

# Principio dell'eccellenza operativa



# Principio dell'eccellenza operativa: Framework AWS Well-Architected

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

---

# Table of Contents

Riassunto e introduzione .....	1
Introduzione .....	1
Eccellenza operativa .....	3
Principi di progettazione .....	3
Definizione .....	5
Organizzazione .....	6
Priorità dell'organizzazione .....	6
OPS01-BP01 Valutazione delle esigenze dei clienti .....	6
OPS01-BP02 Valutazione delle esigenze dei clienti interni .....	8
OPS01-BP03 Valutazione dei requisiti di governance .....	9
OPS01-BP04 Valutazione dei requisiti di conformità .....	12
OPS01-BP05 Valutazione del panorama delle minacce .....	15
OPS01-BP06 Valutazione dei compromessi gestendo vantaggi e rischi .....	17
Modello operativo .....	20
Rappresentazioni del modello operativo 2x2 .....	21
Relazioni e proprietà .....	32
Cultura organizzativa .....	52
OPS03-BP01 Definizione della sponsorizzazione esecutiva .....	53
OPS03-BP02 Potere di intervento dei membri del team quando i risultati sono a rischio .....	56
OPS03-BP03 Incoraggiamento all'escalation .....	59
OPS03-BP04 Comunicazioni tempestive, chiare e fruibili .....	62
OPS03-BP05 Incoraggiamento alla sperimentazione .....	68
OPS03-BP06 Incoraggiamento ai membri del team a mantenere e ampliare le proprie competenze .....	71
OPS03-BP07 Fornitura di risorse appropriate ai team .....	74
Preparazione .....	77
Implementa l'osservabilità .....	77
OPS04-BP01 Identificazione degli indicatori chiave di prestazione .....	78
OPS04-BP02 Implementazione della telemetria dell'applicazione .....	80
OPS04-BP03 Implementazione della telemetria dell'esperienza utente .....	83
OPS04-BP04 Implementazione della telemetria delle dipendenze .....	86
OPS04-BP05 Implementazione del tracciamento distribuito .....	89
Progettazione delle operazioni .....	92
OPS05-BP01 Utilizzo del controllo delle versioni .....	92

OPS05-BP02 Test e convalida delle modifiche .....	94
OPS05-BP03 Utilizzo di sistemi di gestione delle configurazioni .....	97
OPS05-BP04 Utilizzo di sistemi di gestione della compilazione e implementazione .....	100
OPS05-BP05 Esecuzione della gestione delle patch .....	103
OPS05-BP06 Condivisione degli standard di progettazione .....	106
OPS05-BP07 Implementazione di prassi per migliorare la qualità del codice .....	109
OPS05-BP08 Utilizzo di più ambienti .....	112
OPS05-BP09 Applicazione di modifiche frequenti, minime e reversibili .....	113
OPS05-BP10 Automazione completa dell'integrazione e della distribuzione .....	114
Riduzione dei rischi di distribuzione .....	116
OPS06-BP01 Preparazione di un piano in caso di esito negativo delle modifiche .....	116
OPS06-BP02 Implementazioni dei test .....	119
OPS06-BP03 Utilizza strategie di deployment sicure .....	122
OPS06-BP04 Automazione dei test e del rollback .....	125
Prontezza operativa e gestione delle modifiche .....	129
OPS07-BP01 Verifica della capacità del personale .....	130
OPS07-BP02: Revisione costante della prontezza operativa .....	132
OPS07-BP03 Utilizzo di runbook per eseguire le procedure .....	136
OPS07-BP04 Utilizzo dei playbook per analizzare i problemi .....	140
OPS07-BP05 Adozione di decisioni informate per implementare sistemi e modifiche .....	144
OPS07-BP06 Abilitazione dei piani di supporto per i carichi di lavoro di produzione .....	146
Operatività .....	150
Utilizzare l'osservabilità del carico di lavoro .....	150
OPS08-BP01 Analisi delle metriche del carico di lavoro .....	151
OPS08-BP02 Analizza i log relativi ai carichi di lavoro .....	153
OPS08-BP03 Analisi delle tracce del carico di lavoro .....	156
OPS08-BP04 Creare avvisi fruibili .....	158
OPS08-BP05 Creare dashboard .....	161
Comprensione dello stato operativo .....	164
OPS09-BP01 Misura gli obiettivi operativi e i KPI con le metriche .....	165
OPS09-BP02 Comunicare lo stato e le tendenze per garantire la visibilità delle operazioni ..	167
OPS09-BP03 Revisione delle metriche operative e assegnazione delle priorità per favorire il miglioramento .....	169
Risposta agli eventi .....	171
OPS10-BP01 Utilizzo di un processo per la gestione di eventi, incidenti e problemi .....	172
OPS10-BP02 Definizione di un processo per ogni avviso .....	177

OPS10-BP03 Definizione della priorità degli eventi operativi in base agli effetti sul business .....	181
OPS10-BP04 Definizione dei percorsi di escalation .....	184
OPS10-BP05 Definizione di un piano di comunicazione con i clienti per eventi che incidono sul servizio .....	186
OPS10-BP06 Comunicazione dello stato tramite pannelli di controllo .....	189
OPS10-BP07 Automazione delle risposte agli eventi .....	192
Evoluzione .....	196
Imparare, condividere e migliorare .....	196
OPS11-BP01 Definizione di un processo per il miglioramento continuo .....	197
OPS11-BP02 Esecuzione di analisi post-incidente .....	199
OPS11-BP03 Implementazione di cicli di feedback .....	201
OPS11-BP04 Gestione delle informazioni .....	205
OPS11-BP05 Definizione dei fattori che promuovono il miglioramento .....	207
OPS11-BP06 Convalida delle informazioni .....	210
OPS11-BP07 Revisione dei parametri delle operazioni .....	211
OPS11-BP08 Documentazione e condivisione delle conoscenze acquisite .....	213
OPS11-BP09 Allocazione di tempo per i miglioramenti .....	215
Conclusione .....	217
Collaboratori .....	218
Approfondimenti .....	219
Revisioni del documento .....	220

# Principio dell'eccellenza operativa – Framework AWS Well-Architected

Data di pubblicazione: 27 giugno 2024 ([Revisioni del documento](#))

Questo whitepaper tratta del principio dell'eccellenza operativa del Framework AWS Well-Architected. Fornisce istruzioni per aiutarti ad applicare best practice per la progettazione, la distribuzione e la manutenzione dei carichi di lavoro AWS.

## Introduzione

Al [Framework AWS Well-Architected](#) aiuta a comprendere i pro e i contro delle decisioni prese durante la creazione dei carichi di lavoro in AWS. Utilizzando il Framework, scoprirai le best practice operative e architetturali per progettare e gestire carichi di lavoro affidabili, sicuri, efficienti, convenienti e sostenibili nel cloud. Il canone permette di misurare in modo coerente le operazioni e le architetture secondo le best practice e di identificare le aree da migliorare. Disporre di carichi di lavoro ben architettati, progettati tenendo conto delle operazioni, aumenta notevolmente la probabilità di successo aziendale.

Il Canone si basa su sei principi:

- Eccellenza operativa
- Sicurezza
- Affidabilità
- Efficienza delle prestazioni
- Ottimizzazione dei costi
- Sostenibilità

Il presente documento è incentrato sul principio dell'eccellenza operativa e sulla sua applicazione come base di partenza per lo sviluppo di soluzioni ben architettate. L'eccellenza operativa non è facile da raggiungere nei tradizionali ambienti in locale, in cui le operazioni sono considerate come una funzione isolata e distinta dalle linee di business e dai team aziendali e di sviluppo che supporta. Adottando le pratiche descritte in questo documento, puoi creare architetture che forniscano approfondimenti sul loro stato, siano attivate per rispondere in modo efficiente ed efficace ad eventi e operazioni e possano continuare a migliorare e supportare i tuoi obiettivi aziendali.

Questo documento è rivolto a chi ricopre ruoli nell'ambito della tecnologia, ad esempio ai Chief Technology Officer (CTO), ai progettisti, agli sviluppatori e ai membri dei team operativi. Dopo avere letto questo documento, comprenderai le best practice di AWS e le strategie da utilizzare durante la progettazione di architetture di un ambiente cloud caratterizzato dall'eccellenza operativa. Il presente documento non fornisce dettagli sull'implementazione o sui pattern architetturali. Tuttavia, include riferimenti alle risorse adeguate in cui trovare tali informazioni.

# Eccellenza operativa

In Amazon definiamo l'eccellenza operativa come l'impegno a sviluppare correttamente il software garantendo costantemente un'esperienza cliente di alto livello. Contiene best practice per organizzare il tuo team, progettare il tuo carico di lavoro, farlo funzionare su scala e seguire la sua evoluzione nel tempo. L'eccellenza operativa aiuta il tuo team a concentrare la maggior parte del tempo sullo sviluppo di nuove funzionalità a vantaggio del cliente e dedicare meno tempo alla manutenzione e alle operazioni di emergenza. Per sviluppare in modo corretto rispettiamo le best practice che ci consentono di avere sistemi eseguiti correttamente, un carico di lavoro bilanciato per te e per il tuo team e, aspetto fondamentale, un'ottima esperienza clienti.

L'obiettivo dell'eccellenza operativa consiste nell'offrire al cliente nuove funzionalità e correzioni di bug affidabili in tempi rapidi. Le organizzazioni che investono nell'eccellenza operativa deliziano continuamente i clienti, sviluppando nuove funzionalità, apportando modifiche e gestendo gli insuccessi. Lungo il percorso, l'eccellenza operativa porta verso l'integrazione continua e la distribuzione continua (CI/CD) aiutando gli sviluppatori a ottenere costantemente risultati di qualità elevata.

## Principi di progettazione

Di seguito sono riportati i principi di progettazione per l'eccellenza operativa nel cloud.

- Organizza i team sulla base dei risultati aziendali: la capacità di un team di conseguire i risultati aziendali deriva dalla visione della leadership, dalle operazioni efficaci e da un modello operativo allineato all'azienda. È necessario che la leadership sia totalmente coinvolta e impegnata nella trasformazione delle operazioni nel cloud (CloudOps) con un modello operativo cloud adeguato che incentivi i team a operare nel modo più efficiente per raggiungere i risultati aziendali. Il modello operativo corretto include persone, processi e capacità tecnologiche per scalare, ottimizzare la produttività e favorire la differenziazione tramite l'agilità, la reattività e l'adattamento. La visione a lungo termine dell'organizzazione si traduce in obiettivi che vengono comunicati alle parti interessate dell'azienda e agli utenti dei tuoi servizi cloud. Gli obiettivi e i KPI operativi sono allineati a tutti i livelli. Questa procedura promuove il valore a lungo termine derivante dall'implementazione dei seguenti principi di progettazione.
- Implementa l'osservabilità per ottenere informazioni utili: acquisisci informazioni dettagliate su comportamento, prestazioni, affidabilità, costi e stato del carico di lavoro. Stabilisci indicatori chiave delle prestazioni (KPI) e usa la telemetria dell'osservabilità per prendere decisioni informate e

agire tempestivamente quando i risultati aziendali sono a rischio. Migliora in modo proattivo le prestazioni, l'affidabilità e i costi sulla base di dati osservabili utilizzabili.

- Automatizza in sicurezza laddove possibile: nel cloud è possibile applicare a tutto il tuo ambiente la medesima disciplina di progettazione che utilizzi per il codice dell'applicazione. Definisci l'intero carico di lavoro e le relative operazioni (applicazioni, infrastruttura, configurazione e procedure) come codice e aggiornarlo. Quindi, automatizza le operazioni del carico di lavoro avviandole in risposta agli eventi. Nel cloud, utilizzi la sicurezza dell'automazione configurando i guardrail, tra cui il controllo della frequenza, le soglie di errore e le approvazioni. Un'automazione efficiente offre risposte coerenti agli eventi, limita l'errore umano e riduce l'impegno degli operatori.
- Apporta modifiche frequenti, piccole e reversibili: progetta carichi di lavoro scalabili e con accoppiamento debole per l'aggiornamento regolare dei componenti. Le tecniche di implementazione automatizzate insieme a modifiche incrementali più piccole riducono il raggio di esplosione, ovvero l'entità dell'impatto, e consentono un'inversione più rapida in caso di guasti. Ciò aumenta la fiducia necessaria per apportare modifiche strategiche al carico di lavoro mantenendo la qualità e adattandosi rapidamente ai cambiamenti delle condizioni di mercato.
- Perfeziona frequentemente le procedure operative: man mano che potenzi i carichi di lavoro, perfeziona le operazioni in modo appropriato. Se usi procedure operative, cerca delle opportunità per migliorarle. Organizza regolari revisioni per accertarti che tutte le procedure siano efficaci e che i team le conoscano adeguatamente. Se vengono individuate delle lacune, aggiorna le procedure di conseguenza. Comunica gli aggiornamenti procedurali a tutte le parti interessate e ai team. Converti le operazioni in gioco per condividere le best practice e fornire occasioni di formazione ai team.
- Prevedi gli insuccessi: massimizza il successo operativo creando scenari di errore per comprendere il profilo di rischio del carico di lavoro e l'impatto sui risultati aziendali. Testa l'efficacia delle procedure e la risposta del team a questi errori simulati. Prendi decisioni informate per gestire i rischi aperti identificati tramite i test.
- Impara da metriche ed eventi operativi: favorisci il miglioramento tramite le lezioni apprese da tutti gli eventi e gli errori operativi. Condividi ciò che hai imparato con i vari team e con tutta l'organizzazione. Gli insegnamenti evidenziano dati e aneddoti su come le operazioni contribuiscono al conseguimento dei risultati aziendali.
- Usa servizi gestiti: riduci il carico operativo utilizzando servizi gestiti da AWS, laddove possibile. Sviluppa procedure operative basate sulle interazioni con tali servizi.

# Definizione

Esistono quattro aree di best practice per l'eccellenza operativa nel cloud:

- Organizzazione
- Preparazione
- Operatività
- Evoluzione

La leadership dell'organizzazione definisce gli obiettivi aziendali. La tua organizzazione deve comprendere i requisiti e le priorità e utilizzarli per organizzare e condurre attività a supporto del raggiungimento dei risultati aziendali. Il carico di lavoro deve generare le informazioni necessarie per supportarlo. L'implementazione di servizi per attivare l'integrazione, l'implementazione e la consegna del carico di lavoro darà vita a un flusso maggiore di modifiche vantaggiose in fase di produzione attraverso l'automazione dei processi ripetitivi.

Potrebbero esserci rischi inerenti al funzionamento del carico di lavoro. Devi comprendere questi rischi e prendere una decisione consapevole prima di passare alla fase di produzione. I team devono essere in grado di supportare il carico di lavoro. Le metriche aziendali e operative derivate dai risultati aziendali desiderati ti aiuteranno a comprendere lo stato del carico di lavoro e le attività operative e di rispondere agli incidenti. Le priorità cambieranno di pari passo con l'evoluzione delle esigenze aziendali e dell'ambiente aziendale. Utilizza questi aspetti come ciclo di feedback per apportare continui miglioramenti all'organizzazione e alle operazioni legate al carico di lavoro.

# Organizzazione

È necessario comprendere le priorità dell'organizzazione, la struttura organizzativa e il modo in cui l'organizzazione supporta i membri del team affinché essi possano supportare i risultati aziendali.

Per essere in grado di raggiungere l'eccellenza operativa, è necessario comprendere i seguenti aspetti:

## Argomenti

- [Priorità dell'organizzazione](#)
- [Modello operativo](#)
- [Cultura organizzativa](#)

## Priorità dell'organizzazione

È necessario che i team abbiano una comprensione condivisa dell'intero carico di lavoro, del ruolo che vi svolgono, nonché degli obiettivi aziendali condivisi. In questo modo potranno stabilire le priorità che possono favorire il successo aziendale. Un'adeguata definizione delle priorità massimizzerà i risultati dei tuoi sforzi. Rivedi regolarmente le tue priorità in modo che possano essere aggiornate al mutare delle esigenze della tua organizzazione.

## Best practice

- [OPS01-BP01 Valutazione delle esigenze dei clienti](#)
- [OPS01-BP02 Valutazione delle esigenze dei clienti interni](#)
- [OPS01-BP03 Valutazione dei requisiti di governance](#)
- [OPS01-BP04 Valutazione dei requisiti di conformità](#)
- [OPS01-BP05 Valutazione del panorama delle minacce](#)
- [OPS01-BP06 Valutazione dei compromessi gestendo vantaggi e rischi](#)

## OPS01-BP01 Valutazione delle esigenze dei clienti

Coinvolgi le principali parti interessate, compresi i team aziendali, di sviluppo e operativi, per determinare dove concentrare gli sforzi in base alle esigenze dei clienti esterni. Avrai così una

conoscenza approfondita del supporto operativo necessario per raggiungere i risultati aziendali desiderati.

Risultato desiderato:

- Lavori a ritroso partendo dalle esigenze dei clienti.
- Comprendi in che modo le procedure operative supportano i risultati e gli obiettivi aziendali.
- Coinvolgi tutte le parti interessate.
- Disponi di meccanismi per soddisfare le esigenze dei clienti.

Anti-pattern comuni:

- Hai deciso di non fornire il servizio clienti al di fuori dell'orario lavorativo di base, ma non hai esaminato i dati cronologici riguardanti le richieste di supporto. Non sai se questo determinerà un impatto sui tuoi clienti.
- Stai sviluppando una nuova funzionalità, ma non hai coinvolto i clienti per capire se è desiderata ed eventualmente in quale forma; inoltre non hai condotto attività di sperimentazione per convalidarne la necessità e il metodo di distribuzione.

Vantaggi dell'adozione di questa best practice: i clienti le cui esigenze sono soddisfatte hanno maggiori probabilità di rimanere tali. Valutando e comprendendo le esigenze dei clienti esterni sarà possibile organizzare le attività in base alle priorità e offrire valore aggiunto.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

Comprendi le esigenze aziendali: il successo dell'azienda nasce dalla condivisione di obiettivi e conoscenze tra le parti interessate, compresi i team aziendali, di sviluppo e operativi.

Esamina gli obiettivi aziendali, le esigenze e le priorità dei clienti esterni: coinvolgi le principali parti interessate, compresi i team aziendali, di sviluppo e operativi, per discutere obiettivi, esigenze e priorità dei clienti esterni. Otterrai così una conoscenza approfondita del supporto operativo necessario per raggiungere i risultati aziendali e del cliente.

Condividi le conoscenze: condividi le conoscenze sulle funzioni aziendali del carico di lavoro, sui ruoli di ciascuno dei team nel gestire il carico di lavoro e su come questi supportino gli obiettivi aziendali condivisi tra clienti interni ed esterni.

## Risorse

Best practice correlate:

- [OPS11-BP03 Implementazione di cicli di feedback](#)

## OPS01-BP02 Valutazione delle esigenze dei clienti interni

Coinvolgi i principali stakeholder, compresi i team aziendali, di sviluppo e operativi, nel determinare dove concentrare le attività in base alle esigenze dei clienti interni. Questo ti garantirà una conoscenza approfondita del supporto operativo necessario per raggiungere i risultati aziendali.

