenza poliglotta

Scelta indipendente della tecnologia di archiviazione di dati di un microservizio in base ai modelli di accesso ai dati e ad altri requisiti. Se i microservizi utilizzano la stessa tecnologia di archiviazione di dati, possono incontrare problemi di implementazione o registrare prestazioni scadenti. I microservizi vengono implementati più facilmente e ottengono prestazioni e scalabilità migliori se utilizzano l'archivio dati più adatto alle loro esigenze.

## valutazione del portfolio

Un processo di scoperta, analisi e definizione delle priorità del portfolio di applicazioni per pianificare la migrazione. Per ulteriori informazioni, consulta la pagina [Valutazione della preparazione alla migrazione](#).

## predicate

Una condizione di interrogazione che restituisce o, in genere, si trova in una clausola `true`. `false`  
`WHERE`

## predicato pushdown

Una tecnica di ottimizzazione delle query del database che filtra i dati della query prima del trasferimento. Ciò riduce la quantità di dati che devono essere recuperati ed elaborati dal database relazionale e migliora le prestazioni delle query.

## controllo preventivo

Un controllo di sicurezza progettato per impedire il verificarsi di un evento. Questi controlli sono la prima linea di difesa per impedire accessi non autorizzati o modifiche indesiderate alla rete. Per ulteriori informazioni, consulta [Controlli preventivi](#) in Implementazione dei controlli di sicurezza in AWS.

## principale

Un'entità in AWS grado di eseguire azioni e accedere alle risorse. Questa entità è in genere un utente root per un Account AWS ruolo IAM o un utente. Per ulteriori informazioni, consulta Principali in [Termini e concetti dei ruoli](#) nella documentazione di IAM.

## privacy fin dalla progettazione

Un approccio ingegneristico dei sistemi che tiene conto della privacy durante l'intero processo di sviluppo.

## zone ospitate private

Un container che contiene informazioni su come si desidera che Amazon Route 53 risponda alle query DNS per un dominio e i relativi sottodomini all'interno di uno o più VPC. Per ulteriori informazioni, consulta [Utilizzo delle zone ospitate private](#) nella documentazione di Route 53.

## controllo proattivo

Un [controllo di sicurezza](#) progettato per impedire l'implementazione di risorse non conformi. Questi controlli analizzano le risorse prima del loro provisioning. Se la risorsa non è conforme al controllo, non viene fornita. Per ulteriori informazioni, consulta la [guida di riferimento sui controlli](#) nella AWS Control Tower documentazione e consulta Controlli [proattivi in Implementazione dei controlli](#) di sicurezza su. AWS

## gestione del ciclo di vita del prodotto (PLM)

La gestione dei dati e dei processi di un prodotto durante l'intero ciclo di vita, dalla progettazione, sviluppo e lancio, attraverso la crescita e la maturità, fino al declino e alla rimozione.

## Ambiente di produzione

[Vedi ambiente.](#)

## controllore logico programmabile (PLC)

Nella produzione, un computer altamente affidabile e adattabile che monitora le macchine e automatizza i processi di produzione.

## concatenamento rapido

Utilizzo dell'output di un prompt [LLM](#) come input per il prompt successivo per generare risposte migliori. Questa tecnica viene utilizzata per suddividere un'attività complessa in sottoattività o per perfezionare o espandere iterativamente una risposta preliminare. Aiuta a migliorare l'accuratezza e la pertinenza delle risposte di un modello e consente risultati più granulari e personalizzati.

## pseudonimizzazione

Il processo di sostituzione degli identificatori personali in un set di dati con valori segnaposto. La pseudonimizzazione può aiutare a proteggere la privacy personale. I dati pseudonimizzati sono ancora considerati dati personali.

## publish/subscribe (pub/sub)

Un modello che consente comunicazioni asincrone tra microservizi per migliorare la scalabilità e la reattività. Ad esempio, in un [MES](#) basato su microservizi, un microservizio può pubblicare messaggi di eventi su un canale a cui altri microservizi possono abbonarsi. Il sistema può aggiungere nuovi microservizi senza modificare il servizio di pubblicazione.