Risultato desiderato:

- Utilizzi le priorità definite per concentrare le iniziative di miglioramento delle operazioni laddove avranno il maggiore impatto (ad esempio, sviluppare le competenze dei team, migliorare le prestazioni del carico di lavoro, ridurre i costi, automatizzare i runbook o potenziare il monitoraggio).
- Aggiorni le priorità al mutare delle esigenze.

Anti-pattern comuni:

- Per semplificare la gestione della rete hai deciso di modificare l'assegnazione degli indirizzi IP per i team di prodotto senza consultarli. Non conosci l'impatto che questo avrà sui tuoi team di prodotto.
- Stai implementando un nuovo strumento di sviluppo, ma non hai coinvolto i clienti interni per scoprire se è necessario o se è compatibile con le loro pratiche esistenti.
- Stai implementando un nuovo sistema di monitoraggio, ma non hai contattato i clienti interni per scoprire se hanno esigenze di monitoraggio o reporting da tenere in considerazione.

Vantaggi dell'adozione di questa best practice: la valutazione e la comprensione delle esigenze dei clienti interni consente di stabilire le priorità delle attività per offrire valore aziendale.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

- Comprendi le esigenze aziendali: il successo dell'azienda nasce dalla condivisione di obiettivi e conoscenze tra le parti interessate, compresi i team aziendali, di sviluppo e operativi.

- Esamina gli obiettivi aziendali, le esigenze e le priorità dei clienti interni: coinvolgi le principali parti interessate, compresi i team aziendali, di sviluppo e operativi, per discutere obiettivi, esigenze e priorità dei clienti interni. In tal modo otterrai una conoscenza approfondita del supporto operativo necessario per raggiungere i risultati aziendali e del cliente.
- Stabilisci una comprensione condivisa: stabilisci una comprensione condivisa delle funzioni aziendali del carico di lavoro, dei ruoli di ciascuno dei team nel gestire il carico di lavoro e di come questi supportino gli obiettivi aziendali condivisi tra clienti interni ed esterni.

## Risorse

Best practice correlate:

- [OPS11-BP03 Implementazione di cicli di feedback](#)

## OPS01-BP03 Valutazione dei requisiti di governance

Con governance si intende l'insieme di policy, regole o framework che un'azienda usa per raggiungere i propri obiettivi. I requisiti di governance vengono generati all'intero dell'organizzazione. Possono influire sui tipi di tecnologia che scegli o sul modo in cui sviluppi il tuo carico di lavoro. Integra i requisiti di governance della tua organizzazione nel tuo carico di lavoro. La conformità è la capacità di dimostrare che hai implementato i requisiti di governance.

Risultato desiderato:

- I requisiti di governance sono integrati nel progetto architetturale e nell'operatività del tuo carico di lavoro.
- Puoi dimostrare di aver seguito i requisiti di governance.
- I requisiti di governance vengono rivisti e aggiornati con regolarità.

Anti-pattern comuni:

- La tua azienda richiede che l'account root abbia l'autenticazione multi-fattore. Non sei riuscito a implementare questo requisito e l'account root è compromesso.
- Durante la progettazione del carico di lavoro hai scelto un tipo di istanza non approvata dal dipartimento IT. Non riesci ad avviare il tuo carico di lavoro e devi procedere a una nuova progettazione.

- Devi avere un piano di ripristino di emergenza. Non ne hai uno e il tuo carico di lavoro è vittima di un'interruzione prolungata.
- Il tuo team vuole usare nuove istanze, ma i requisiti di governance non sono stati aggiornati e pertanto non sono consentite.

Vantaggi dell'adozione di questa best practice:

- Rispettare i requisiti di governance permette di allineare il carico di lavoro a policy organizzative di più ampio respiro.
- I requisiti di governance si basano su standard e best practice di settore per la tua organizzazione.

Livello di rischio associato se questa best practice non fosse adottata: Elevato

## Guida all'implementazione

Identifica il requisito di governance collaborando con le parti interessate e le organizzazioni preposte. Includi i requisiti di governance nel tuo carico di lavoro. Dimostra di aver seguito i requisiti di governance.

### Esempio del cliente

In AnyCompany Retail, il team operativo nell'ambiente cloud collabora con le parti interessate dell'organizzazione per sviluppare i requisiti di governance. Ad esempio, proibiscono l'accesso SSH alle istanze Amazon EC2. Se i team hanno necessità di accedere ai sistemi, devono usare AWS Systems Manager Session Manager. Il team operativo nell'ambiente cloud aggiorna con regolarità i requisiti di governance nel momento in cui vengono rilasciati nuovi servizi.

### Passaggi dell'implementazione

1. Identifica le parti interessate per il tuo carico di lavoro, inclusi eventuali team centralizzati.
2. Collabora con le parti interessate per identificare i requisiti di governance.
3. Dopo aver generato un elenco, dai la priorità alle voci relative a migliorie e inizia a implementarle nel tuo carico di lavoro.
  - a. Usa servizi come [AWS Config](#) per creare governance-as-code e verificare che tali requisiti di governance siano rispettati.
  - b. Se usi [AWS Organizations](#), puoi avvalerti di Policy di controllo dei servizi per implementare requisiti di governance.

#### 4. Fornisci la documentazione che convalida l'implementazione.

Livello di impegno per il piano di implementazione: Medio. L'implementazione di requisiti di governance mancanti può causare la rielaborazione del tuo carico di lavoro.

### Risorse

Best practice correlate:

- [OPS01-BP04 Valutazione dei requisiti di conformità](#) - La conformità è come la governance, ma è esterna all'organizzazione.

Documenti correlati:

- [Guida AWS sull'ambiente cloud di governance e gestione](#)
- [Best practice per le policy di controllo dei servizi di AWS Organizations in un ambiente multi-account](#)
- [Governance nel Cloud AWS: il giusto equilibrio tra agilità e sicurezza](#)
- [Cosa si intende per Governance, Rischio e Conformità \(GRC\)?](#)

Video correlati:

- [Gestione e governance AWS: configurazione, conformità e revisione - AWS Online Tech Talks](#)
- [AWS re:Inforce 2019: Governance per l'età del cloud \(DEM12-R1\)](#)
- [AWS re:Invent 2020: Ottieni compliance as code con AWS Config](#)
- [AWS re:Invent 2020: Governance agile su AWS GovCloud \(US\)](#)

Esempi correlati:

- [Esempi di pacchetti di conformità AWS Config](#)

Servizi correlati:

- [AWS Config](#)
- [AWS Organizations - Policy di controllo dei servizi](#)

## OPS01-BP04 Valutazione dei requisiti di conformità

I requisiti di conformità interna, di settore e normativa sono un fattore importante per la definizione delle priorità della tua organizzazione. L'assetto di conformità della tua azienda potrebbe impedirti di usare tecnologie specifiche o posizioni geografiche. Applica la due diligence in assenza di contesti di conformità esterni. Genera revisioni o report per convalidare la conformità.

Se comunichi all'esterno che il tuo prodotto è in linea con standard specifici di conformità, devi avere un processo interno in grado di garantire costantemente che tale affermazione sia vera. Gli esempi di standard di conformità includono PCI DSS, FedRAMP e HIPAA. Gli standard di conformità applicabili vengono stabiliti in base a diversi fattori, come il tipo di dati che la soluzione archivia o trasmette e quali aree geografiche sono supportate dalla soluzione.

Risultato desiderato:

- Requisiti di conformità interni, di settore e normativi sono parte integrante della selezione dell'architettura.
- Puoi verificare la conformità e generare report di audit.

Anti-pattern comuni:

- Parti del tuo carico di lavoro rientrano nel framework Payment Card Industry Data Security Standard (PCI-DSS), ma il tuo carico di lavoro archivia dati di carte di credito non crittografati.
- Gli architetti e gli sviluppatori software non conoscono il contesto di conformità che la tua organizzazione è tenuta a rispettare.
- L'audit annuale Systems and Organizations Control (SOC2) Type II avrà luogo a breve e tu non sei in grado di verificare la presenza dei controlli richiesti.

Vantaggi dell'adozione di questa best practice:

- Valutando e comprendendo i requisiti di conformità che si applicano al carico di lavoro sarà possibile organizzare le attività in base a priorità e offrire valore aggiunto.
- Scegli le sedi e le tecnologie corrette, in linea con il tuo contesto di integrità.
- La progettazione del tuo carico di lavoro ai fini dell'auditing ti consente di dimostrare la tua aderenza al modello di conformità.

Livello di rischio associato se questa best practice non fosse adottata: Elevato

## Guida all'implementazione

Implementare questa best practice significa integrare requisiti di conformità nel processo di progettazione dell'architettura. I membri del tuo team sono a conoscenza del contesto di conformità richiesto. Convalida la conformità in linea con il contesto.

### Esempio del cliente

AnyCompany Retail archivia informazioni sulle carte di credito per i clienti. Gli sviluppatori del team di archiviazione delle carte sono al corrente della necessità di rispettare la conformità agli standard PCI-DSS. Hanno adottato misure per verificare che le informazioni sulle carte di credito siano archiviate e consultabili in totale sicurezza, in linea con quanto stabilito dagli standard PCI-DSS: Ogni anno collaborano con il team di sicurezza per confermare la conformità.

### Passaggi dell'implementazione

1. Collabora con i team di sicurezza e governance per stabilire quali conformità interne, normative o di settore deve rispettare il tuo carico di lavoro. Integra gli standard di conformità nel tuo carico di lavoro.
  - a. Convalida continuamente la conformità delle risorse AWS con servizi come [AWS Compute Optimizer](#) e [AWS Security Hub](#).
2. Comunica ai membri del tuo team i requisiti di conformità, in modo che possano gestire e far evolvere il carico di lavoro in linea con tali requisiti. I requisiti di conformità devono essere inclusi nelle scelte tecnologiche e architetturali.
3. A seconda del contesto di conformità, potresti dover generare un report di audit o conformità. Collabora con la tua organizzazione per automatizzare il più possibile questo processo.
  - a. Usa servizi come [AWS Audit Manager](#) per convalidare la conformità e generare report di audit.
  - b. Puoi scaricare documenti su conformità e sicurezza AWS con [AWS Artifact](#).

Livello di impegno per il piano di implementazione: Medio. Implementare i requisiti di conformità può essere complesso. Generare report di audit o documenti di conformità aggiunge altre complessità.

## Risorse

Best practice correlate:

- [SEC01-BP03 Identificazione e convalida degli obiettivi di controllo](#) - Gli obiettivi di controllo della sicurezza sono una parte importante della conformità nel suo complesso.

- [SEC01-BP06 Automatizzazione dei test e della convalida dei controlli di sicurezza nelle pipeline](#)  
- Come parte delle tue pipeline, convalida i controlli di sicurezza. Puoi anche generare la documentazione di conformità per le nuove modifiche.
- [SEC07-BP02 Definizione dei controlli di protezione dei dati](#) - Molti standard di conformità si basano su policy di gestione e di archiviazione dei dati.
- [SEC10-BP03 Preparazione di funzionalità forensi](#) - Le funzionalità forensi possono a volte essere usate nella conformità di auditing.

#### Documenti correlati:

- [AWS Compliance Center](#)
- [Risorse di conformità AWS](#)
- Whitepaper [Risk and Compliance AWS](#):
- [Modello di responsabilità condivisa AWS](#)
- [Servizi AWS inerenti secondo i programmi di conformità](#)

#### Video correlati:

- [AWS re:Invent 2020: Ottieni compliance as code con AWS Compute Optimizer](#)
- [AWS re:Invent 2021 - Conformità, garanzia e auditing del cloud](#)
- [AWS Summit ATL 2022 - Implementare conformità, garanzia e auditing su AWS \(COP202\)](#)

#### Esempi correlati:

- [PCI DSS e best practice per la sicurezza di base di AWS su AWS](#)

#### Servizi correlati:

- [AWS Artifact](#)
- [AWS Audit Manager](#)
- [AWS Compute Optimizer](#)
- [AWS Security Hub](#)

## OPS01-BP05 Valutazione del panorama delle minacce

Valuta le minacce per l'azienda (ad esempio, concorrenza, rischi e responsabilità aziendali, rischi operativi e minacce per la sicurezza delle informazioni) e conserva le informazioni aggiornate in un registro dei rischi. Quando stabilisci dove concentrare gli sforzi, tieni in considerazione l'impatto dei rischi.

Il [Framework Well-Architected](#) favorisce l'apprendimento, la misurazione e il miglioramento. Fornisce una strategia coerente per la valutazione delle architetture e l'implementazione di progetti in grado di scalare nel corso del tempo. AWS mette a disposizione lo [AWS Well-Architected Tool](#) per aiutarti ad analizzare l'approccio prima dello sviluppo e lo stato dei carichi di lavoro prima e durante la fase di produzione. Puoi confrontare il tuo approccio con le best practice architetture AWS più recenti, monitorare lo stato complessivo dei carichi di lavoro e ottenere approfondimenti sui potenziali rischi.

I clienti AWS possono usufruire della revisione Well-Architected dei carichi di lavoro mission-critical per [misurare le loro architetture](#) rispetto alle best practice AWS. I clienti del supporto Enterprise possono utilizzare una [revisione delle operazioni](#) ideata per agevolare l'identificazione delle lacune nell'approccio che usano nel cloud.

Il coinvolgimento trasversale dei team per tali controlli aiuta a comprendere a livello comune i carichi di lavoro e come i ruoli del team contribuiscano al successo. Le esigenze identificate nel corso dell'analisi possono aiutarti a definire le priorità.

[AWS Trusted Advisor](#) è uno strumento che fornisce l'accesso a una serie di controlli di base che propongono ottimizzazioni utili per la definizione delle priorità. [I clienti del supporto Business ed Enterprise](#) hanno accesso a ulteriori controlli a livello di sicurezza, affidabilità, prestazioni e ottimizzazione dei costi che possono essere utili per definire le priorità.

Risultato desiderato:

- Esamini e intervieni periodicamente sui risultati forniti da Well-Architected e Trusted Advisor.
- Sei a conoscenza dello stato delle patch più recenti dei servizi.
- Comprendi il rischio e l'impatto delle minacce note e intervieni di conseguenza.
- Implementi le mitigazioni necessarie.
- Comunichi azioni e contesto.

Anti-pattern comuni:

- Stai utilizzando la versione precedente di una libreria software nel tuo prodotto. Non sei a conoscenza di aggiornamenti di sicurezza alla libreria per problemi che potrebbero avere un impatto imprevisto sul carico di lavoro.
- Il tuo concorrente ha appena rilasciato una versione del proprio prodotto che risolve i reclami di molti dei tuoi clienti relativi al tuo prodotto. Non hai dato priorità alla risoluzione di questi problemi noti.
- Le autorità di regolamentazione hanno perseguito aziende come la tua che non sono conformi ai requisiti di conformità alla normativa legale. Non hai dato priorità ai requisiti di conformità in sospeso.

Vantaggi dell'adozione di questa best practice: identificando e comprendendo le minacce rivolte all'organizzazione e al carico di lavoro puoi determinare quali problematiche affrontare, la loro priorità e le risorse necessarie per farlo.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

- Valuta il panorama delle minacce: valuta le minacce per l'azienda, ad esempio concorrenza, rischi e responsabilità aziendali, rischi operativi e minacce alla sicurezza delle informazioni, in modo da poterne includere l'impatto quando determini dove concentrare le attività.
  - [Ultimi bollettini sulla sicurezza AWS](#)
  - [AWS Trusted Advisor](#)
- Mantieni un modello delle minacce: definisci e mantieni un modello delle minacce che identifichi le potenziali minacce, le mitigazioni pianificate e predisposte nonché la loro priorità. Esamina la probabilità che le minacce si manifestino come incidenti, il costo del recupero dagli incidenti, il danno previsto causato e il costo per prevenire tali incidenti. Modifica le priorità man mano che i contenuti del modello di minaccia cambiano.

## Risorse

Best practice correlate:

- [SEC01-BP07 Identificare le minacce e dare priorità alle mitigazioni utilizzando un modello di minaccia](#)

## Documenti correlati:

- [Conformità di Cloud AWS](#)
- [Ultimi bollettini sulla sicurezza AWS](#)
- [AWS Trusted Advisor](#)

## Video correlati:

- [AWS re:Inforce 2023 - A tool to help improve your threat modeling](#)

## OPS01-BP06 Valutazione dei compromessi gestendo vantaggi e rischi

Gli interessi contrastanti di più parti possono rendere difficile l'assegnazione delle priorità agli impegni, lo sviluppo delle capacità e il conseguimento di risultati in linea con le strategie aziendali. Ad esempio, è possibile che ti venga chiesto di accelerare la commercializzazione di nuove funzionalità piuttosto che ottimizzare i costi dell'infrastruttura IT. Questa richiesta può mettere due parti interessate in conflitto tra loro. In queste situazioni, le decisioni devono essere prese da un'autorità superiore che risolve il conflitto. I dati sono necessari per rimuovere l'attaccamento emotivo dal processo decisionale.

La stessa sfida può verificarsi a livello strategico. Ad esempio, la scelta tra l'utilizzo di tecnologie di database relazionali o non relazionali può avere un impatto significativo sul funzionamento di un'applicazione. È fondamentale comprendere i risultati prevedibili delle varie scelte.

AWS può aiutarti a istruire i team su AWS e i suoi servizi, affinché comprendano meglio in che modo le loro scelte possono influire sul carico di lavoro. Per istruire i team, utilizza le risorse fornite da [AWS Support](#) ([Centro conoscenze AWS](#), [AWS Discussion Forums](#) e [AWS Support Center](#)) e la [documentazione AWS](#). Per ulteriori domande, contatta AWS Support.

AWS condivide inoltre best practice e modelli operativi nella [Amazon Builders' Library](#). Un'ampia gamma di altre informazioni utili è disponibile tramite il [blog AWS](#) e il [podcast ufficiale di AWS](#).

Risultato desiderato: disponi di un framework di governance del processo decisionale chiaramente definito per facilitare le decisioni importanti a tutti i livelli dell'organizzazione di distribuzione del cloud. Questo framework include funzionalità come un registro dei rischi, i ruoli definiti autorizzati a prendere decisioni e i modelli prestabiliti per ogni livello di decisione che può essere presa. Il framework definisce in anticipo come vengono risolti i conflitti, quali dati devono essere presentati e come

viene stabilita la priorità delle opzioni, in modo che una volta prese le decisioni sia subito possibile lavorare per applicarle. Il framework del processo decisionale include un approccio standardizzato alla revisione e alla valutazione dei vantaggi e dei rischi di ogni decisione per comprenderne i compromessi. Possono essere inclusi fattori esterni, come l'aderenza ai requisiti di conformità normativa.