## Q

### Piano di query

Una serie di passaggi, come le istruzioni, utilizzati per accedere ai dati in un sistema di database relazionale SQL.

## regressione del piano di query

Quando un ottimizzatore del servizio di database sceglie un piano non ottimale rispetto a prima di una determinata modifica all'ambiente di database. Questo può essere causato da modifiche a statistiche, vincoli, impostazioni dell'ambiente, associazioni dei parametri di query e aggiornamenti al motore di database.

# R

## Matrice RACI

Vedi [responsabile, responsabile, consultato, informato \(RACI\)](#).

## RAG

Vedi [Retrieval](#) Augmented Generation.

## ransomware

Un software dannoso progettato per bloccare l'accesso a un sistema informatico o ai dati fino a quando non viene effettuato un pagamento.

## Matrice RASCI

Vedi [responsabile, responsabile, consultato, informato \(RACI\)](#).

## RCAC

Vedi controllo dell'[accesso a righe e colonne](#).

## replica di lettura

Una copia di un database utilizzata per scopi di sola lettura. È possibile indirizzare le query alla replica di lettura per ridurre il carico sul database principale.

## riprogettare

Vedi [7 Rs](#).

## obiettivo del punto di ripristino (RPO)

Il periodo di tempo massimo accettabile dall'ultimo punto di ripristino dei dati. Questo determina ciò che si considera una perdita di dati accettabile tra l'ultimo punto di ripristino e l'interruzione del servizio.

## obiettivo del tempo di ripristino (RTO)

Il ritardo massimo accettabile tra l'interruzione del servizio e il ripristino del servizio.

## rifattorizzare

Vedi [7 R.](#)

## Region

Una raccolta di AWS risorse in un'area geografica. Ciascuna Regione AWS è isolata e indipendente dalle altre per fornire tolleranza agli errori, stabilità e resilienza. Per ulteriori informazioni, consulta [Specificare cosa può usare Regioni AWS il tuo account.](#)

## regressione

Una tecnica di ML che prevede un valore numerico. Ad esempio, per risolvere il problema "A che prezzo verrà venduta questa casa?" un modello di ML potrebbe utilizzare un modello di regressione lineare per prevedere il prezzo di vendita di una casa sulla base di dati noti sulla casa (ad esempio, la metratura).

## riospitare

Vedi [7 R.](#)

## rilascio

In un processo di implementazione, l'atto di promuovere modifiche a un ambiente di produzione.

## trasferisco

Vedi [7 Rs.](#)

## ripiattaforma

Vedi [7 Rs.](#)

## riacquisto

Vedi [7 Rs.](#)

## resilienza

La capacità di un'applicazione di resistere o ripristinare le interruzioni. [L'elevata disponibilità e il disaster recovery](#) sono considerazioni comuni quando si pianifica la resilienza in Cloud AWS. [Per ulteriori informazioni, vedere Cloud AWS Resilience.](#)

## policy basata su risorse

Una policy associata a una risorsa, ad esempio un bucket Amazon S3, un endpoint o una chiave di crittografia. Questo tipo di policy specifica a quali principali è consentito l'accesso, le azioni supportate e qualsiasi altra condizione che deve essere soddisfatta.

## matrice di assegnazione di responsabilità (RACI)

Una matrice che definisce i ruoli e le responsabilità di tutte le parti coinvolte nelle attività di migrazione e nelle operazioni cloud. Il nome della matrice deriva dai tipi di responsabilità definiti nella matrice: responsabile (R), responsabile (A), consultato (C) e informato (I). Il tipo di supporto (S) è facoltativo. Se includi il supporto, la matrice viene chiamata matrice RASCI e, se la escludi, viene chiamata matrice RACI.

## controllo reattivo

Un controllo di sicurezza progettato per favorire la correzione di eventi avversi o deviazioni dalla baseline di sicurezza. Per ulteriori informazioni, consulta [Controlli reattivi](#) in Implementazione dei controlli di sicurezza in AWS.

## retain

Vedi [7 R](#).

## andare in pensione

Vedi [7 Rs](#).

## Retrieval Augmented Generation (RAG)

Una tecnologia di [intelligenza artificiale generativa](#) in cui un [LLM](#) fa riferimento a una fonte di dati autorevole esterna alle sue fonti di dati di formazione prima di generare una risposta. Ad esempio, un modello RAG potrebbe eseguire una ricerca semantica nella knowledge base o nei dati personalizzati di un'organizzazione. Per ulteriori informazioni, consulta [Cos'è il RAG](#).

## rotazione

Processo di aggiornamento periodico di un [segreto](#) per rendere più difficile l'accesso alle credenziali da parte di un utente malintenzionato.

## controllo dell'accesso a righe e colonne (RCAC)

L'uso di espressioni SQL di base e flessibili con regole di accesso definite. RCAC è costituito da autorizzazioni di riga e maschere di colonna.

## RPO

Vedi [obiettivo del punto di ripristino](#).

## VERSO

Vedi [obiettivo del tempo di ripristino](#).

## runbook

Un insieme di procedure manuali o automatizzate necessarie per eseguire un'attività specifica. In genere sono progettati per semplificare operazioni o procedure ripetitive con tassi di errore elevati.

## S

### SAML 2.0

Uno standard aperto utilizzato da molti provider di identità (IdPs). Questa funzionalità abilita il single sign-on (SSO) federato, in modo che gli utenti possano accedere Console di gestione AWS o chiamare le operazioni AWS API senza che tu debba creare un utente in IAM per tutti i membri dell'organizzazione. Per ulteriori informazioni sulla federazione basata su SAML 2.0, consulta [Informazioni sulla federazione basata su SAML 2.0](#) nella documentazione di IAM.

### SCADA

Vedi [controllo di supervisione e acquisizione dati](#).

### SCP

Vedi la [politica di controllo del servizio](#).

### Secret

In Gestione dei segreti AWS, informazioni riservate o riservate, come una password o le credenziali utente, archiviate in forma crittografata. È costituito dal valore segreto e dai relativi metadati. Il valore segreto può essere binario, una stringa singola o più stringhe. Per ulteriori informazioni, consulta [Cosa c'è in un segreto di Secrets Manager?](#) nella documentazione di Secrets Manager.

### sicurezza fin dalla progettazione

Un approccio di ingegneria dei sistemi che tiene conto della sicurezza durante l'intero processo di sviluppo.

## controllo di sicurezza

Un guardrail tecnico o amministrativo che impedisce, rileva o riduce la capacità di un autore di minacce di sfruttare una vulnerabilità di sicurezza. [Esistono quattro tipi principali di controlli di sicurezza: preventivi, investigativi, reattivi e proattivi.](#)

## rafforzamento della sicurezza

Il processo di riduzione della superficie di attacco per renderla più resistente agli attacchi. Può includere azioni come la rimozione di risorse che non sono più necessarie, l'implementazione di best practice di sicurezza che prevedono la concessione del privilegio minimo o la disattivazione di funzionalità non necessarie nei file di configurazione.

## sistema di gestione delle informazioni e degli eventi di sicurezza (SIEM)

Strumenti e servizi che combinano sistemi di gestione delle informazioni di sicurezza (SIM) e sistemi di gestione degli eventi di sicurezza (SEM). Un sistema SIEM raccoglie, monitora e analizza i dati da server, reti, dispositivi e altre fonti per rilevare minacce e violazioni della sicurezza e generare avvisi.

## automazione della risposta alla sicurezza

Un'azione predefinita e programmata progettata per rispondere o porre rimedio automaticamente a un evento di sicurezza. Queste automazioni fungono da controlli di sicurezza [investigativi](#) o [reattivi](#) che aiutano a implementare le migliori pratiche di sicurezza. AWS Esempi di azioni di risposta automatizzate includono la modifica di un gruppo di sicurezza VPC, l'applicazione di patch a un'istanza Amazon EC2 o la rotazione delle credenziali.