Anti-pattern comuni:

- Gli investitori richiedono di dimostrare la conformità agli standard PCI DSS (Payment Card Industry Data Security Standard). Non prendi in considerazione i compromessi tra soddisfare la loro richiesta e continuare con le attività di sviluppo già in corso. Al contrario, prosegui con il lavoro di sviluppo senza dimostrare la conformità. Gli investitori interrompono il supporto all'azienda per i dubbi relativi alla sicurezza della piattaforma e dei loro investimenti.
- Hai deciso di includere una libreria che uno dei tuoi sviluppatori ha trovato su Internet. Non hai valutato i rischi derivanti dall'adozione di questa libreria da un'origine sconosciuta e non sai se contiene vulnerabilità o codice dannoso.
- La giustificazione aziendale originale per la migrazione si basava sulla modernizzazione del 60% dei carichi di lavoro delle applicazioni. Tuttavia, a causa di difficoltà tecniche, è stata presa la decisione di modernizzare solo il 20%, con una riduzione dei vantaggi pianificati a lungo termine, un maggiore impegno operativo dei team dell'infrastruttura per supportare manualmente i sistemi legacy e un accresciuto affidamento sullo sviluppo di nuove competenze nei team dell'infrastruttura che non avevano pianificato questo cambiamento.

Vantaggi dell'adozione di questa best practice: ottieni un totale allineamento e il supporto per le priorità aziendali a livello dirigenziale, comprendi i rischi per raggiungere il risultato, prendi decisioni informate e agisci in modo appropriato quando i rischi ostacolano le opportunità di successo. Comprendere le implicazioni e le conseguenze delle tue decisioni ti aiuta a stabilire le priorità delle opzioni e a ottenere l'accordo dei leader più rapidamente, fornendo risultati aziendali migliori. Identifici i vantaggi derivanti dalle tue scelte e riconosci i rischi per l'organizzazione in modo da prendere decisioni basate sui dati, piuttosto che basate su aneddoti.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

La gestione dei vantaggi e dei rischi deve essere definita da un organo direttivo che stabilisca i requisiti del processo decisionale chiave. Le decisioni devono essere prese e assegnate alle priorità

in base ai vantaggi che ne derivano per l'organizzazione, con una comprensione dei rischi connessi. L'accuratezza delle informazioni è fondamentale quando si prendono le decisioni organizzative. Deve basarsi su misurazioni solide ed essere definita secondo le procedure comuni del settore per l'analisi costi-benefici. Per prendere questo tipo di decisioni, occorre trovare un equilibrio tra l'autorità centralizzata e quella decentralizzata. C'è sempre un compromesso ed è importante capire in che modo ogni scelta influisce sulle strategie definite e sui risultati aziendali desiderati.

## Passaggi dell'implementazione

1. Formalizza le procedure di misurazione dei vantaggi in un framework olistico di governance del cloud.
  - a. Bilancia il controllo centrale del processo decisionale con l'autorità decentralizzata per alcune decisioni.
  - b. Comprendi che i gravosi processi decisionali imposti per ogni decisione possono rallentare le operazioni.
  - c. Incorpora nel processo decisionale i fattori esterni, come i requisiti di conformità.
2. Stabilisci un framework del processo decisionale concordato per vari livelli di decisioni, che includa chi è tenuto a prendere le decisioni soggette a conflitti di interessi.
  - a. Centralizza le decisioni definitive che potrebbero essere irreversibili.
  - b. Consenti ai leader dell'organizzazione di livello inferiore di prendere decisioni reversibili.
3. Comprendi e gestisci i vantaggi e i rischi. Bilancia i vantaggi delle decisioni rispetto ai rischi connessi.
  - a. Identifica i vantaggi: identifica i vantaggi in base agli obiettivi aziendali, alle esigenze e alle priorità, ad esempio l'impatto del caso aziendale, il time-to-market, la sicurezza, l'affidabilità, le prestazioni e i costi.
  - b. Identifica i rischi: identifica i rischi in base agli obiettivi aziendali, alle esigenze e alle priorità, ad esempio il time-to-market, la sicurezza, l'affidabilità, le prestazioni e il costo.
  - c. Valuta i vantaggi rispetto ai rischi e prendi decisioni informate: determina l'impatto dei vantaggi e dei rischi in base agli obiettivi, alle esigenze e alle priorità delle principali parti interessate, inclusi business, sviluppo e operazioni. Valuta il valore del vantaggio rispetto alla probabilità di realizzazione del rischio e al costo del suo impatto. Ad esempio, enfatizzare la velocità di accesso al mercato rispetto all'affidabilità potrebbe offrire un vantaggio competitivo. Tuttavia, potrebbe causare tempi di attività ridotti in presenza di problemi di affidabilità.
4. Applica in modo programmatico le decisioni chiave che automatizzano l'aderenza ai requisiti di conformità.

5. Impiega le funzionalità e i framework noti del settore, come Value Stream Analysis e LEAN, per definire la linea di base per le prestazioni dello stato attuale, le metriche aziendali e le iterazioni dei progressi verso il miglioramento di queste metriche.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS01-BP05 Valutazione del panorama delle minacce](#)

Documenti correlati:

- [Elementi della cultura del Giorno 1 di Amazon | Prendi decisioni rapide e di qualità](#)
- [Governance del cloud](#)
- [Management and Governance Cloud Environment](#)
- [Governance in the Cloud and in the Digital Age: Parts One and Two](#)

Video correlati:

- [Podcast | Jeff Bezos | On how to make decisions](#)

Esempi correlati:

- [Make informed decisions using data \(The DevOps Sagas\)](#)
- [Using development value stream mapping to identify constraints to DevOps outcomes](#)

## Modello operativo

I tuoi team devono comprendere quale contributo offrono nel raggiungimento dei risultati aziendali. I team devono avere obiettivi condivisi e devono comprendere il proprio ruolo nel successo degli altri team. Comprendere la responsabilità, la proprietà, il modo in cui vengono prese le decisioni e chi ha l'autorità decisionale aiuterà a concentrare gli sforzi e a ottimizzare i contributi dei team.

Le esigenze di un team sono influenzate dal settore, dall'organizzazione, dalla composizione del team e dalle caratteristiche del carico di lavoro. Non è ragionevole aspettarsi che un singolo modello operativo sia in grado di supportare tutti i team e i relativi carichi di lavoro.

È probabile che il numero di modelli operativi presenti in un'organizzazione aumenti di pari passo con il numero dei team di sviluppo. Potrebbe essere necessario utilizzare una combinazione di diversi modelli operativi.

L'adozione di standard e l'utilizzo di servizi può semplificare le operazioni e limitare l'onere del supporto del modello operativo. Il vantaggio dato dallo sviluppo di standard condivisi è amplificato dal numero di team che adottano tali standard e che adotteranno nuove funzionalità.

È fondamentale che esistano meccanismi per richiedere aggiunte, modifiche ed eccezioni agli standard a supporto delle attività dei team. Senza questa opzione, gli standard diventano un ostacolo per l'innovazione. Le richieste devono essere approvate quando vengono ritenute fattibili e appropriate dopo una valutazione dei vantaggi e dei rischi.

Un set di responsabilità ben definito ridurrà la frequenza degli sforzi conflittuali e ridondanti. È più facile conseguire i risultati aziendali in presenza di un forte allineamento e di solide relazioni tra i team aziendali, di sviluppo e operativi.

## Rappresentazioni del modello operativo 2x2

Queste rappresentazioni del modello operativo 2x2 sono illustrazioni che ti aiutano a comprendere le relazioni tra i team nel tuo ambiente. Questi diagrammi si concentrano sulle competenze e sulle relazioni tra i team, ma discuteremo anche di governance e processi decisionali nel contesto di questi esempi.

I nostri team possono avere responsabilità in vari ambiti di più modelli, a seconda del carico di lavoro che supportano. Potrebbe esserti utile effettuare una suddivisione in aree disciplinari più specializzate, rispetto a quelle di alto livello qui descritte. Questi modelli offrono infinite possibilità di variazioni, separando o aggregando le attività o sovrapponendo i team e fornendo dettagli più specifici.

Potresti rilevare che disponi di funzionalità sovrapposte o non riconosciute da un team all'altro, che possono fornire ulteriori vantaggi o migliorare l'efficienza. Potresti inoltre identificare le esigenze insoddisfatte nella tua organizzazione e pianificare di affrontarle.

Durante la valutazione del cambiamento organizzativo, esamina i compromessi tra i diversi modelli, la posizione dei singoli team all'interno dei modelli (allo stato attuale e dopo il cambiamento), come

cambieranno le relazioni e le responsabilità dei team e se i vantaggi meritano l'impatto sulla tua organizzazione.

È possibile utilizzare con successo ciascuno dei seguenti quattro modelli operativi. Alcuni modelli sono più appropriati per casi d'uso specifici o in momenti specifici dello sviluppo. Alcuni di questi modelli possono offrire vantaggi rispetto a quelli in uso nel tuo ambiente.

### Argomenti

- [Modello operativo completamente separato](#)
- [Progettazione e operazioni delle applicazioni \(Application Engineering and Operations, AEO\) e progettazione e operazioni dell'infrastruttura \(Infrastructure Engineering and Operations, IEO\) separati con governance centralizzata](#)
- [AEO e IEO separati con governance centralizzata e un fornitore di servizi](#)
- [AEO e IEO separati con governance centralizzata, provider di servizi gestiti interno e partner di consulenza](#)
- [AEO e IEO separati con governance decentralizzata](#)

### Modello operativo completamente separato

Nel seguente diagramma, sull'asse verticale si trovano applicazioni e infrastruttura. Le applicazioni si riferiscono al carico di lavoro preposto a un risultato aziendale e possono consistere di software sviluppato in modo personalizzato o pronto all'uso. L'infrastruttura si riferisce all'infrastruttura fisica e virtuale e ad altri software che supportano tale carico di lavoro.

Sull'asse orizzontale si trovano progettazione e operazioni. La progettazione si riferisce allo sviluppo, alla realizzazione e all'esecuzione di test di applicazioni e infrastruttura. Le operazioni consistono nella distribuzione, l'aggiornamento e il supporto continuo delle applicazioni e dell'infrastruttura.

## Modello tradizionale



Questo modello "completamente separato" è presente in molte organizzazioni. Le attività in ciascun quadrante vengono eseguite da un team separato. Il lavoro è trasferito da un team all'altro attraverso meccanismi quali richieste di lavoro, code di lavoro, ticket o utilizzando un sistema di gestione dei servizi IT (ITSM).

La transizione delle attività verso o tra i team aumenta la complessità e crea ritardi e colli di bottiglia. Le richieste tendono a essere posticipate finché non rappresentano una priorità. I difetti identificati in ritardo potrebbero richiedere una rilavorazione significativa e un secondo passaggio attraverso gli stessi team e le loro funzioni. Se ci sono incidenti che richiedono un'azione da parte dei team di tecnici, le loro risposte vengono ritardate dall'attività di consegna.

Quando i team aziendali, di sviluppo e operativi sono organizzati in base alle attività o alle funzioni eseguite, esiste un maggiore rischio di disallineamento. Questo può portare i team a concentrarsi sulle loro responsabilità specifiche anziché sul raggiungimento dei risultati aziendali. I team possono essere strettamente specializzati, fisicamente isolati o logicamente isolati, ostacolando la comunicazione e la collaborazione.

Progettazione e operazioni delle applicazioni (Application Engineering and Operations, AEO) e progettazione e operazioni dell'infrastruttura (Infrastructure Engineering and Operations, IEO) separati con governance centralizzata

Il modello "AEO e IEO separati" segue una metodologia "chi crea, esegue".

Gli ingegneri e gli sviluppatori delle applicazioni si occupano sia della progettazione sia del funzionamento dei propri carichi di lavoro. Analogamente, i tecnici dell'infrastruttura si occupano sia della progettazione sia del funzionamento delle piattaforme che utilizzano per supportare i team dell'applicazione.



Per questo esempio, prenderemo in esame la governance centralizzata. Gli standard sono distribuiti, forniti o condivisi ai team applicativi.

È consigliabile utilizzare strumenti o servizi che consentono di gestire centralmente gli ambienti su più account, ad esempio [AWS Organizations](#). Servizi come [AWS Control Tower](#) ampliano questa

funzionalità di gestione consentendoti di definire piani (a supporto dei tuoi modelli operativi) per configurare gli account, applicare la governance continua tramite AWS Organizations e automatizzare il provisioning di nuovi account.

"Chi crea, esegue" non significa che il team applicativo sia responsabile dello stack, della catena di strumenti e della piattaforma nel loro complesso.

Il team di progettazione della piattaforma fornisce un set standardizzato di servizi (ad esempio, strumenti di sviluppo, strumenti di monitoraggio, strumenti di backup e ripristino, e rete) al team applicativo. Il team della piattaforma può anche fornire al team applicativo l'accesso a servizi di fornitori di servizi cloud approvati, configurazioni specifiche dello stesso o entrambi.

I meccanismi che forniscono una capacità self-service per la distribuzione di servizi e configurazioni approvati, come [Service Catalog](#), possono contribuire a limitare i ritardi associati alle richieste di adempimento applicando al contempo la governance.

Il team della piattaforma consente la visibilità completa dello stack, in modo che i team applicativi possano distinguere tra i problemi dei componenti dell'applicazione e i problemi dei componenti dei servizi e dell'infrastruttura utilizzati dalle applicazioni. Il team della piattaforma può inoltre fornire assistenza per la configurazione di questi servizi, nonché indicazioni su come migliorare le operazioni dei team applicativi.

Come illustrato in precedenza, è fondamentale che il team applicativo disponga di meccanismi per richiedere aggiunte, modifiche ed eccezioni agli standard, a supporto delle attività dei team e dell'innovazione della loro applicazione.

Il modello AEO e IEO separati fornisce cicli di feedback solidi ai team applicativi. Le operazioni quotidiane di un carico di lavoro aumentano il contatto con i clienti attraverso l'interazione diretta o indirettamente, attraverso richieste di supporto e funzionalità. Questa maggiore visibilità consente ai team applicativi di risolvere i problemi più rapidamente. Il coinvolgimento più profondo e la relazione più stretta forniscono informazioni sulle esigenze dei clienti e consentono un'innovazione più rapida.

Tutto questo vale anche per il team della piattaforma che supporta i team applicativi.

Gli standard adottati possono essere pre-approvati per l'uso, riducendo la quantità di revisione necessaria per entrare in produzione. L'utilizzo di standard supportati e testati forniti dal team della piattaforma può ridurre la frequenza dei problemi relativi a tali servizi. L'adozione degli standard consente ai team applicativi di concentrarsi sulla differenziazione dei propri carichi di lavoro.

## AEO e IEO separati con governance centralizzata e un fornitore di servizi

Il modello "AEO e IEO separati" segue una metodologia "chi crea, esegue".

Gli ingegneri e gli sviluppatori delle applicazioni si occupano sia della progettazione sia del funzionamento dei propri carichi di lavoro.

La tua organizzazione potrebbe non avere le competenze o il personale per supportare un team dedicato alla progettazione e alle operazioni della piattaforma, oppure potresti non volere investire tempo e fatica in questa direzione.

In alternativa, potresti disporre di un team di piattaforma incentrato sulla creazione di funzionalità che differenzieranno la tua attività, ma desideri delegare le operazioni quotidiane indifferenziate a un fornitore esterno.

I fornitori di servizi gestiti, come [AWS Managed Services](#), [AWS Managed Services Partners](#) o i fornitori di servizi gestiti della [Rete dei partner AWS](#) offrono esperienza nell'implementazione di ambienti cloud e supportano i requisiti di sicurezza e conformità e gli obiettivi aziendali.



Per questa variante, prenderemo in esame la governance centralizzata e gestita dal team della piattaforma, con la creazione di account e la gestione di policy con AWS Organizations e AWS Control Tower.

Questo modello richiede di modificare i tuoi meccanismi affinché funzionino con quelli del fornitore di servizi. Non risolve i colli di bottiglia e i ritardi creati dalla transizione delle attività tra i team, tra cui il fornitore di servizi, né le potenziali rilavorazioni correlate all'identificazione tardiva dei difetti.

Puoi trarre vantaggio dagli standard, dalle best practice, dai processi e dalle competenze dei tuoi fornitori. Inoltre, ottieni i vantaggi dello sviluppo continuo delle loro offerte di servizi.

L'aggiunta di servizi gestiti al tuo modello operativo ti consente di risparmiare tempo e risorse e ti permette di mantenere i team interni snelli e focalizzati sui risultati strategici che differenzieranno la tua attività, anziché sullo sviluppo di nuove competenze e funzionalità.

## AEO e IEO separati con governance centralizzata, provider di servizi gestiti interno e partner di consulenza

Il modello "AEO e IEO separati" cerca di stabilire una metodologia "chi crea, esegue".

Si desidera che i team applicativi svolgano le attività di ingegneria e di gestione dei loro carichi di lavoro e che adottino una cultura più simile a DevOps.

I team applicativi potrebbero essere in fase di migrazione, di adozione del cloud o di modernizzazione dei carichi di lavoro e non disporre delle competenze necessarie per supportare adeguatamente il cloud e le operazioni cloud. Tale mancanza di capacità o di familiarità del team applicativo può costituire un ostacolo agli sforzi.

Per affrontare questa problematica è necessario creare un team di Cloud Center of Enablement (CCoE) che metta a disposizione un forum per porre domande, discutere le esigenze e identificare le soluzioni. A seconda delle esigenze dell'organizzazione, il CCoE può essere costituito da un team dedicato di esperti o da un team virtuale con partecipanti selezionati da tutta l'organizzazione. Il CCoE consente la trasformazione del cloud per i team, stabilisce una governance centralizzata del cloud e definisce gli standard di gestione degli account e dell'organizzazione. Identifica inoltre architetture di riferimento e modelli di successo per l'uso aziendale.

Ci riferiamo al CCoE come Cloud Center of Enablement, invece del più comune Cloud Center of Excellence, per porre l'accento sull'abilitazione del successo dei team supportati e sul raggiungimento dei risultati aziendali.

Il team di ingegneri della piattaforma costruisce le funzionalità di base della piattaforma condivisa sulla base di questi standard che i team applicativi devono adottare. Codifica le architetture e i modelli di riferimento aziendali che vengono forniti ai team applicativi attraverso un meccanismo self-service. Utilizzando un servizio come AWS Service Catalog, i team applicativi possono distribuire architetture di riferimento, modelli, servizi e configurazioni approvati, conformi per impostazione predefinita agli standard di sicurezza e di governance centralizzati.

Il team di progettazione della piattaforma fornisce un set standardizzato di servizi (ad esempio, strumenti di sviluppo, strumenti di monitoraggio, strumenti di backup e ripristino, e rete) ai team applicativi.

La tua organizzazione dispone di un "Provider di servizi gestiti interno e partner di consulenza" che gestisce e supporta i servizi standardizzati e fornisce assistenza ai team applicativi i quali stabiliscono la loro presenza nel cloud sulla base delle architetture e dei modelli di riferimento. Il team

"Cloud Operations and Platform Enablement (COPE)" collabora con i team applicativi per aiutarli a stabilire le operazioni di base, mentre i team applicativi assumono progressivamente una maggiore responsabilità per i loro sistemi e le loro risorse nel corso del tempo. Il team COPE promuove il miglioramento continuo insieme ai team CCoE e Platform Engineering e agisce come promotore per i team applicativi.