## Crittografia lato server

Crittografia dei dati a destinazione, da parte di chi li riceve. Servizio AWS

## Policy di controllo dei servizi (SCP)

Una policy che fornisce il controllo centralizzato sulle autorizzazioni per tutti gli account di un'organizzazione in AWS Organizations. Le SCP definiscono i guardrail o fissano i limiti alle azioni che un amministratore può delegare a utenti o ruoli. Puoi utilizzare le SCP come elenchi consentiti o elenchi di rifiuto, per specificare quali servizi o azioni sono consentiti o proibiti. Per ulteriori informazioni, consulta [le politiche di controllo del servizio](#) nella AWS Organizations documentazione.

## endpoint del servizio

L'URL del punto di ingresso per un Servizio AWS. Puoi utilizzare l'endpoint per connetterti a livello di programmazione al servizio di destinazione. Per ulteriori informazioni, consulta [Endpoint del Servizio AWS](#) nei Riferimenti generali di AWS.

## accordo sul livello di servizio (SLA)

Un accordo che chiarisce ciò che un team IT promette di offrire ai propri clienti, ad esempio l'operatività e le prestazioni del servizio.

## indicatore del livello di servizio (SLI)

Misurazione di un aspetto prestazionale di un servizio, ad esempio il tasso di errore, la disponibilità o la velocità effettiva.

## obiettivo a livello di servizio (SLO)

[Una metrica target che rappresenta lo stato di un servizio, misurato da un indicatore del livello di servizio.](#)

## Modello di responsabilità condivisa

Un modello che descrive la responsabilità condivisa AWS per la sicurezza e la conformità del cloud. AWS è responsabile della sicurezza del cloud, mentre tu sei responsabile della sicurezza nel cloud. Per ulteriori informazioni, consulta [Modello di responsabilità condivisa](#).

## Shadow AI

Applicazioni di [intelligenza artificiale](#) non autorizzate create o utilizzate al di fuori dei canali regolamentati all'interno di un'organizzazione.

## SIEM

Vedi il [sistema di gestione delle informazioni e degli eventi sulla sicurezza](#).

## punto di errore singolo (SPOF)

Un guasto in un singolo componente critico di un'applicazione che può disturbare il sistema.

## SLAM

Vedi il contratto sul [livello di servizio](#).

## SLI

Vedi l'indicatore del [livello di servizio](#).

## LENTA

Vedi obiettivo del [livello di servizio](#).

### modello split-and-seed

Un modello per dimensionare e accelerare i progetti di modernizzazione. Man mano che vengono definite nuove funzionalità e versioni dei prodotti, il team principale si divide per creare nuovi team di prodotto. Questo aiuta a dimensionare le capacità e i servizi dell'organizzazione, migliora la produttività degli sviluppatori e supporta una rapida innovazione. Per ulteriori informazioni, vedere [Approccio graduale alla modernizzazione delle applicazioni in](#). Cloud AWS

## SPOF

Vedi [punto di errore singolo](#).

### schema a stella

Una struttura organizzativa di database che utilizza un'unica tabella dei fatti di grandi dimensioni per archiviare i dati transazionali o misurati e utilizza una o più tabelle dimensionali più piccole per memorizzare gli attributi dei dati. Questa struttura è progettata per l'uso in un [data warehouse](#) o per scopi di business intelligence.

### modello del fico strangolatore

Un approccio alla modernizzazione dei sistemi monolitici mediante la riscrittura e la sostituzione incrementali delle funzionalità del sistema fino alla disattivazione del sistema legacy. Questo modello utilizza l'analogia di una pianta di fico che cresce fino a diventare un albero robusto e alla fine annienta e sostituisce il suo ospite. Il modello è stato [introdotto da Martin Fowler](#) come metodo per gestire il rischio durante la riscrittura di sistemi monolitici. Per un esempio di come applicare questo modello, consulta [Modernizzare i servizi Web Microsoft ASP.NET \(ASMX\) legacy in modo incrementale utilizzando contenitori e Amazon API Gateway](#).