I team applicativi ricevono assistenza per la creazione di ambienti, pipeline CI/CD, gestione delle modifiche, osservabilità e monitoraggio, nonché per la creazione di processi di gestione degli incidenti e degli eventi con il team COPE integrati con quelli dell'azienda, secondo necessità. Il team COPE partecipa con i team applicativi allo svolgimento di tali attività operative, riducendo gradualmente l'impegno del team COPE nel corso del tempo, man mano che i team applicativi se ne assumono la responsabilità.

Il team applicativo beneficia delle competenze del team COPE e delle lezioni apprese dall'organizzazione. È protetto dai guardrail stabiliti dalla governance centralizzata. Il team applicativo si basa sui successi riconosciuti e ottiene il beneficio di uno sviluppo continuo degli standard organizzativi adottati. Grazie al processo di osservabilità e monitoraggio, le persone acquisiscono una maggiore conoscenza del funzionamento del loro carico di lavoro e sono in grado di comprendere meglio l'impatto delle modifiche ad esso apportate.



Il team COPE mantiene l'accesso necessario per supportare le attività operative, fornire una visione delle operazioni aziendali che abbracci i team applicativi e fornire un supporto alla gestione degli incidenti critici. Il team COPE mantiene la responsabilità delle attività considerate come indifferenziate e a basso valore aggiunto, che soddisfa attraverso soluzioni standard supportabili su larga scala. Continua inoltre a gestire attività programmatiche e automatizzate ben comprese dai team applicativi, in modo che possano concentrarsi sulla differenziazione delle loro applicazioni.

In questo modo approfitti del vantaggio degli standard, delle best practice, dei processi e delle competenze dell'organizzazione, derivanti dai successi dei suoi team. Stabilisci un meccanismo per replicare questi modelli di successo per i nuovi team che adottano o modernizzano il cloud. Questo modello pone l'accento sulla capacità del team COPE di aiutare il team applicativo ad avviare le attività e a trasferire conoscenze e artefatti. Riduce gli oneri operativi dei team applicativi con il rischio che questi ultimi non riescano a diventare ampiamente indipendenti. Definisce le relazioni tra CCoE, COPE e team applicativi, creando un ciclo di feedback per supportare ulteriori evoluzioni e innovazioni.

L'istituzione di team CCoE e COPE e la definizione di standard a livello di organizzazione possono facilitare l'adozione del cloud e sostenere gli sforzi di modernizzazione. Attraverso il supporto aggiuntivo di un team COPE che agisce come consulente e partner dei team applicativi, è possibile rimuovere le barriere che rallentano l'adozione delle vantaggiose funzionalità cloud da parte dei team applicativi.

## AEO e IEO separati con governance decentralizzata

Il modello "AEO e IEO separati" segue una metodologia "chi crea, esegue".

Gli ingegneri e gli sviluppatori delle applicazioni si occupano sia della progettazione sia del funzionamento dei propri carichi di lavoro. Analogamente, i tecnici dell'infrastruttura si occupano sia della progettazione sia del funzionamento delle piattaforme che utilizzano per supportare i team dell'applicazione.



Per questo esempio, prenderemo in esame la governance decentralizzata.

Gli standard sono ancora distribuiti, forniti o condivisi ai team applicativi dal team della piattaforma, ma i team applicativi sono liberi di progettare e gestire nuove funzionalità della piattaforma a supporto del proprio carico di lavoro.

In questo modello, ci sono meno vincoli per il team applicativo, ma ciò comporta un aumento significativo delle responsabilità. Devono essere presenti ulteriori competenze, e potenzialmente altro personale, per supportare le funzionalità aggiuntive della piattaforma. Il rischio di rilavorazione significativa aumenta se i set di competenze non sono adeguati e i difetti non vengono riconosciuti in anticipo.

È consigliabile applicare policy che non siano specificamente delegate ai team applicativi. Utilizza strumenti o servizi che consentono di gestire centralmente gli ambienti su più account, ad esempio [AWS Organizations](#). Servizi come [AWS Control Tower](#) ampliano questa funzionalità di gestione consentendoti di definire piani (a supporto dei tuoi modelli operativi) per configurare gli account, applicare la governance continua tramite AWS Organizations e automatizzare il provisioning di nuovi account.

È preferibile che il team applicativo disponga di meccanismi per richiedere aggiunte e modifiche agli standard. Il team potrebbe essere in grado di contribuire a nuovi standard che possono fornire vantaggi ad altri team applicativi. I team della piattaforma possono decidere che fornire supporto diretto per queste funzionalità aggiuntive sia un contributo efficace ai risultati aziendali.

Questo modello limita i vincoli all'innovazione e presenta requisiti significativi in termini di competenze e personale. Risolve molti dei colli di bottiglia e dei ritardi creati dalla transizione delle attività tra i team, promuovendo allo stesso tempo lo sviluppo di relazioni efficaci tra team e clienti.

## Relazioni e proprietà

Il modello operativo definisce le relazioni tra i team e supporta proprietà e responsabilità identificabili.

### Best practice

- [OPS02-BP01 Associazione di proprietari identificati alle risorse](#)
- [OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure](#)
- [OPS02-BP03 Assegnazione di proprietari identificati alle operazioni che siano responsabili delle relative prestazioni](#)
- [OPS02-BP04 Definizione di meccanismi per gestire responsabilità e titolarità](#)
- [OPS02-BP05 Definizione di meccanismi per richiedere aggiunte, modifiche ed eccezioni](#)

- [OPS02-BP06 Predefinizione o negoziazione delle responsabilità tra i team](#)

## OPS02-BP01 Associazione di proprietari identificati alle risorse

Le risorse per il tuo carico di lavoro devono disporre di proprietari identificati per il controllo delle modifiche, la risoluzione dei problemi e altre funzioni. I titolari sono assegnati a carichi di lavoro, account, infrastrutture, piattaforme e applicazioni. La proprietà viene registrata tramite strumenti come un registro centrale o metadati collegati alle risorse. Il valore aziendale dei componenti è alla base dei processi e delle procedure applicate.

Risultato desiderato:

- Le risorse hanno identificato i titolari tramite i metadati o un registro centrale.
- I membri del team possono identificare chi è il titolare delle risorse.
- Gli account hanno un solo proprietario, laddove possibile.

Anti-pattern comuni:

- I contatti alternativi per il tuo Account AWS non sono inseriti.
- Le risorse non hanno tag che identificano i team che le possiedono.
- Hai una coda ITSM senza una mappatura delle e-mail.
- Due team hanno entrambi la proprietà di una parte critica dell'infrastruttura.

Vantaggi dell'adozione di questa best practice:

- Il controllo delle modifiche per le risorse è immediato con la proprietà assegnata.
- Puoi coinvolgere i proprietari corretti quando risolvi i problemi.

Livello di rischio associato se questa best practice non fosse adottata: Elevato

Guida all'implementazione

Definisci qual è il significato della proprietà per i casi d'uso delle risorse nel tuo ambiente. Proprietà significa supervisionare le modifiche alla risorsa, supportare la risorsa durante la risoluzione dei problemi o essere finanziariamente affidabile. Specifica e registra i proprietari delle risorse, con nome, informazioni di contatto, organizzazione e team.

## Esempio del cliente

AnyCompany Retail definisce il proprietario come il team o l'individuo responsabile delle modifiche e del supporto per le risorse. Si avvale di AWS Organizations per gestire gli Account AWS. Contatti alternativi degli account vengono configurati con caselle di posta di gruppo. Ogni coda ITSM è mappata su un alias e-mail. I tag identificano il proprietario delle risorse AWS. Per altre piattaforme e infrastrutture hanno una pagina wiki che identifica la proprietà e le informazioni di contatto.

## Passaggi dell'implementazione

1. Inizia definendo la proprietà dell'organizzazione. La proprietà può significare essere proprietari del rischio collegato alla risorsa, essere proprietari delle modifiche alla risorsa o supportare la risorsa durante la risoluzione dei problemi. Proprietà può anche significare proprietà amministrativa o finanziaria della risorsa.
2. Usa [AWS Organizations](#) per gestire gli account. Puoi gestire centralmente i contatti alternativi per gli account.
  - a. Se usi indirizzi e-mail e numeri di telefono aziendali come informazioni di contatto, puoi accedervi anche se le persone a cui appartengono non fanno più parte dell'organizzazione. Ad esempio, crea elenchi di distribuzione delle e-mail separati per fatturazione, operazioni e sicurezza e configurali come contatti per Fatturazione, Sicurezza e Operazioni in ogni Account AWS attivo. Molteplici persone riceveranno notifiche AWS e potranno rispondere, anche se qualcuno è in vacanza, ha cambiato ruolo o ha lasciato l'azienda.
  - b. Se un account non è gestito da [AWS Organizations](#), i contatti alternativi dell'account consentono ad AWS di comunicare con il personale richiesto, se necessario. Configura i contatti alternativi dell'account per indirizzare le persone a un gruppo invece che a un individuo.
3. Usa i tag per identificare i proprietari delle risorse AWS. Puoi specificare sia i proprietari sia le loro informazioni di contatto in tag separati.
  - a. Puoi usare le regole [AWS Config](#) per avere la certezza che le risorse abbiano i tag di proprietà richiesti.
  - b. Per linee guida dettagliate su come creare una strategia per l'applicazione di tag per l'organizzazione, consulta il [whitepaper Best Practices for Tagging AWS Resources](#).
4. Usa [Amazon Q Business](#), un assistente conversazionale che utilizza l'IA generativa per migliorare la produttività della forza lavoro, rispondere alle domande e completare le attività in base alle informazioni presenti nei sistemi aziendali.
  - a. Connetti Amazon Q Business all'origine dati dell'azienda. Amazon Q Business offre connettori predefiniti per oltre 40 origini dati supportate, tra cui Amazon Simple Storage Service (Amazon

S3), Microsoft SharePoint, Salesforce e Atlassian Confluence. Per ulteriori informazioni, consulta [Connettori di Amazon Q Business](#).

5. Per altre risorse, piattaforme e infrastrutture, crea la documentazione che stabilisce la proprietà. Tutti i membri del team devono poter accedere a queste informazioni.

Livello di impegno per il piano di implementazione: Basso. Sfrutta le informazioni di contatto e i tag degli account per assegnare la proprietà delle risorse AWS. Per altre risorse puoi usare qualcosa di semplice come una tabella in un wiki per registrare le informazioni su proprietà e contatti o usare uno strumento ITSM per mappare la proprietà.

## Risorse

Best practice correlate:

- [OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure](#)
- [OPS02-BP04 Definizione di meccanismi per gestire responsabilità e proprietà](#)

Documenti correlati:

- [AWS Account Management - Updating contact information](#)
- [AWS Organizations - Updating alternative contacts in your organization](#)
- [Whitepaper Best Practices for Tagging AWS Resources](#)
- [Build private and secure enterprise generative AI apps with Amazon Q Business and AWS IAM Identity Center](#)
- [Amazon Q Business, now generally available, helps boost workforce productivity with generative AI](#)
- [Cloud AWS Operations & Migrations Blog - Implementing automated and centralized tagging controls with AWS Config and AWS Organizations](#)
- [AWS Security Blog - Extend your pre-commit hooks with AWS CloudFormation Guard](#)
- [AWS DevOps Blog - Integrating AWS CloudFormation Guard into CI/CD pipelines](#)

Workshop correlati:

- [AWS Workshop - Tagging](#)

Esempi correlati:

- [Regole di AWS Config - Amazon EC2 with required tags and valid values](#)

Servizi correlati:

- [Regole di AWS Config - required-tags](#)
- [AWS Organizations](#)

## OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure

È utile comprendere chi ha la proprietà della definizione di singoli processi e procedure, perché tali processi e procedure specifici vengono utilizzati e perché tale proprietà esiste. Comprendere i motivi per cui vengono utilizzati processi e procedure specifici aiuta a identificare le opportunità di miglioramento.

Risultato desiderato: l'organizzazione dispone di una serie di processi e procedure per le attività operative ben definiti e gestiti. I processi e le procedure sono archiviati in una posizione centrale e disponibili per i membri del team. I processi e le procedure vengono aggiornati frequentemente attraverso l'assegnazione chiara della proprietà. Ove possibile, script, modelli e documenti di automazione vengono implementati come codice.

Anti-pattern comuni:

- I processi non sono documentati. Possono essere presenti script frammentati su workstation degli operatori isolate.
- La conoscenza relativa all'uso degli script è nelle mani di pochi individui oppure viene acquisita in modo informale come conoscenza di team.
- È necessario aggiornare un processo legacy, ma la proprietà dell'aggiornamento non è chiara e l'autore originale non fa più parte dell'organizzazione.
- I processi e gli script non sono individuabili, quindi non sono immediatamente disponibili quando necessario (ad esempio, in risposta a un incidente).

Vantaggi dell'adozione di questa best practice:

- I processi e le procedure incentivano l'impegno nella gestione dei carichi di lavoro.
- I nuovi membri del team diventano efficienti più rapidamente.
- Tempi ridotti per mitigare gli incidenti.

- Membri del team (e team) diversi possono utilizzare gli stessi processi e procedure in modo coerente.
- I team scalano i processi tramite processi ripetibili.
- Processi e procedure standardizzati aiutano a mitigare l'impatto del trasferimento delle responsabilità del carico di lavoro tra i team.

Livello di rischio associato se questa best practice non fosse adottata: elevato

### Guida all'implementazione

- Esistono proprietari identificati di processi e procedure, che sono responsabili della loro definizione.
  - Identifica le attività operative eseguite a supporto dei carichi di lavoro. Documenta queste attività in un percorso individuabile.
  - Identifica in modo univoco la persona o il team responsabile della specifica di un'attività. Questo soggetto deve verificare che essa possa essere eseguita correttamente dal componente di un team adeguatamente qualificato, che disponga di autorizzazioni, accesso e strumenti adeguati. In caso di problemi nello svolgimento di tale attività, i membri del team che la eseguono sono responsabili della redazione dei feedback dettagliati necessari per migliorarla.
  - Acquisisci la responsabilità dei metadati dell'artefatto dell'attività tramite servizi come AWS Systems Manager, documenti e AWS Lambda. Acquisisci la responsabilità delle risorse utilizzando tag o gruppi di risorse, specificando proprietà e informazioni di contatto. Utilizza AWS Organizations per creare policy di tagging e garantire l'acquisizione di proprietà e informazioni di contatto.
- Nel tempo, queste procedure si evolvono per essere eseguibili come codice, riducendo la necessità dell'intervento umano.
  - Ad esempio, prendi in considerazione le funzioni AWS Lambda, i modelli CloudFormation o i documenti di automazione AWS Systems Manager.
  - Esegui il controllo delle versioni nei repository appropriati.
  - Applica i tag adeguati alle risorse, in modo da facilitare l'identificazione di proprietari e documentazione.

### Esempio del cliente

AnyCompany Retail definisce come titolare il team o l'individuo responsabile dei processi per un'applicazione o gruppi di applicazioni (che condividono procedure e tecnologie architetturali

comuni). Inizialmente, i processi e le procedure sono documentati nel sistema di gestione dei documenti come guide dettagliate, individuabili tramite i tag dell'Account AWS che ospita l'applicazione e di gruppi specifici di risorse all'interno dell'account. Si avvalgono di AWS Organizations per gestire gli Account AWS. Nel tempo, questi processi vengono convertiti in codice e le risorse vengono definite utilizzando l'infrastruttura as code (ad esempio CloudFormation o modelli AWS Cloud Development Kit (AWS CDK)). I processi operativi diventano documenti di automazione in AWS Systems Manager o nelle funzioni AWS Lambda, che possono essere avviati come attività pianificate in risposta a eventi, ad esempio allarmi AWS CloudWatch o eventi AWS EventBridge, oppure attivati da richieste di una piattaforma di gestione dei servizi IT (ITSM). Tutti i processi dispongono dei tag per identificare la titolarità. La documentazione per l'automazione e il processo viene mantenuta all'interno delle pagine wiki generate dal repository di codice per il processo.

### Passaggi dell'implementazione

1. Documenta i processi e le procedure esistenti.
  - a. Rivedili e mantienili aggiornati.
  - b. Identifica un proprietario per ogni processo o procedura.
  - c. Applica a ognuno il controllo della versione.
  - d. Ove possibile, condividi processi e procedure tra carichi di lavoro e ambienti che condividono progetti architetturali.
2. Stabilisci meccanismi di feedback e miglioramento.
  - a. Definisci policy relative alla frequenza con cui i processi devono essere rivisti.
  - b. Definisci i processi per revisori e approvatori.
  - c. Implementa i problemi o crea una coda di ticket per fornire e monitorare il feedback.
  - d. Ove possibile, i processi e le procedure devono essere approvati preventivamente e classificati in base ai rischi da parte di un comitato di approvazione delle modifiche (CAB).
3. Verifica che i processi e le procedure siano accessibili e individuabili da chi deve eseguirli.
  - a. Utilizza i tag per indicare dove è possibile accedere ai processi e alle procedure per il carico di lavoro.
  - b. Utilizza messaggi di errore ed eventi significativi per indicare i processi o le procedure appropriati per risolvere un problema.
  - c. Usa i wiki e la gestione dei documenti per rendere i processi e le procedure consultabili in modo coerente in tutta l'organizzazione.
4. Automatizza quando appropriato.

- a. È opportuno eseguire le automazioni quando servizi e tecnologie forniscono un'API.
- b. Istruisci adeguatamente in merito ai processi. Sviluppa i casi utente e i requisiti per automatizzare i processi.
- c. Misura correttamente l'uso di processi e procedure e sfrutta i problemi come un'opportunità di miglioramento continuo.

Livello di impegno per il piano di implementazione: medio

Risorse

Best practice correlate:

- [OPS02-BP01 Associazione di titolari identificati alle risorse](#)
- [OPS02-BP04 Definizione di meccanismi per gestire responsabilità e titolarità](#)
- [OPS11-BP04 Gestione delle conoscenze](#)

Documenti correlati:

- [Whitepaper AWS - Introduction to DevOps on AWS](#)
- [AWS Whitepaper - Best Practices for Tagging AWS Resources](#)
- [AWS Whitepaper - Organizing Your AWS Environment Using Multiple Accounts](#)
- [Cloud AWS Operations & Migrations Blog - Build a Cloud Automation Practice for Operational Excellence: Best Practices from AWS Managed Services](#)
- [Cloud AWS Operations & Migrations Blog - Implementing automated and centralized tagging controls with AWS Config and AWS Organizations](#)
- [AWS Security Blog - Extend your pre-commit hooks with AWS CloudFormation Guard](#)
- [AWS DevOps Blog - Integrating AWS CloudFormation Guard into CI/CD pipelines](#)

Workshop correlati:

- [AWS Well-Architected Operational Excellence Workshop](#)
- [AWS Workshop - Tagging](#)

Video correlati:

- [How to automate IT Operations on AWS](#)
- [AWS re:Invent 2020 - Automate anything with AWS Systems Manager](#)
- [AWS re:Inforce 2022 - Automating patch management and compliance using AWS \(NIS306\)](#)
- [AWS Supports You - Diving Deep into AWS Systems Manager](#)

Servizi correlati:

- [AWS Systems Manager - Automation](#)
- [AWS Service Management Connector](#)

OPS02-BP03 Assegnazione di proprietari identificati alle operazioni che siano responsabili delle relative prestazioni

È utile comprendere chi ha la responsabilità di eseguire attività specifiche su carichi di lavoro definiti e perché tale responsabilità esiste. Comprendere chi ha la responsabilità di eseguire le attività fornisce indicazioni su chi eseguirà l'attività, su chi convaliderà il risultato e su chi fornirà feedback al titolare dell'attività.

Risultato desiderato:

L'organizzazione definisce chiaramente le responsabilità per eseguire attività specifiche su carichi di lavoro stabiliti e rispondere agli eventi generati dai carichi di lavoro. L'organizzazione documenta la responsabilità dei processi e degli impegni e rende queste informazioni individuabili. Esamini e aggiorna le responsabilità quando avvengono cambiamenti organizzativi e i team monitorano e misurano le prestazioni delle attività di identificazione dei difetti e delle inefficienze. Implementi i meccanismi di feedback per monitorare i difetti e i miglioramenti e supportare il miglioramento continuo.

Anti-pattern comuni:

- Non documenta le responsabilità.
- Sono presenti script frammentati sulle workstation degli operatori isolate. Solo poche persone sanno come usarli o li chiamano informalmente conoscenze del team.
- Un processo legacy deve essere aggiornato, ma non si sa chi è il proprietario e l'autore originale non fa più parte dell'organizzazione.
- I processi e gli script non sono individuabili, quindi non sono immediatamente disponibili quando necessario, ad esempio, in risposta a un incidente.

Vantaggi dell'adozione di questa best practice:

- Comprendi chi è responsabile dell'esecuzione di un'attività, a chi notificare un'azione necessaria e chi esegue l'azione, convalida il risultato e fornisce il feedback al titolare dell'attività.
- I processi e le procedure incentivano l'impegno nella gestione dei carichi di lavoro.
- I nuovi membri del team diventano efficienti più rapidamente.
- Riduci il tempo necessario per mitigare gli incidenti.
- Team diversi utilizzano medesimi processi e procedure per eseguire le attività in modo coerente.
- I team scalano i processi tramite processi ripetibili.
- Processi e procedure standardizzati aiutano a mitigare l'impatto del trasferimento delle responsabilità del carico di lavoro tra i team.

Livello di rischio associato se questa best practice non fosse adottata: elevato

### Guida all'implementazione

Per definire le responsabilità, inizia usando la documentazione esistente, ad esempio matrici di responsabilità, processi e procedure, ruoli e responsabilità, strumenti e automazione. Esamina la documentazione e organizza discussioni sulle responsabilità dei processi documentati. Collaborando con i team identifica i disallineamenti tra le responsabilità e i processi documentati. Parla dei servizi offerti con i clienti interni di un team per identificare le divergenze nelle aspettative tra i team.

Analizza e risolvi le discrepanze. Identifica le opportunità di miglioramento e individua le attività richieste di frequente e con uso intensivo di risorse, che in genere sono ottime candidate al miglioramento. Esamina le best practice, i modelli e le linee guida prescrittive per semplificare e standardizzare i miglioramenti. Registra le opportunità di miglioramento e monitora i miglioramenti fino al completamento.

Nel tempo, queste procedure si evolvono per essere eseguibili come codice, riducendo la necessità dell'intervento umano. Ad esempio, le procedure possono essere avviate come funzioni AWS Lambda, modelli AWS CloudFormation o documenti di automazione AWS Systems Manager. Verifica che queste procedure siano sottoposte al controllo delle versioni nei repository appropriati e includano i corretti tag delle risorse in modo che i team possano identificare prontamente i responsabili e la documentazione. Documenta la responsabilità dello svolgimento delle attività, quindi monitora l'avvio e il funzionamento delle automazioni, nonché le prestazioni dei risultati desiderati.

### Esempio del cliente

AnyCompany Retail definisce come titolare il team o l'individuo responsabile dei processi per un'applicazione o gruppi di applicazioni (che condividono procedure e tecnologie architetturali comuni). Inizialmente, l'azienda documenta i processi e le procedure come guide dettagliate nel sistema di gestione dei documenti. Rende le procedure individuabili applicando i tag nell'Account AWS che ospita l'applicazione e in gruppi specifici di risorse dell'account, utilizzando AWS Organizations per gestire gli Account AWS. Nel tempo, AnyCompany Retail converte questi processi in codice e definisce le risorse utilizzando l'infrastruttura as code, tramite servizi come CloudFormation o modelli AWS Cloud Development Kit (AWS CDK). I processi operativi diventano documenti di automazione in AWS Systems Manager o nelle funzioni AWS Lambda, che possono essere avviati come attività pianificate in risposta a eventi, ad esempio allarmi Amazon CloudWatch o eventi Amazon EventBridge, oppure da richieste di una piattaforma di gestione dei servizi IT (ITSM). Tutti i processi dispongono dei tag per identificare il proprietario. I team gestiscono la documentazione per l'automazione e il processo nelle pagine wiki generate dal repository di codice per il processo.

## Passaggi dell'implementazione

1. Documenta i processi e le procedure esistenti.
  - a. Esamina e verifica che siano aggiornati.
  - b. Verifica che ogni processo o procedura abbia un titolare.
  - c. Applica alle procedure il controllo delle versioni.
  - d. Ove possibile, condividi processi e procedure tra carichi di lavoro e ambienti che condividono strutture architetturali.
2. Stabilisci meccanismi di feedback e miglioramento.
  - a. Definisci policy relative alla frequenza con cui i processi devono essere rivisti.
  - b. Definisci i processi per revisori e approvatori.
  - c. Implementa i problemi o crea una coda di ticket per fornire e monitorare il feedback.
  - d. Ove possibile, i processi e le procedure devono essere approvati preventivamente e classificati in base ai rischi da parte di un comitato di approvazione delle modifiche (CAB).
3. Rendi i processi e le procedure accessibili e individuabili dagli utenti che devono eseguirli.
  - a. Utilizza i tag per indicare dove è possibile accedere ai processi e alle procedure per il carico di lavoro.
  - b. Utilizza messaggi di errore ed eventi significativi per indicare i processi o le procedure appropriati per risolvere il problema.

- c. Usa i wiki o la gestione dei documenti per rendere i processi e le procedure consultabili in modo coerente in tutta l'organizzazione.
4. Automatizza quando è opportuno farlo.
    - a. Laddove servizi e tecnologie forniscono un'API, sviluppa le automazioni.
    - b. Verifica che i processi siano ben compresi e sviluppa i casi utente e i requisiti per automatizzare i processi.
    - c. Misura l'uso corretto di processi e procedure e sfrutta i problemi per supportare il miglioramento continuo.

Livello di impegno per il piano di implementazione: medio

Risorse

Best practice correlate:

- [OPS02-BP01 Associazione di titolari identificati alle risorse](#)
- [OPS02-BP02 Assegnazione di titolari identificati a processi e procedure](#)
- [OPS02-BP04 Definizione di meccanismi per gestire responsabilità e titolarità](#)
- [OPS02-BP05 Definizione di meccanismi per identificare responsabilità e titolarità](#)
- [OPS11-BP04 Gestione delle conoscenze](#)

Documenti correlati:

- [Whitepaper AWS | Introduction to DevOps on AWS](#)
- [AWS Whitepaper | Best Practices for Tagging AWS Resources](#)
- [AWS Whitepaper | Organizing Your AWS Environment Using Multiple Accounts](#)
- [Cloud AWS Operations & Migrations Blog | Build a Cloud Automation Practice for Operational Excellence: Best Practices from AWS Managed Services](#)
- [AWS Workshop - Tagging](#)
- [AWS Service Management Connector](#)

Video correlati:

- [AWS Knowledge Center Live | Tagging AWS Resources](#)

- [AWS re:Invent 2020 | Automate anything with AWS Systems Manager](#)
- [AWS re:Inforce 2022 | Automating patch management and compliance using AWS \(NIS306\)](#)
- [AWS Supports You | Diving Deep into AWS Systems Manager](#)

Esempi correlati:

- [AWS Well-Architected Operational Excellence Workshop](#)

## OPS02-BP04 Definizione di meccanismi per gestire responsabilità e titolarità

Comprendi le responsabilità del tuo ruolo e il modo in cui contribuisce ai risultati aziendali in quanto questa conoscenza fornisce indicazioni sulle priorità delle tue attività e sul perché il tuo ruolo è importante. I membri del team possono quindi riconoscere le esigenze e rispondere in modo appropriato. Quando i membri del team comprendono il proprio ruolo, possono stabilire la titolarità, identificare le opportunità di miglioramento e capire come influenzare o apportare le modifiche appropriate.

Occasionalmente, una responsabilità potrebbe non avere un titolare definito. In queste situazioni, progetta un meccanismo per risolvere la lacuna. Crea un percorso di escalation ben definito a qualcuno con l'autorità di assegnare la responsabilità o il piano per risolvere il problema.

Risultato desiderato: i team all'interno dell'organizzazione hanno responsabilità chiaramente definite che includono il modo in cui sono correlate alle risorse, alle azioni da eseguire, ai processi e alle procedure. Queste responsabilità sono in linea con le responsabilità e gli obiettivi del team, nonché con le responsabilità degli altri team. Documenti i percorsi di escalation in modo coerente e individuabile e inserisci queste decisioni in artefatti di documentazione, come matrici di responsabilità, definizioni di team o pagine wiki.

Anti-pattern comuni:

- Le responsabilità del team sono ambigue o mal definite.
- Il team non allinea i ruoli alle responsabilità.
- Il team non allinea scopi e obiettivi alle responsabilità, rendendo difficile misurare il successo delle attività.
- Le responsabilità dei membri del team non sono in linea con il team e l'organizzazione in generale.
- Il team non mantiene aggiornate le responsabilità rendendole incoerenti con le attività svolte dal team.

- I percorsi di escalation per determinare le responsabilità non sono definiti o non sono chiari.
- I percorsi di escalation non hanno un unico responsabile del thread per garantire una risposta tempestiva.
- Ruoli, responsabilità e percorsi di escalation non sono individuabili e quindi non sono immediatamente disponibili quando richiesto, ad esempio in risposta a un incidente.

Vantaggi dell'adozione di questa best practice:

- Una volta compreso chi ha la responsabilità o la titolarità, puoi contattare il team o il membro del team appropriato per effettuare una richiesta o trasferire un'attività.
- Per ridurre il rischio di inattività e di esigenze non soddisfatte, identifichi una persona che ha l'autorità di assegnare responsabilità o titolarità.
- Quando si definisce chiaramente l'ambito di una responsabilità, i membri del team acquisiscono autonomia e titolarità.
- Le tue responsabilità forniscono indicazioni sulle decisioni che prendi, sulle azioni che intraprendi e sulle tue attività di distribuzione ai titolari appropriati.
- Ti sarà facile identificare le responsabilità abbandonate perché hai una chiara comprensione di ciò che non rientra nelle responsabilità del tuo team e quindi potrai effettuare l'escalation per chiedere chiarimenti.
- I team evitano confusione e tensione e possono gestire in modo più adeguato i carichi di lavoro e le risorse.

Livello di rischio associato se questa best practice non fosse adottata: elevato

### Guida all'implementazione

Identifica i ruoli e le responsabilità dei membri del team e verifica che comprendano le aspettative del proprio ruolo. Rendi queste informazioni individuabili in modo che i membri della tua organizzazione possano identificare il team o la persona da contattare per esigenze specifiche. Man mano che le organizzazioni capitalizzano le opportunità di migrazione e modernizzazione su AWS, i ruoli e le responsabilità potrebbero cambiare. Rendi i team e i membri consapevoli delle loro responsabilità e offri la formazione appropriata per svolgere le attività durante questo cambiamento.

Determina il ruolo o il team che deve ricevere le escalation per identificare responsabilità e titolarità. Questo team può interagire con varie parti interessate per prendere le decisioni. Tuttavia, è proprietario della gestione del processo decisionale.

Fornisci ai membri della tua organizzazione meccanismi accessibili per scoprire e identificare titolarità e responsabilità. Questi meccanismi insegnano loro a chi rivolgersi per esigenze specifiche.

### Esempio del cliente

AnyCompany Retail ha recentemente completato una migrazione dei carichi di lavoro da un ambiente on-premises alla zona di destinazione in AWS con un approccio lift and shift. Ha eseguito una revisione delle operazioni per esaminare come vengono svolte le attività operative comuni e ha verificato che la matrice di responsabilità esistente rifletta le operazioni nel nuovo ambiente. Quando ha eseguito la migrazione dall'ambiente on-premises ad AWS, ha ridotto le responsabilità dei team dell'infrastruttura relative all'hardware e all'infrastruttura fisica. Questo passaggio ha anche rivelato nuove opportunità per evolvere il modello operativo dei carichi di lavoro.

Oltre ad aver identificato, risolto e documentato la maggior parte delle responsabilità, ha anche definito i percorsi di escalation per eventuali responsabilità mancanti o che potrebbero cambiare con l'evolversi delle procedure operative. Per la ricerca di nuove opportunità per standardizzare e migliorare l'efficienza dei carichi di lavoro, fornisce l'accesso a strumenti operativi come AWS Systems Manager e strumenti di sicurezza come AWS Security Hub e Amazon GuardDuty. AnyCompany Retail combina una revisione delle responsabilità e della strategia sulla base dei miglioramenti che intende eseguire per primi. Man mano che l'azienda adotta nuovi modi di lavorare e modelli tecnologici, aggiorna la propria matrice di responsabilità di conseguenza.

### Passaggi dell'implementazione

1. Inizia con la documentazione esistente. Alcuni documenti di origine tipici possono essere:
  - a. Matrici di responsabilità o responsabili, affidabili, consultabili e informate (RACI).
  - b. Definizioni dei team o pagine wiki.
  - c. Definizioni e offerte di servizi.
  - d. Ruolo o descrizione delle mansioni lavorative.
2. Esamina la documentazione e organizza discussioni sulle responsabilità documentate:
  - a. Collaborando con i team identifica i disallineamenti tra le responsabilità documentate e quelle normalmente assunte dai team.
  - b. Esamina i potenziali servizi offerti dai clienti interni per identificare le lacune nelle aspettative tra i team.
3. Analizza e risolvi le discrepanze.
4. Identifica le opportunità di miglioramento.

- a. Identifica le richieste più frequenti e con uso intensivo di risorse, che in genere sono ottime candidate al miglioramento.
  - b. Esamina le best practice, i modelli e le linee guida prescrittive per semplificare e standardizzare i miglioramenti.
  - c. Registra le opportunità di miglioramento e monitorale fino al completamento.
5. Se nessuno nel team è responsabile della gestione e del monitoraggio dell'assegnazione delle responsabilità, identifica qualcuno che assuma tale responsabilità.
6. Definisci un processo per consentire ai team di richiedere chiarimenti sulla responsabilità.
- a. Esamina il processo e verifica che sia chiaro e semplice da usare.
  - b. Assicurati che qualcuno sia proprietario e segua le escalation fino al completamento.
  - c. Stabilisci le metriche operative per misurare l'efficacia.
  - d. Crea un meccanismo di feedback per verificare che i team possano evidenziare le opportunità di miglioramento.
  - e. Implementa un meccanismo di revisione periodica.
7. Rendi i documenti disponibili in una posizione individuabile e accessibile.
- a. I wiki o il portale di documentazione sono le posizioni normalmente scelte.

Livello di impegno per il piano di implementazione: medio

Risorse

Best practice correlate:

- [OPS01-BP06 Valutazione dei compromessi](#)
- [OPS03-BP02 Potere di intervento dei membri del team quando i risultati sono a rischio](#)
- [OPS03-BP03 Incoraggiamento all'escalation](#)
- [OPS03-BP07 Fornitura di risorse appropriate ai team](#)
- [OPS09-BP01 Misura gli obiettivi operativi e i KPI con le metriche](#)
- [OPS09-BP03 Revisione delle metriche operative e assegnazione delle priorità per favorire il miglioramento](#)
- [OPS11-BP01 Definizione di un processo per il miglioramento continuo](#)

Documenti correlati:

- [Whitepaper AWS - Introduction to DevOps on AWS](#)
- [AWS Whitepaper - Cloud AWS Adoption Framework: Operations Perspective](#)
- [AWS Well-Architected Framework Operational Excellence - Workload level Operating model topologies](#)
- [AWS Prescriptive Guidance - Building your Cloud Operating Model](#)
- [AWS Prescriptive Guidance - Create a RACI or RASCI matrix for a cloud operating model](#)
- [Cloud AWS Operations & Migrations Blog - Delivering Business Value with Cloud Platform Teams](#)
- [Cloud AWS Operations & Migrations Blog - Why a Cloud Operating Model?](#)
- [AWS DevOps Blog - How organizations are modernizing for cloud operations](#)

Video correlati:

- [AWS Summit Online - Cloud Operating Models for Accelerated Transformation](#)
- [AWS re:Invent 2023 - Future-proofing cloud security: A new operating model](#)

## OPS02-BP05 Definizione di meccanismi per richiedere aggiunte, modifiche ed eccezioni

È possibile effettuare richieste ai proprietari di processi, procedure e risorse. Tra le richieste figurano aggiunte, modifiche ed eccezioni. Tali richieste passano attraverso un processo di gestione delle modifiche. Prendi decisioni informate per approvare le richieste quando vengono ritenute fattibili e appropriate dopo una valutazione dei vantaggi e dei rischi.

Risultato desiderato:

- Puoi effettuare richieste per modificare processi, procedure e risorse sulla base della titolarità assegnata.
- Le modifiche vengono eseguite in modo deliberato, valutando benefici e rischi.

Anti-pattern comuni:

- Devi aggiornare il modo di implementare la tua applicazione, ma non esiste un metodo per richiedere una modifica al processo di implementazione al team operativo.
- Il piano di ripristino di emergenza deve essere aggiornato, ma non è stato identificato il proprietario a cui richiedere le modifiche.

Vantaggi dell'adozione di questa best practice:

- Processi, procedure e risorse possono evolvere mentre cambiano i requisiti.
- I titolari possono prendere decisioni mirate su quando effettuare le modifiche.
- Le modifiche vengono eseguite deliberatamente.

Livello di rischio associato se questa best practice non fosse adottata: medio

Guida all'implementazione

Per implementare questa best practice devi essere in grado di richiedere modifiche a processi, procedure e risorse. Il processo di gestione delle modifiche può essere semplice. Documenta il processo di gestione delle modifiche.

Esempio del cliente

AnyCompany Retail usa una matrice di assegnazione delle responsabilità (RACI) per identificare il proprietario delle modifiche per processi, procedure e risorse. L'azienda dispone di un processo documentato di gestione delle modifiche, semplice e facile da seguire. Tramite il processo e la matrice RACI, tutti possono inviare richieste di modifiche.

Passaggi dell'implementazione

1. Identifica i processi, le procedure e le risorse per il tuo carico di lavoro e i proprietari di ciascun elemento. Documentali nel tuo sistema di gestione delle conoscenze.
  - a. Se non hai implementato [OPS02-BP01 Associazione di proprietari identificati alle risorse](#), [OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure](#) o [OPS02-BP03 Assegnazione di proprietari identificati alle operazioni che siano responsabili delle relative prestazioni](#), comincia per prima cosa da questi.
2. Collabora con le parti interessate all'interno della tua azienda per sviluppare un processo di gestione delle modifiche. Il processo deve includere aggiunte, modifiche ed eccezioni per risorse, processi e procedure.
  - a. Puoi utilizzare [AWS Systems Manager Change Manager](#) come piattaforma di gestione delle modifiche per le risorse dei carichi di lavoro.
3. Documenta il processo di gestione delle modifiche nel tuo sistema di gestione delle conoscenze.