### sottorete

Un intervallo di indirizzi IP nel VPC. Una sottorete deve risiedere in una singola zona di disponibilità.

### controllo di supervisione e acquisizione dati (SCADA)

Nella produzione, un sistema che utilizza hardware e software per monitorare gli asset fisici e le operazioni di produzione.

## crittografia simmetrica

Un algoritmo di crittografia che utilizza la stessa chiave per crittografare e decrittografare i dati.

## test sintetici

Test di un sistema in modo da simulare le interazioni degli utenti per rilevare potenziali problemi o monitorare le prestazioni. Puoi usare [Amazon CloudWatch Synthetics](#) per creare questi test.

## prompt di sistema

Una tecnica per fornire contesto, istruzioni o linee guida a un [LLM](#) per indirizzarne il comportamento. I prompt di sistema aiutano a impostare il contesto e stabilire regole per le interazioni con gli utenti.

# T

## tag

Key-value coppie che fungono da metadati per l'organizzazione delle risorse. AWS Con i tag è possibile a gestire, identificare, organizzare, cercare e filtrare le risorse. Per ulteriori informazioni, consulta [Tagging delle risorse AWS](#).

## variabile di destinazione

Il valore che stai cercando di prevedere nel machine learning supervisionato. Questo è indicato anche come variabile di risultato. Ad esempio, in un ambiente di produzione la variabile di destinazione potrebbe essere un difetto del prodotto.

## elenco di attività

Uno strumento che viene utilizzato per tenere traccia dei progressi tramite un runbook. Un elenco di attività contiene una panoramica del runbook e un elenco di attività generali da completare. Per ogni attività generale, include la quantità stimata di tempo richiesta, il proprietario e lo stato di avanzamento.

## ambiente di test

Vedi [ambiente](#).

## training

Fornire dati da cui trarre ispirazione dal modello di machine learning. I dati di training devono contenere la risposta corretta. L'algoritmo di apprendimento trova nei dati di addestramento i

pattern che mappano gli attributi dei dati di input al target (la risposta che si desidera prevedere). Produce un modello di ML che acquisisce questi modelli. Puoi quindi utilizzare il modello di ML per creare previsioni su nuovi dati di cui non si conosce il target.

strumento

Una funzione o API che un [agente](#) può richiamare per eseguire operazioni in sistemi esterni.

Transit Gateway

Un hub di transito di rete che è possibile utilizzare per collegare i VPC e le reti on-premise. Per ulteriori informazioni, consulta [Cos'è un gateway di transito](#) nella AWS Transit Gateway documentazione.

flusso di lavoro basato su trunk

Un approccio in cui gli sviluppatori creano e testano le funzionalità localmente in un ramo di funzionalità e quindi uniscono tali modifiche al ramo principale. Il ramo principale viene quindi integrato negli ambienti di sviluppo, produzione e produzione, in sequenza.

Accesso attendibile

Concessione delle autorizzazioni a un servizio specificato dall'utente per eseguire attività all'interno dell'organizzazione AWS Organizations e nei suoi account per conto dell'utente. Il servizio attendibile crea un ruolo collegato al servizio in ogni account, quando tale ruolo è necessario, per eseguire attività di gestione per conto dell'utente. Per ulteriori informazioni, consulta [Utilizzo AWS Organizations con altri AWS servizi](#) nella AWS Organizations documentazione.

regolazione

Modificare alcuni aspetti del processo di training per migliorare la precisione del modello di ML. Ad esempio, puoi addestrare il modello di ML generando un set di etichette, aggiungendo etichette e quindi ripetendo questi passaggi più volte con impostazioni diverse per ottimizzare il modello.

team da due pizze

Una piccola DevOps squadra che puoi sfamare con due pizze. Un team composto da due persone garantisce la migliore opportunità possibile di collaborazione nello sviluppo del software.