Livello di impegno per il piano di implementazione: Medio. Sviluppare un processo di gestione delle modifiche significa garantire un allineamento con più parti interessate all'interno dell'organizzazione.

## Risorse

### Best practice correlate:

- [OPS02-BP01 Associazione di proprietari identificati alle risorse](#) - I titolari delle risorse devono essere identificati prima di definire un processo di gestione delle modifiche.
- [OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure](#) - I titolari dei processi devono essere identificati prima di definire un processo di gestione delle modifiche.
- [OPS02-BP03 Assegnazione di proprietari identificati alle operazioni che siano responsabili delle relative prestazioni](#) - I titolari delle attività operative devono essere identificati prima di definire un processo di gestione delle modifiche.

### Documenti correlati:

- [Prontuario AWS - Playbook di base per grandi migrazioni AWS: creazione di matrici RACI](#)
- [Whitepaper sulla Gestione delle modifiche nel cloud](#)

### Servizi correlati:

- [AWS Systems Manager Change Manager](#)

## OPS02-BP06 Predefinizione o negoziazione delle responsabilità tra i team

Fai in modo che esistano accordi definiti o negoziati tra i team che descrivono come funzionano e si supportano reciprocamente (ad esempio, tempi di risposta, obiettivi o contratti relativi al livello di servizio). I canali di comunicazione tra team sono documentati. Comprendere l'impatto del lavoro dei team sui risultati aziendali e sui risultati di altri team e organizzazioni fornisce indicazioni in merito alla priorità dei loro compiti e consente loro di rispondere in modo appropriato.

Quando la responsabilità e la proprietà sono indefinite o sconosciute, rischi sia di non affrontare le attività necessarie in modo tempestivo sia di impiegare sforzi ridondanti e potenzialmente conflittuali per rispondere a tali esigenze.

### Risultato desiderato:

- Il lavoro tra team o gli accordi di assistenza vengono concordati e documentati.
- I team che supportano o lavorano con altri hanno definito i canali di comunicazione e le aspettative in termini di risposte.

## Anti-pattern comuni:

- In produzione si verifica un problema e due team separati iniziano a cercare la soluzione senza confrontarsi. Il loro impegno separato prolunga l'interruzione.
- Il team operativo ha bisogno di assistenza dal team di sviluppo, ma non c'è un accordo sui tempi di risposta. La richiesta si blocca nel backlog.

## Vantaggi dell'adozione di questa best practice:

- I team sanno come interagire e supportarsi a vicenda.
- Le aspettative relative ai tempi di risposta sono note.
- I canali di comunicazione sono definiti in modo chiaro.

Livello di rischio associato se questa best practice non fosse adottata: basso

## Guida all'implementazione

Se si implementa questa best practice non ci saranno dubbi sulla collaborazione tra team. Gli accordi formali codificano il modo di collaborare o di assistersi a vicenda dei team. I canali di comunicazione tra team sono documentati.

## Esempio del cliente

Il team SRE di AnyCompany Retail ha un contratto sul livello di servizio (SLA) con il team di sviluppo. Ogni volta che il team di sviluppo effettua una richiesta nel sistema di ticketing, riceverà una risposta entro 15 minuti. Se si verifica un malfunzionamento presso la sede, il team SRE assume il comando delle indagini con il supporto del team di sviluppo.

## Passaggi dell'implementazione

1. Collaborando con le parti interessate all'interno dell'organizzazione, sviluppa accordi tra team basati su processi e procedure.
  - a. Se i due team condividono un processo o una procedura, crea un runbook su come i team devono collaborare.
  - b. Se esistono dipendenze tra i team, concorda uno SLA per le risposte alle richieste.
2. Inserisci le responsabilità nel tuo sistema di gestione delle conoscenze.

Livello di impegno per il piano di implementazione: Medio. Se non esistono accordi tra i team, può essere impegnativo raggiungere un accordo con le parti interessate all'interno dell'organizzazione.

## Risorse

### Best practice correlate:

- [OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure](#) - La titolarità dei processi deve essere stabilita prima di definire gli accordi tra i team.
- [OPS02-BP03 Assegnazione di proprietari identificati alle operazioni che siano responsabili delle relative prestazioni](#) - La titolarità delle attività operative deve essere stabilita prima di definire gli accordi tra i team.

### Documenti correlati:

- [AWS Executive Insights - Promuovere l'innovazione con i team da due pizze](#)
- [Introduzione a DevOps su AWS - I team da due pizze](#)

## Cultura organizzativa

Fornisci supporto ai membri del team in modo che possano essere più efficaci nell'azione e nel supporto dei risultati aziendali.

### Best practice

- [OPS03-BP01 Definizione della sponsorizzazione esecutiva](#)
- [OPS03-BP02 Potere di intervento dei membri del team quando i risultati sono a rischio](#)
- [OPS03-BP03 Incoraggiamento all'escalation](#)
- [OPS03-BP04 Comunicazioni tempestive, chiare e fruibili](#)
- [OPS03-BP05 Incoraggiamento alla sperimentazione](#)
- [OPS03-BP06 Incoraggiamento ai membri del team a mantenere e ampliare le proprie competenze](#)
- [OPS03-BP07 Fornitura di risorse appropriate ai team](#)

## OPS03-BP01 Definizione della sponsorizzazione esecutiva

Ai massimi livelli, gli alti dirigenti fungono da sponsor esecutivo per definire chiaramente le aspettative e la direzione dei risultati dell'organizzazione, compresa la valutazione del successo. Lo sponsor sostiene e promuove l'adozione delle best practice e l'evoluzione dell'organizzazione.

Risultato desiderato: le organizzazioni che si impegnano ad adottare, trasformare e ottimizzare le operazioni cloud stabiliscono chiare linee di leadership e responsabilità per i risultati desiderati.

L'organizzazione comprende ogni capacità richiesta per raggiungere un nuovo risultato e assegna la titolarità ai team funzionali per lo sviluppo. La leadership implementa attivamente questa direzione, assegna la titolarità, si assume la responsabilità e definisce il lavoro. Di conseguenza, le persone in tutta l'organizzazione possono mobilitarsi, sentirsi ispirate e lavorare attivamente per raggiungere gli obiettivi desiderati.

Anti-pattern comuni:

- I titolari dei carichi di lavoro sono tenuti a migrare i carichi di lavoro su AWS senza uno sponsor e un piano chiari per le operazioni cloud. I team pertanto non collaborano consapevolmente per migliorare e consolidare le proprie capacità operative. La mancanza di standard operativi sulle best practice mette in difficoltà i team, ad esempio il lavoro degli operatori, le chiamate e il debito tecnico, limitando l'innovazione.
- È stato fissato un nuovo obiettivo a livello di organizzazione per adottare una tecnologia emergente senza fornire sponsor e strategia di leadership. I team interpretano gli obiettivi in modo diverso, il che crea confusione su dove concentrare gli impegni, sul perché sono importanti e su come misurare l'impatto. Di conseguenza, l'organizzazione perde slancio nell'adozione della tecnologia.

Vantaggi dell'adozione di questa best practice: quando lo sponsor esecutivo comunica e condivide chiaramente visione, direzione e obiettivi, i membri del team sanno cosa devono fare. Quando i leader sono coinvolti attivamente, le persone e i team iniziano a concentrare attivamente gli impegni nella stessa direzione per raggiungere gli obiettivi definiti. L'organizzazione di conseguenza massimizza la capacità di successo. Quando si valuta il successo, è possibile identificare meglio gli ostacoli al suo conseguimento in modo che possano essere affrontati attraverso l'intervento dello sponsor esecutivo.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

- In ogni fase del percorso verso il cloud (migrazione, adozione oppure ottimizzazione), il successo richiede un coinvolgimento attivo ai massimi livelli della leadership con uno sponsor esecutivo designato. Lo sponsor esecutivo allinea la mentalità, le competenze e le modalità di lavoro del team alla strategia definita.
- Spiega il perché: fai chiarezza e spiega il ragionamento alla base della visione e della strategia.
- Stabilisci le aspettative: definisci e pubblica gli obiettivi delle organizzazioni, incluso il modo in cui vengono misurati i progressi e il successo.
- Monitora il raggiungimento degli obiettivi: misura periodicamente il raggiungimento incrementale degli obiettivi, non solo il completamento delle attività. Condividi i risultati in modo da poter intraprendere le azioni appropriate se si evidenziano dei rischi.
- Fornisci le risorse necessarie per raggiungere gli obiettivi: favorisci la collaborazione tra persone e team per creare le soluzioni giuste che portino ai risultati definiti. Ciò riduce o elimina gli attriti organizzativi.
- Sostegno ai team: mantieni un coinvolgimento attivo con i tuoi team in modo da comprendere come stanno e se ci sono fattori esterni che li influenzano. Individua gli ostacoli che impediscono l'avanzamento dei team. Agisci per conto dei tuoi team per superare gli ostacoli e rimuovere gli oneri superflui. Quando i team sono influenzati da fattori esterni, rivaluta gli obiettivi e modifica i target in base alle esigenze.
- Promuovi l'adozione delle best practice: riconosci le best practice che offrono vantaggi quantificabili e identifica creatori e destinatari. Incoraggia ulteriormente l'adozione per amplificare i vantaggi ottenuti.
- Incoraggia l'evoluzione dei team: crea una cultura del miglioramento continuo e apprendi in modo proattivo dai progressi compiuti e dagli errori. Incoraggia la crescita e lo sviluppo sia personale sia organizzativo. Usa dati e aneddoti per migliorare la visione e la strategia.

### Esempio del cliente

AnyCompany Retail è in fase di trasformazione aziendale attraverso il rapido rinnovamento delle esperienze dei clienti, il miglioramento della produttività e l'accelerazione della crescita con l'IA generativa.

## Passaggi dell'implementazione

1. Stabilisci una leadership a thread singolo e assegna uno sponsor esecutivo principale per guidare e gestire la trasformazione.
2. Definisci chiaramente i risultati aziendali della trasformazione e assegna titolarità e responsabilità. Fornisci allo sponsor esecutivo principale l'autorità di guidare e prendere decisioni critiche.
3. Verifica che la strategia di trasformazione sia stata definita molto chiaramente e sia stata ampiamente comunicata dallo sponsor esecutivo a tutti i livelli dell'organizzazione.
  - a. Definisci chiaramente gli obiettivi aziendali per le iniziative IT e cloud.
  - b. Documenta le principali metriche aziendali per promuovere la trasformazione dell'IT e del cloud.
  - c. Comunica la visione in modo coerente a tutti i team e alle persone responsabili di parti della strategia.
4. Sviluppa matrici di pianificazione della comunicazione che specifichino quale messaggio deve essere recapitato a leader, manager e singoli collaboratori specifici. Specifica la persona o il team che deve recapitare questo messaggio.
  - a. Rispetta i piani di comunicazione in modo coerente e affidabile.
  - b. Stabilisci e gestisci le aspettative attraverso eventi di persona su base regolare.
  - c. Accetta il feedback sull'efficacia delle comunicazioni, quindi modifica le comunicazioni e pianifica di conseguenza.
  - d. Pianifica gli eventi di comunicazione per comprendere in modo proattivo le sfide dei team e stabilire un ciclo di feedback coerente che consenta di correggere la direzione laddove necessario.
5. Coinvolgi attivamente ogni iniziativa dal punto di vista della leadership per verificare che tutti i team interessati comprendano i risultati di cui sono responsabili.
6. In ogni riunione sullo stato, gli sponsor esecutivi devono individuare gli ostacoli, esaminare metriche, aneddoti o feedback dei team nonché misurare i progressi verso il raggiungimento degli obiettivi.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS03-BP04 Comunicazioni tempestive, chiare e fruibili](#)

- [OPS11-BP01 Definizione di un processo per il miglioramento continuo](#)
- [OPS11-BP07 Revisione dei parametri delle operazioni](#)

Documenti correlati:

- [Untangling Your Organisational Hairball: Highly Aligned](#)
- [The Living Transformation: Pragmatically approaching changes](#)
- [Becoming a Future-Ready Enterprise](#)
- [7 Pitfalls to Avoid When Building a CCOE](#)
- [Navigating the Cloud: Key Performance Indicators for Success](#)

Video correlati:

- [AWS re:Invent 2023: A leader's guide to generative AI: Using history to shape the future \(SEG204\)](#)

Esempi correlati:

- [Prosci: Primary Sponsor's Role & Importance](#)

## OPS03-BP02 Potere di intervento dei membri del team quando i risultati sono a rischio

Il comportamento culturale della responsabilità instillato dalla leadership fa sì che ogni dipendente si senta autorizzato ad agire per conto dell'intera azienda, al di là del proprio ambito definito da ruolo e responsabilità. I dipendenti possono intervenire per identificare in modo proattivo i rischi man mano che emergono e intraprendere le azioni appropriate. Tale cultura consente ai dipendenti di prendere decisioni di alto valore in quanto sono consapevoli della situazione.

Ad esempio, Amazon utilizza i [principi di leadership](#) come linee guida per favorire il comportamento desiderato dei dipendenti per andare a gestire situazioni complesse, risolvere i problemi, affrontare i conflitti e intraprendere le azioni adeguate.

Risultato desiderato: la leadership ha stabilito una nuova cultura che consente a persone e team di prendere decisioni critiche, anche ai livelli più bassi dell'organizzazione, purché tali decisioni siano definite con autorizzazioni e meccanismi di sicurezza verificabili. L'errore non è una mancanza, i team imparano in modo iterativo a migliorare il processo decisionale e le risposte per affrontare situazioni

simili in futuro. Se le azioni già intraprese portano a un miglioramento che può avvantaggiare altri team, occorre condividere in modo proattivo le conoscenze derivanti da tali azioni. La leadership misura i miglioramenti operativi e incentiva le persone e l'organizzazione all'adozione di tali modelli.

Anti-pattern comuni:

- Nell'organizzazione non esistono linee guida o meccanismi chiari su cosa fare quando viene identificato un rischio. Ad esempio, quando un dipendente nota un attacco di phishing, non lo segnala al team di sicurezza, con il risultato che gran parte dell'organizzazione è vittima dell'attacco, causando una violazione dei dati.
- I clienti si lamentano dell'indisponibilità del servizio, che deriva principalmente da implementazioni non riuscite. Il team SRE è responsabile dello strumento di implementazione e il rollback automatico per le implementazioni è nella roadmap a lungo termine. In un recente rollout dell'applicazione, uno degli ingegneri ha fornito una soluzione per automatizzare il ripristino dell'applicazione a una versione precedente. Sebbene la soluzione possa diventare il modello per i team SRE, altri team non la adottano perché non esiste un processo per monitorare i miglioramenti. L'organizzazione continua a ottenere implementazioni non corrette che hanno un impatto sui clienti e provocano ulteriore insoddisfazione.
- Per rispettare la conformità, il team di infosec supervisiona un processo consolidato per ruotare regolarmente le chiavi SSH condivise per conto degli operatori che si connettono alle loro istanze Linux Amazon EC2. I team di infosec impiegano diversi giorni per completare la rotazione delle chiavi e la connessione alle istanze viene bloccata. Nessuno all'interno o all'esterno di infosec suggerisce di utilizzare altre opzioni su AWS per ottenere lo stesso risultato.

Vantaggi dell'adozione di questa best practice: decentralizzando l'autorità del potere decisionale e consentendo ai team di prendere le decisioni chiave, è possibile risolvere i problemi più rapidamente con tassi di successo sempre crescenti. Inoltre, i team iniziano a percepire un senso di appartenenza e gli errori sono accettabili. La sperimentazione diventa un pilastro culturale. I manager e i direttori non si sentono controllati in ogni aspetto del loro lavoro.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

1. Sviluppa una cultura in cui si prevede che possano verificarsi errori.
2. Definisci chiaramente titolarità e responsabilità per le varie aree funzionali all'interno dell'organizzazione.

3. Comunica la titolarità e la responsabilità a tutti in modo che le persone sappiano chi può facilitare le decisioni decentralizzate.
4. Stabilisci le decisioni definitive e reversibili per permettere alle persone di sapere quando è necessario eseguire l'escalation a livelli più alti di leadership.
5. Crea la consapevolezza organizzativa che tutti i dipendenti hanno la capacità di agire a vari livelli quando i risultati sono a rischio. Fornisci ai membri del team la documentazione sulla governance, i livelli di autorizzazione, gli strumenti e le opportunità per mettere in pratica le competenze necessarie e intervenire in modo efficace.
6. Offri ai membri del team l'opportunità di mettere in pratica le competenze necessarie per rispondere a varie decisioni. Una volta definiti i livelli decisionali, organizza delle giornate di gioco per verificare che tutti i singoli collaboratori comprendano e possano usare il processo.
  - a. Fornisci ambienti sicuri alternativi in cui testare i processi e sottoporre i membri del team alla dovuta formazione.
  - b. Riconosci e crea la consapevolezza che i membri del team hanno l'autorità di agire quando il risultato ha un livello di rischio prestabilito.
  - c. Definisci l'autorità dei membri del team per intervenire assegnando le autorizzazioni e l'accesso ai carichi di lavoro e ai componenti supportati.
7. Offri ai team la possibilità di condividere le proprie conoscenze (successi e fallimenti operativi).
8. Consenti ai team di sfidare lo status quo e fornisci i meccanismi per monitorare e misurare i miglioramenti, nonché il loro impatto sull'organizzazione.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS01-BP06 Valutazione dei compromessi gestendo vantaggi e rischi](#)
- [OPS02-BP05 Definizione di meccanismi per identificare responsabilità e titolarità](#)

Documenti correlati:

- [AWS Blog Post | The agile enterprise](#)
- [AWS Blog Post | Measuring success : A paradox and a plan](#)
- [AWS Blog Post | Letting go : Enabling autonomy in teams](#)

- [Centralize or Decentralize?](#)

## Video correlati:

- [re:Invent 2023 | How to not sabotage your transformation \(SEG201\)](#)
- [re:Invent 2021 | Amazon Builders' Library: Operational Excellence at Amazon](#)
- [Centralization vs. Decentralization](#)

## Esempi correlati:

- [Using architectural decision records to streamline technical decision-making for a software development project](#)

## OPS03-BP03 Incoraggiamento all'escalation

I membri del team sono incoraggiati dalla leadership a segnalare problemi e preoccupazioni ai responsabili delle decisioni e alle parti interessate di alto livello se ritengono che i risultati desiderati siano a rischio e gli standard previsti non siano rispettati. Questa è una funzionalità della cultura dell'organizzazione ed è implementata a tutti i livelli. L'escalation deve essere eseguita in anticipo e di frequente, in modo che i rischi possano essere identificati e limitati prima che provochino incidenti. La leadership non rimprovera le persone per aver effettuato l'escalation di un problema.