## U

### incertezza

Un concetto che si riferisce a informazioni imprecise, incomplete o sconosciute che possono minare l'affidabilità dei modelli di machine learning predittivi. Esistono due tipi di incertezza: l'incertezza epistemica, che è causata da dati limitati e incompleti, mentre l'incertezza aleatoria è causata dal rumore e dalla casualità insiti nei dati.

### compiti indifferenziati

Conosciuto anche come sollevamento di carichi pesanti, è un lavoro necessario per creare e far funzionare un'applicazione, ma che non apporta valore diretto all'utente finale né offre vantaggi competitivi. Esempi di attività indifferenziate includono l'approvvigionamento, la manutenzione e la pianificazione della capacità.

### ambienti superiori

[Vedi ambiente.](#)

## V

### vacuum

Un'operazione di manutenzione del database che prevede la pulizia dopo aggiornamenti incrementali per recuperare lo spazio di archiviazione e migliorare le prestazioni.

### controllo delle versioni

Processi e strumenti che tengono traccia delle modifiche, ad esempio le modifiche al codice di origine in un repository.

### Peering VPC

Una connessione tra due VPC che consente di instradare il traffico tramite indirizzi IP privati. Per ulteriori informazioni, consulta [Che cos'è il peering VPC?](#) nella documentazione di Amazon VPC.

### vulnerabilità

Un difetto software o hardware che compromette la sicurezza del sistema.

## W

### cache calda

Una cache del buffer che contiene dati correnti e pertinenti a cui si accede frequentemente. L'istanza di database può leggere dalla cache del buffer, il che richiede meno tempo rispetto alla lettura dalla memoria dal disco principale.

### dati caldi

Dati a cui si accede raramente. Quando si eseguono interrogazioni di questo tipo di dati, in genere sono accettabili interrogazioni moderatamente lente.

### funzione finestra

Una funzione SQL che esegue un calcolo su un gruppo di righe che si riferiscono in qualche modo al record corrente. Le funzioni della finestra sono utili per l'elaborazione di attività, come il calcolo di una media mobile o l'accesso al valore delle righe in base alla posizione relativa della riga corrente.

### Carico di lavoro

Una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione rivolta ai clienti o un processo back-end.

### flusso di lavoro

Gruppi funzionali in un progetto di migrazione responsabili di una serie specifica di attività. Ogni flusso di lavoro è indipendente ma supporta gli altri flussi di lavoro del progetto. Ad esempio, il flusso di lavoro del portfolio è responsabile della definizione delle priorità delle applicazioni, della pianificazione delle ondate e della raccolta dei metadati di migrazione. Il flusso di lavoro del portfolio fornisce queste risorse al flusso di lavoro di migrazione, che quindi migra i server e le applicazioni.

## VERME

Vedi [scrivere una volta, leggere molti](#).

## WQF

Vedi [AWS Workload Qualification Framework](#).

---

scrivi una volta, leggi molte (WORM)

Un modello di storage che scrive i dati una sola volta e ne impedisce l'eliminazione o la modifica. Gli utenti autorizzati possono leggere i dati tutte le volte che è necessario, ma non possono modificarli. Questa infrastruttura di archiviazione dei dati è considerata [immutabile](#).

## Z

exploit zero-day

[Un attacco, in genere malware, che sfrutta una vulnerabilità zero-day.](#)

vulnerabilità zero-day

Un difetto o una vulnerabilità assoluta in un sistema di produzione. Gli autori delle minacce possono utilizzare questo tipo di vulnerabilità per attaccare il sistema. Gli sviluppatori vengono spesso a conoscenza della vulnerabilità causata dall'attacco.

prompt zero-shot

Fornire a un [LLM](#) le istruzioni per eseguire un'attività ma non esempi (immagini) che possano aiutarla. Il LLM deve utilizzare le sue conoscenze pre-addestrate per gestire l'attività. L'efficacia del prompt zero-shot dipende dalla complessità dell'attività e dalla qualità del prompt. [Vedi anche few-shot prompting.](#)

applicazione zombie

Un'applicazione che prevede un utilizzo CPU e memoria inferiore al 5%. In un progetto di migrazione, è normale ritirare queste applicazioni.

---

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.