Risultato desiderato: le persone in tutta l'organizzazione si sentono a proprio agio nell'eseguire l'escalation dei problemi ai livelli di leadership immediati e più elevati. La leadership ha stabilito deliberatamente e consapevolmente l'aspettativa che i propri team si sentano tranquilli nell'eseguire l'escalation di qualsiasi problema. Esiste un meccanismo per eseguire l'escalation dei problemi a ogni livello dell'organizzazione. Quando un dipendente esegue l'escalation al proprio manager, insieme decidono il livello di impatto e se il problema debba essere ulteriormente scalato. Per iniziare l'escalation, i dipendenti sono tenuti a includere un piano di lavoro consigliato per risolvere il problema. Se la direzione non interviene tempestivamente, i dipendenti sono incoraggiati a inoltrare i problemi al massimo livello di leadership se ritengono fermamente che i rischi per l'organizzazione giustifichino l'escalation.

## Anti-pattern comuni:

- I dirigenti non pongono domande approfondite durante la riunione sullo stato del programma di trasformazione del cloud per scoprire dove si verificano problemi e ostacoli. Solo le buone notizie

vengono presentate nello stato. Il CIO ha chiarito che le piacciono solo le buone notizie, poiché qualsiasi sfida sollevata fa pensare che il programma non stia andando bene.

- Sei un ingegnere delle operazioni cloud e noti che il nuovo sistema di gestione delle conoscenze non è ampiamente adottato dai team applicativi. L'azienda ha investito un anno di tempo e diversi milioni di dollari per implementare questo nuovo sistema di gestione delle conoscenze, ma le persone continuano a creare i propri runbook localmente e a condividerli su una condivisione cloud aziendale, rendendo difficile il recupero delle conoscenze pertinenti ai carichi di lavoro supportati. Cerchi di portare questo aspetto all'attenzione della dirigenza perché l'uso coerente del sistema può migliorare l'efficienza operativa. Quando lo comunichi alla direttrice a capo dell'implementazione del sistema di gestione delle conoscenze, ti rimprovera perché mette in discussione l'investimento.
- Il team di infosec responsabile del potenziamento delle risorse di calcolo ha deciso di mettere in atto un processo che richiede l'esecuzione delle scansioni necessarie per garantire che le istanze EC2 siano completamente protette prima che il team di calcolo rilasci la risorsa per l'uso. Ciò ha comportato il ritardo di una settimana per l'implementazione delle risorse e il mancato rispetto dello SLA. Il team di calcolo non desidera inoltrare la questione al vicepresidente tramite cloud perché ciò mette in cattiva luce il vicepresidente della sicurezza delle informazioni.

Vantaggi dell'adozione di questa best practice:

I problemi complessi o critici vengono risolti prima che abbiano impatto sull'azienda. Si perde meno tempo. I rischi sono ridotti al minimo. I team diventano più proattivi e concentrati sui risultati della risoluzione dei problemi.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

La volontà e la capacità di crescere liberamente a tutti i livelli dell'organizzazione sono la base organizzativa e culturale che deve essere sviluppata consapevolmente attraverso una formazione appropriata, le comunicazioni della leadership, la definizione delle aspettative e l'implementazione di meccanismi a tutti i livelli dell'organizzazione.

### Passaggi dell'implementazione

1. Definisci policy, standard e aspettative per l'organizzazione.
  1. Garantisci un'ampia adozione e comprensione delle policy, delle aspettative e degli standard.

2. Incoraggia, forma e responsabilizza i lavoratori a eseguire un'escalation anticipata e frequente quando gli standard non vengono rispettati.
3. Riconosci a livello organizzativo che l'escalation anticipata e frequente è la best practice. Accetti che le escalation possono rivelarsi infondate e che è meglio avere l'opportunità di prevenire un incidente piuttosto che privarsi di quell'opportunità senza escalation.
  - a. Crea un meccanismo per l'escalation, ad esempio un [sistema Andon Cord](#).
  - b. È opportuno disporre di procedure documentate che definiscano quando e come deve verificarsi l'escalation.
  - c. Definisci la serie di persone in ordine di autorità cui è consentito intraprendere o approvare azioni, nonché le informazioni di contatto di ciascuna parte interessata.
4. Un'escalation deve continuare fino a quando il membro del team non è convinto che il rischio sia stato mitigato attraverso le azioni guidate dalla leadership.
  - a. Le escalation devono includere:
    - i. la descrizione della situazione e la natura del rischio;
    - ii. le criticità della situazione;
    - iii. chi o cosa è interessato;
    - iv. il livello dell'impatto;
    - v. l'urgenza in caso di impatto;
    - vi. i rimedi suggeriti e i piani di mitigazione.
  - b. Proteggi i dipendenti coinvolti nell'escalation. È necessario predisporre una policy che protegga i membri del team da eventuali penalità se si trovano a dover scavalcare una parte interessata o un responsabile delle decisioni non reattivo. Metti in atto dei meccanismi per identificare se ciò si verifica e rispondere in modo appropriato.
5. Incoraggia la cultura del miglioramento continuo e dei cicli di feedback in tutto ciò che l'organizzazione produce. I cicli di feedback fungono da piccole escalation per le persone responsabili e identificano le opportunità di miglioramento, anche quando l'escalation non è necessaria. La cultura del miglioramento continuo obbliga tutti a essere più proattivi.
6. La leadership deve periodicamente ribadire le policy, gli standard, i meccanismi e il desiderio di un'escalation aperta e di cicli di feedback continui senza penalità.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS02-BP05 Definizione di meccanismi per richiedere aggiunte, modifiche ed eccezioni](#)

Documenti correlati:

- [How do you foster a culture of continuous improvement and learning from Andon and escalation systems?](#)
- [The Andon Cord \(IT Revolution\)](#)
- [AWS DevOps Guidance | Establish clear escalation paths and encourage constructive disagreement](#)

Video correlati:

- [Jeff Bezos on how to make decisions \(& increase velocity\)](#)
- [Toyota Product System: Stopping Production, a Button, and an Andon Electric Board](#)
- [Andon Cord in LEAN Manufacturing](#)

Esempi correlati:

- [Working with escalation plans in Incident Manager](#)

## OPS03-BP04 Comunicazioni tempestive, chiare e fruibili

La leadership è responsabile della creazione di comunicazioni forti ed efficaci, soprattutto quando l'organizzazione adotta nuove strategie, tecnologie o modalità di lavoro. I leader devono stabilire le aspettative affinché tutto il personale lavori per raggiungere gli obiettivi aziendali. Elabora meccanismi di comunicazione che creino e mantengano la consapevolezza tra i team responsabili della gestione dei piani finanziati e sponsorizzati dalla leadership. Utilizza la diversità interorganizzativa e ascolta con attenzione i molteplici punti di vista. Usa questa prospettiva per incrementare l'innovazione, mettere in discussione le tue ipotesi e ridurre il rischio di bias confermativi. Favorisci l'inclusione, la diversità e l'accessibilità all'interno dei team per ottenere prospettive vantaggiose.

Risultato desiderato: l'organizzazione progetta strategie di comunicazione per affrontare l'impatto del cambiamento sull'organizzazione. I team sono informati e motivati a continuare a lavorare

insieme anziché l'uno contro l'altro. Le persone comprendono quanto sia importante il proprio ruolo per raggiungere gli obiettivi stabiliti. L'e-mail è solo un meccanismo passivo per le comunicazioni e viene utilizzato di conseguenza. La direzione trascorre del tempo con i singoli collaboratori per aiutarli a comprendere le proprie responsabilità, le attività da completare e in che modo il loro lavoro contribuisce alla missione generale. Quando necessario, i leader coinvolgono direttamente le persone in un ambiente più piccolo per trasmettere il messaggio e verificare che venga recepito in modo efficace. Come risultato di buone strategie di comunicazione, l'organizzazione si comporta in misura pari o superiore alle aspettative della leadership. La leadership incoraggia e desidera esaminare opinioni diverse all'interno dell'organizzazione e tra i team.

Anti-pattern comuni:

- L'organizzazione ha un piano quinquennale per migrare tutti i carichi di lavoro su AWS. Il business case per il cloud include la modernizzazione del 25% di tutti i carichi di lavoro per utilizzare la tecnologia serverless. Il CIO comunica questa strategia ai collaboratori diretti e si aspetta che ogni leader trasmetta questa presentazione a manager, direttori e singoli collaboratori senza comunicazioni di persona. Il CIO fa un passo indietro e si aspetta che l'organizzazione esegua la nuova strategia.
- La leadership non fornisce né utilizza un meccanismo di feedback e aumenta il divario nelle aspettative, causando ostacoli per i progetti.
- Ti viene chiesto di apportare una modifica ai gruppi di sicurezza, ma non ti vengono forniti i dettagli sulle modifiche da apportare, sull'impatto della modifica su tutti i carichi di lavoro e sulla data della modifica. Il manager inoltra un'e-mail del vicepresidente di infosec e aggiunge il messaggio "Make this happen."
- Sono state apportate modifiche alla strategia di migrazione che riducono la percentuale di modernizzazione pianificata dal 25% al 10%. La riduzione ha effetti a valle sull'organizzazione delle operazioni. Questo cambiamento strategico non è stato comunicato e quindi non è disponibile la capacità qualificata sufficiente per supportare un numero maggiore di carichi di lavoro lift and shift AWS.

Vantaggi dell'adozione di questa best practice:

- L'organizzazione è ben informata sulle strategie nuove o modificate e agisce di conseguenza con una forte motivazione alla collaborazione per raggiungere gli obiettivi e le metriche generali stabiliti dalla leadership.

- Esistono meccanismi che vengono utilizzati per fornire tempestivamente notifiche ai membri del team in merito a rischi noti ed eventi pianificati.
- Le nuove modalità di lavoro, compresi i cambiamenti delle persone o dell'organizzazione, dei processi o della tecnologia, insieme alle competenze richieste, vengono adottate in modo più efficace dall'organizzazione che quindi realizza i vantaggi aziendali più rapidamente.
- I membri del team hanno il contesto necessario per ricevere le comunicazioni e possono essere più efficaci nel loro lavoro.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

Per implementare questa best practice devi collaborare con le parti interessate presenti nell'organizzazione per concordare gli standard di comunicazione. Comunica tali standard alla tua organizzazione. Per qualsiasi transizione IT significativa, il team di pianificazione definito può gestire con maggiore successo l'impatto del cambiamento sulle persone rispetto a un'organizzazione che ignora questa procedura. Le organizzazioni più grandi possono essere più impegnative nella gestione del cambiamento perché richiedono un forte consenso sulla nuova strategia di tutti i singoli collaboratori. In assenza di un team di pianificazione della transizione, la leadership ha il 100% della responsabilità di condurre comunicazioni efficaci. Quando si crea un team di pianificazione della transizione, comunica ai membri del team di collaborare con tutta la leadership organizzativa per definire e gestire comunicazioni efficaci a tutti i livelli.

### Esempio del cliente

AnyCompany Retail si è registrata al supporto Enterprise AWS e dipende da altri fornitori di terze parti per le operazioni cloud. L'azienda utilizza chat e chatop come principale mezzo di comunicazione per le attività operative. Allarmi e altre informazioni popolano canali specifici. Quando qualcuno deve intervenire, il risultato desiderato viene definito in modo chiaro e, in molti casi, la persona riceve un runbook o un playbook da usare. Usa un calendario delle modifiche per pianificare i cambiamenti più importanti ai sistemi di produzione.

### Passaggi dell'implementazione

1. Crea un team principale all'interno dell'organizzazione che abbia la responsabilità di elaborare e avviare i piani di comunicazione dei cambiamenti che avvengono a più livelli all'interno dell'organizzazione.

2. Istituisce la titolarità a thread singolo per la supervisione. Offri ai singoli team la capacità di innovare in modo indipendente e bilanciare l'uso di meccanismi coerenti, consentendo così il giusto livello di ispezione e visione della direzione.
3. Collabora con le parti interessate di tutta l'organizzazione per concordare standard, procedure e piani di comunicazione.
4. Verifica che il team di comunicazione principale collabori con la leadership dell'organizzazione e del programma per creare messaggi per il personale appropriato per conto dei leader.
5. Sviluppa meccanismi di comunicazione strategici per gestire il cambiamento attraverso annunci, calendari condivisi, riunioni plenarie e metodi di persona o individuali, in modo che i membri del team abbiano le giuste aspettative sulle azioni da intraprendere.
6. Quando possibile, fornisci il contesto, i dettagli e il tempo necessari per determinare se è richiesta un'azione. Quando è necessaria un'azione, fornisci l'azione richiesta e il suo impatto.
7. Implementa gli strumenti che facilitino le comunicazioni tattiche, come chat interna, e-mail e gestione delle conoscenze.
8. Implementa i meccanismi per misurare e verificare che tutte le comunicazioni portino ai risultati desiderati.
9. Stabilisci un ciclo di feedback che misuri l'efficacia delle comunicazioni, specialmente quando sono correlate alla resistenza ai cambiamenti nell'organizzazione.
- 10 Per tutti gli Account AWS, stabilisci [contatti alternativi](#) per la fatturazione, la sicurezza e le operazioni. Idealmente, ogni contatto deve essere una distribuzione di e-mail anziché una comunicazione individuale specifica.
- 11 Stabilisci un piano di comunicazione di escalation e annullamento dell'escalation per interagire con i team interni ed esterni, compreso il supporto AWS e altri fornitori di terze parti.
- 12 Avvia ed esegui le strategie di comunicazione in modo coerente per tutta la durata di ciascun programma di trasformazione.
- 13 Assegna le priorità alle azioni ripetibili, ove possibile, per automatizzarle in sicurezza su larga scala.
- 14 Quando le comunicazioni sono richieste in scenari con azioni automatizzate, lo scopo della comunicazione deve essere quello di informare i team, per il controllo o una parte del processo di gestione delle modifiche.
- 15 Analizza le comunicazioni provenienti dai sistemi di avviso per individuare i falsi positivi o gli avvisi che vengono creati costantemente. Rimuovi o modifica questi avvisi in modo che vengano inviati quando è richiesto l'intervento umano. Se viene attivato un avviso, fornisci un runbook o un playbook.

- a. Puoi usare [AWS Systems Manager Documents](#) per creare playbook e runbook per gli avvisi.
16. Sono stati attivati meccanismi per fornire tempestivamente notifiche in merito ai rischi o agli eventi pianificati in modo chiaro e fruibile al fine di consentire risposte appropriate. Usa elenchi di indirizzi e-mail o canali di chat per inviare le notifiche di preavviso rispetto agli eventi pianificati.
- a. [AWS Chatbot](#) può essere utilizzato per inviare allarmi e rispondere agli eventi all'interno della piattaforma di messaggistica dell'organizzazione.
17. Fornisci una fonte di informazioni accessibile dove è possibile individuare gli eventi pianificati. Fornisci le notifiche degli eventi pianificati dallo stesso sistema.
- a. [Il calendario delle modifiche di AWS Systems Manager](#) può essere usato per creare sessioni in cui possono verificarsi le variazioni. In questo modo i membri del team ricevono un preavviso su quando poter effettuare la modifica in modo sicuro.
18. Monitora le notifiche di vulnerabilità e le informazioni sulle patch per comprendere le vulnerabilità nell'ambiente sregolato e i potenziali rischi associati ai componenti del carico di lavoro. Invia notifiche ai membri del team in modo che possano intervenire.
- a. Puoi iscriverti ai [Bollettini sulla sicurezza AWS](#) per ricevere notifiche sulla vulnerabilità di AWS.
19. Cerca opinioni e prospettive diverse: incoraggia la condivisione dei contributi da parte di tutti. Offri opportunità di comunicazione ai gruppi sottorappresentati. Distribuisci a rotazione i ruoli e le responsabilità nelle riunioni.
- a. Amplia ruoli e responsabilità: offri ai membri del team l'opportunità di assumere ruoli che altrimenti potrebbero non ricoprire mai. Ciò consentirà loro di acquisire esperienza e nuove prospettive grazie anche alle interazioni con i nuovi membri del team, con i quali potrebbero non interagire altrimenti. Un mutuo scambio di esperienze e punti di vista vantaggioso per tutti. Con l'aumento delle prospettive, identifica le opportunità aziendali emergenti o le nuove opportunità di miglioramento. Fai in modo che i membri di un team svolgano a turno attività comuni eseguite normalmente da altri affinché comprendano le richieste e l'impatto delle loro prestazioni.
  - b. Garantisci un ambiente sicuro e ospitale: adotta policy e controlli che consentano di proteggere la sicurezza fisica e mentale dei membri del team all'interno dell'organizzazione. I membri del team devono essere in grado di interagire senza alcun timore. Quando i membri del team si sentono al sicuro e ben accolti, è più probabile che siano coinvolti e produttivi. Più è diversificata la tua organizzazione, migliore sarà la comprensione nei confronti delle persone supportate, inclusi i clienti. Quando i membri del team si sentono a loro agio, sono liberi di parlare e sono sicuri che verranno ascoltati, con maggiori probabilità condivideranno approfondimenti preziosi (ad esempio, opportunità di marketing, esigenze di accessibilità, segmenti di mercato non serviti, rischi non riconosciuti nel tuo ambiente).

- c. Incoraggia la totale partecipazione dei membri del team: fornisci le risorse necessarie ai dipendenti affinché partecipino appieno a tutte le attività correlate al lavoro. I membri del team che affrontano sfide quotidiane hanno sviluppato competenze per superarle. Queste competenze esclusive possono offrire vantaggi significativi alla tua organizzazione. Grazie al supporto di strutture adeguate, i membri del team possono apportare contributi vantaggiosi.

## Risorse

### Best practice correlate:

- [OPS03-BP01 Definizione della sponsorizzazione esecutiva](#)
- [OPS07-BP03 Utilizzo di runbook per eseguire le procedure](#)
- [OPS07-BP04 Utilizzo dei playbook per analizzare i problemi](#)

### Documenti correlati:

- [AWS Blog post | Accountability and empowerment are key to high-performing agile organizations](#)
- [AWS Executive Insights | Learn to scale innovation, not complexity | Single-threaded Leaders](#)
- [Bollettini sulla sicurezza AWS](#)
- [Open CVE](#)
- [AWS Support App in Slack to Manage Support Cases](#)
- [Manage AWS resources in your Slack channels with AWS Chatbot](#)

### Esempi correlati:

- [Well-Architected Labs: Inventario e gestione delle patch \(Livello 100\)](#)

### Servizi correlati:

- [AWS Chatbot](#)
- [AWS Systems Manager Change Calendar](#)
- [AWS Systems Manager Documents](#)

## OPS03-BP05 Incoraggiamento alla sperimentazione

La sperimentazione è un catalizzatore per trasformare nuove idee in prodotti e funzionalità. La sperimentazione accelera l'apprendimento e mantiene acceso l'interesse e il coinvolgimento dei membri del team. I membri del team sono incoraggiati a sperimentare spesso per promuovere l'innovazione. Anche quando si verifica un risultato indesiderato, è comunque utile sapere quello che non bisogna fare. I membri del team non vengono puniti per gli esperimenti riusciti con risultati indesiderati.

Risultato desiderato:

- La tua organizzazione incoraggia la sperimentazione per promuovere l'innovazione.
- Gli esperimenti sono utilizzati come un'opportunità per imparare.

Anti-pattern comuni:

- Vuoi eseguire un test A/B, ma non esiste un meccanismo per eseguire l'esperimento. Distribuisce una modifica all'interfaccia utente senza la possibilità di testarla. Questo comporta un'esperienza cliente negativa.
- La tua azienda ha solo un ambiente di test e uno di produzione. Non esiste un ambiente di sperimentazione (sandbox) in cui provare nuove funzionalità o prodotti, per cui le sperimentazioni vengono effettuate all'interno dell'ambiente di produzione.

Vantaggi dell'adozione di questa best practice:

- La sperimentazione incoraggia l'innovazione.
- Grazie alla sperimentazione puoi reagire più velocemente al feedback degli utenti.
- La tua organizzazione sviluppa una cultura dell'apprendimento.

Livello di rischio associato se questa best practice non fosse adottata: medio

### Guida all'implementazione

Le sperimentazioni devono essere eseguite in modo sicuro. Sfrutta più ambienti per sperimentare senza mettere a rischio le risorse di produzione. Usa il test A/B e le flag delle funzionalità per testare gli esperimenti. Offri ai membri del team la possibilità di eseguire esperimenti in un ambiente di sperimentazione (sandbox).

## Esempio del cliente

AnyCompany Retail incoraggia la sperimentazione. I membri del team possono dedicare il 20% della propria settimana lavorativa alla sperimentazione o all'apprendimento di nuove tecnologie. Hanno a disposizione un ambiente di sperimentazione (sandbox) in cui possono innovare. Il test A/B viene utilizzato per nuove funzionalità che possono essere così convalidate con il feedback di utenti reali.

## Passaggi dell'implementazione

1. Lavora con la direzione della tua organizzazione per supportare la sperimentazione. I membri del team devono essere incoraggiati a eseguire esperimenti in modo sicuro.
2. Offri ai membri del team un ambiente in cui possono sperimentare in modo sicuro. Devono avere accesso a un ambiente simile alla produzione.
  - a. Puoi usare un Account AWS separato per creare un ambiente di sperimentazione (sandbox). [AWS Control Tower](#) può essere usato per eseguire il provisioning di questi account.
3. Usa flag delle funzionalità e test A/B per sperimentare in modo sicuro e raccogliere il feedback degli utenti.
  - a. [AWS AppConfig Feature Flags](#) offre la possibilità di creare flag delle funzionalità.
  - b. [Amazon CloudWatch Evidently](#) può essere utilizzato per eseguire test A/B per un'implementazione limitata.
  - c. Puoi usare le [versioni AWS Lambda](#) per implementare una nuova versione di una funzionalità per il test beta.

Livello di impegno per il piano di implementazione: alto Offrire ai membri del team un ambiente in cui sperimentare in modo sicuro può richiedere investimenti significativi. Potresti anche aver bisogno di modificare il codice dell'applicazione per usare flag di funzionalità o supportare il test A/B.

## Risorse

Best practice correlate:

- [OPS11-BP02 Esecuzione di analisi post-incidente](#) - Imparare dagli incidenti è un fattore importante di innovazione e sperimentazione.
- [OPS11-BP03 Implementazione di cicli di feedback](#) - I cicli di feedback sono una parte importante della sperimentazione.

Documenti correlati:

- [Uno sguardo approfondito alla cultura di Amazon: Sperimentazione, Fallimento e Ossessione per il cliente](#)
- [Best practice per creare e gestire account per un ambiente di sperimentazione \(sandbox\) in AWS](#)
- [Creare una cultura della sperimentazione abilitata dal Cloud](#)
- [Promuovere sperimentazione e innovazione nel cloud presso SulAmérica Seguros](#)
- [Sperimenta con più frequenza, sbaglia di meno](#)
- [Organizzazione dell'ambiente AWS con l'utilizzo di account multipli - OU Sandbox](#)
- [Usare AWS AppConfig Feature Flags](#)

#### Video correlati:

- [AWS On Air con Amazon CloudWatch Evidently | Eventi AWS](#)
- [AWS On Air San Fran Summit 2022 con integrazione di AWS AppConfig Feature Flags con Jira](#)
- [AWS re:Invent 2022 - Un'implementazione non è un rilascio: controlla i tuoi rilasci con flag di funzionalità \(BOA305-R\)](#)
- [Creazione programmatica di un Account AWS con AWS Control Tower](#)
- [Impostazione di un ambiente AWS multi-account che utilizzi le best practice di AWS Organizations](#)

#### Esempi correlati:

- [AWS Innovation Sandbox](#)
- [Personalizzazione end-to-end 101 per l'e-commerce](#)

#### Servizi correlati:

- [Amazon CloudWatch Evidently](#)
- [AWS AppConfig](#)
- [AWS Control Tower](#)

## OPS03-BP06 Incoraggiamento ai membri del team a mantenere e ampliare le proprie competenze

I team devono aumentare le proprie competenze per adottare nuove tecnologie e supportare i cambiamenti di domanda e responsabilità a supporto dei carichi di lavoro. L'ampliamento delle competenze nelle nuove tecnologie è spesso fonte di soddisfazione per i membri del team e supporta l'innovazione. Incoraggia i membri del team a perseguire e mantenere le certificazioni di settore in modo da convalidare e riconoscere le loro crescenti competenze. Pratica la formazione trasversale per promuovere il trasferimento di conoscenze e ridurre il rischio di impatto significativo in caso di perdita di membri del team qualificati ed esperti con competenze a livello istituzionale. Fornisci tempo strutturato dedicato per l'apprendimento.

AWS offre le risorse, tra cui il [centro risorse per le nozioni di base di AWS](#), i [blog AWS](#), i [Tech talk online di AWS](#), [gli eventi e i webinar di AWS](#) e gli [AWS Well-Architected Labs](#), che forniscono indicazioni, esempi e procedure guidate dettagliate per formare i team.

Risorse come [AWS Support](#), ([AWS re:Post](#), [AWS Support Center](#)) e [AWS Documentation](#) aiutano a rimuovere gli ostacoli tecnici e a migliorare le operazioni. Se hai domande, contatta AWS Support tramite AWS Support Center.

Inoltre, AWS condivide le best practice e i modelli appresi attraverso la gestione di AWS nella [Amazon Builders' Library](#) e un'ampia gamma di altri materiali didattici utili tramite il [blog AWS](#) e il [podcast ufficiale di AWS](#).

[AWS Training and Certification](#) offre la formazione gratuita attraverso corsi digitali personalizzati, insieme a piani di apprendimento per ruolo o dominio. Per supportare ulteriormente lo sviluppo delle competenze AWS dei team, è anche possibile iscriversi a corsi di formazione con istruttore.

Risultato desiderato: l'organizzazione valuta costantemente le lacune nelle competenze e le colma con budget e investimenti strutturati. I team incoraggiano e incentivano i membri con attività di miglioramento delle competenze, come l'acquisizione delle principali certificazioni del settore. I team traggono beneficio da programmi dedicati alla condivisione incrociata delle conoscenze, come corsi di formazione in pausa pranzo, giornate di full immersion, hackathon e giornate di gioco. L'organizzazione mantiene i sistemi delle conoscenze aggiornati e pertinenti per la formazione incrociata dei membri dei team, compresi i corsi di formazione per l'onboarding dei nuovi assunti.

Anti-pattern comuni:

- In assenza di un programma di formazione strutturato e di un budget, i team riscontrano difficoltà nel tentativo di tenere il passo con l'evoluzione della tecnologia, il che si traduce in un aumento dell'attrito.
- Nell'ambito della migrazione ad AWS, l'organizzazione dimostra lacune nelle competenze e una padronanza del cloud variabile tra i team. Senza un impegno per il miglioramento delle competenze, i team si ritrovano oberati di attività di gestione legacy e inefficienti dell'ambiente cloud, causando un aumento del lavoro degli operatori. Questo stato di esaurimento dei team aumenta l'insoddisfazione dei dipendenti.

Vantaggi dell'adozione di questa best practice: quando un'organizzazione investe consapevolmente nel miglioramento delle competenze dei team, aiuta anche ad accelerare e scalare l'adozione e l'ottimizzazione del cloud. I programmi di apprendimento mirati favoriscono l'innovazione e creano capacità operative per consentire ai team di essere preparati a gestire gli eventi. I team investono consapevolmente nell'implementazione e nell'evoluzione delle best practice. Il morale dei team è alto e i membri apprezzano il contributo che offrono all'azienda.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Per adottare nuove tecnologie, promuovere l'innovazione e stare al passo con i cambiamenti della domanda e delle responsabilità a supporto dei carichi di lavoro, investi continuamente nella crescita professionale dei team.

### Passaggi dell'implementazione

1. Utilizza programmi strutturati di sostegno per il cloud: [AWS Skills Guild](#) offre una formazione di consulenza per aumentare la fiducia nelle competenze cloud e promuovere la cultura dell'apprendimento continuo.
2. Metti a disposizione le risorse per la formazione: metti a disposizione tempo strutturato dedicato, accesso ai materiali di formazione, risorse di laboratorio e supporto alla partecipazione a conferenze e organizzazioni professionali che offrono opportunità di apprendimento da docenti e colleghi. Offri ai membri dei team junior la possibilità di contattare i membri dei team senior affinché questi fungano da mentori o possano mostrare loro come lavorano trasmettendo metodi e competenze consolidati. Incoraggia l'apprendimento dei contenuti non direttamente correlati al lavoro per avere una prospettiva più ampia.
3. Incoraggia l'uso di risorse tecniche esperte: utilizza risorse come [AWS re:Post](#) per accedere a conoscenze curate e a un'intraprendente community.

4. Crea e mantieni aggiornato l'archivio delle conoscenze: utilizza le piattaforme di condivisione delle conoscenze, come wiki e runbook. Crea la tua fonte di conoscenze specialistiche riutilizzabile con [AWS re:Post Private](#) per semplificare la collaborazione, migliorare la produttività e accelerare l'onboarding dei dipendenti.
5. Formazione del team e coinvolgimento tra team: pianifica le esigenze di formazione continua dei membri del tuo team. Offri loro l'opportunità di unirsi ad altri team (temporaneamente o definitivamente) per condividere competenze e best practice a beneficio dell'intera organizzazione.
6. Supporta il perseguimento e il mantenimento delle certificazioni di settore: favorisci l'acquisizione e il mantenimento da parte dei membri del tuo team di certificazioni di settore che convalidano le loro conoscenze e riconoscono i loro risultati.

Livello di impegno per il piano di deployment: elevato

## Risorse

Best practice correlate:

- [OPS03-BP01 Definizione della sponsorizzazione esecutiva](#)
- [OPS11-BP04 Gestione delle conoscenze](#)

Documenti correlati:

- [AWS Whitepaper | Cloud Adoption Framework: People Perspective](#)
- [Investing in continuous learning to grow your organization's future](#)
- [AWS Skills Guild](#)
- [AWS Training and Certification](#)
- [AWS Support](#)
- [AWS re:Post](#)
- [AWS Getting Started Resource Center](#)
- [AWS Blogs](#)
- [Conformità di Cloud AWS](#)
- [AWS Documentation](#)
- [The Official AWS Podcast.](#)
- [AWS Online Tech Talks](#)

- [Eventi e webinar AWS](#)
- [AWS Well-Architected Labs](#)
- [The Amazon Builders' Library](#)

Video correlati:

- [AWS re:Invent 2023 | Reskilling at the speed of cloud: Turning employees into entrepreneurs](#)
- [WS re:Invent 2023 | Building a culture of curiosity through gamification](#)

## OPS03-BP07 Fornitura di risorse appropriate ai team

Stabilisci il giusto numero di membri competenti del team e gli strumenti e le risorse per supportare le esigenze di carico di lavoro. Il sovraccarico dei membri del team aumenta il rischio di errore umano. Gli investimenti in strumenti e risorse, come l'automazione, possono aumentare l'efficacia del team consentendogli di supportare un numero maggiore di carichi di lavoro senza richiedere capacità aggiuntiva.

Risultato desiderato:

- Hai istituito un team con il personale necessario e le competenze richieste per gestire i carichi di lavoro in AWS in conformità al piano di migrazione. Man mano che il team si è ampliato nel corso del progetto di migrazione, ha acquisito competenza nelle tecnologie AWS di base che l'azienda intende utilizzare per la migrazione o la modernizzazione delle applicazioni.
- Hai preparato con attenzione il piano per i membri del team per fare un uso efficiente delle risorse, sfruttando l'automazione e il flusso di lavoro. Un team più piccolo può ora gestire più infrastrutture per conto dei team di sviluppo delle applicazioni.
- Con il cambiamento delle priorità operative, qualsiasi vincolo di risorse viene identificato in modo proattivo per proteggere il successo delle iniziative aziendali.
- Le metriche che segnalano le difficoltà operative, ad esempio l'affaticamento da chiamata o il paging eccessivo, vengono esaminate per verificare che il personale non sia sovraccaricato.

Anti-pattern comuni:

- Il personale non ha incrementato le competenze AWS all'approssimarsi del piano pluriennale di migrazione al cloud, comportando rischi per il supporto dei carichi di lavoro e abbassando il morale dei dipendenti.

- L'intera organizzazione IT adotta le modalità di lavoro agili. L'azienda assegna le priorità al portafoglio di prodotti e stabilisce le metriche per le funzionalità che devono essere sviluppate per prime. Il processo agile non richiede che i team assegnino story point ai piani di lavoro. Di conseguenza, è impossibile conoscere il livello di capacità richiesto per il successivo lavoro o se le competenze giuste sono state assegnate al lavoro.
- Un partner AWS migra i tuoi carichi di lavoro e non hai un piano di transizione di supporto per i team una volta completato il progetto di migrazione. I team hanno difficoltà a supportare i carichi di lavoro in modo efficiente ed efficace.

Vantaggi dell'adozione di questa best practice: nell'organizzazione sono disponibili membri dei team adeguatamente qualificati per supportare i carichi di lavoro. L'allocazione delle risorse può adattarsi al cambiamento delle priorità senza influire sulle prestazioni. Il risultato è che i team sono in grado di supportare i carichi di lavoro massimizzando al contempo il tempo per concentrarsi sull'innovazione per i clienti e aumentando a sua volta la soddisfazione dei dipendenti.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

La pianificazione delle risorse per la migrazione al cloud deve avvenire a un livello organizzativo in linea con il piano di migrazione e l'implementazione del modello operativo desiderato per supportare il nuovo ambiente cloud. Ciò deve includere la comprensione delle tecnologie cloud utilizzate per i team di sviluppo aziendale e delle applicazioni. La leadership dell'infrastruttura e delle operazioni deve pianificare l'analisi del divario delle competenze, la formazione e la definizione dei ruoli per gli ingegneri che guidano l'adozione del cloud.

### Passaggi dell'implementazione

1. Definisci i criteri per il successo dei team con metriche operative pertinenti, come la produttività del personale (ad esempio, i costi di supporto di un carico di lavoro o le ore spese dall'operatore per gli incidenti).
2. Definisci i meccanismi di pianificazione e ispezione della capacità delle risorse per verificare che il giusto equilibrio di capacità qualificata sia disponibile quando necessario e possa essere modificato nel tempo.
3. Crea i meccanismi, ad esempio inviando un sondaggio mensile ai team, per comprendere le sfide legate al lavoro che hanno un impatto sui team, come l'aumento delle responsabilità, i cambiamenti nella tecnologia, la mancanza di personale o l'aumento dei clienti supportati.

4. Utilizza questi meccanismi per interagire con i team e individuare le tendenze che possono contribuire alle sfide relative alla produttività dei dipendenti. Quando i team sono influenzati da fattori esterni, rivaluta gli obiettivi e modifica i target in base alle esigenze. Individua gli ostacoli che impediscono l'avanzamento dei team.
5. Verifica regolarmente se le risorse attualmente messe a disposizione sono ancora sufficienti o se devono essere aggiunte e apporta le modifiche appropriate ai team di supporto.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS03-BP06 Incoraggiamento ai membri del team a mantenere e ampliare le proprie competenze](#)
- [OPS09-BP03 Revisione delle metriche operative e assegnazione delle priorità per favorire il miglioramento](#)
- [OPS10-BP01 Utilizzo di un processo per la gestione di eventi, incidenti e problemi](#)
- [OPS10-BP07 Automazione delle risposte agli eventi](#)

Documenti correlati:

- [Cloud AWS Adoption Framework: People Perspective](#)
- [Becoming a Future-Ready Enterprise](#)
- [Prioritize your Employees' Skills to Drive Business Growth](#)
- [High performing organization - the Amazon Two-Pizza team](#)
- [How Cloud-Mature Enterprises Succeed](#)

# Preparazione

Per prepararti all'eccellenza operativa devi comprendere i carichi di lavoro e i loro comportamenti previsti. Sarai dunque in grado di progettare i carichi di lavoro in modo tale che forniscano informazioni sul loro stato e di creare le procedure per supportarli adeguatamente.

Per prepararti all'eccellenza operativa, è necessario eseguire le seguenti operazioni:

## Argomenti

- [Implementa l'osservabilità](#)
- [Progettazione delle operazioni](#)
- [Riduzione dei rischi di distribuzione](#)
- [Prontezza operativa e gestione delle modifiche](#)

## Implementa l'osservabilità

Implementare l'osservabilità nel carico di lavoro ti permette di comprendere lo stato di quest'ultimo e di adottare decisioni basate sui dati e che riflettono i requisiti aziendali.

L'osservabilità va oltre il semplice monitoraggio, in quanto fornisce una comprensione completa del funzionamento interno di un sistema basata sui suoi output esterni. L'osservabilità è legata a doppio filo a metriche, log e tracce per offrire informazioni approfondite sul comportamento e sulle dinamiche del sistema. Grazie a un'osservabilità efficace, i team possono distinguere modelli, anomalie e tendenze, così da essere in grado di affrontare in modo proattivo potenziali problemi e mantenere l'integrità del sistema.

L'identificazione degli indicatori chiave di prestazione (KPI) è fondamentale per garantire l'allineamento tra le attività di monitoraggio e gli obiettivi aziendali. Questo allineamento garantisce che i team prendano decisioni basate sui dati e su metriche realmente importanti, ottimizzando sia le prestazioni del sistema sia i risultati aziendali.

Inoltre, l'osservabilità consente alle aziende di essere proattive anziché reattive. I team possono comprendere le relazioni causa-effetto all'interno dei loro sistemi, prevedendo e prevenendo i problemi anziché limitarsi a reagire quando si verificano. Con l'evolversi dei carichi di lavoro, è essenziale riesaminare e perfezionare la strategia di osservabilità, assicurandosi che rimanga pertinente ed efficace.

## Best practice

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS04-BP02 Implementazione della telemetria dell'applicazione](#)
- [OPS04-BP03 Implementazione della telemetria dell'esperienza utente](#)
- [OPS04-BP04 Implementazione della telemetria delle dipendenze](#)
- [OPS04-BP05 Implementazione del tracciamento distribuito](#)

## OPS04-BP01 Identificazione degli indicatori chiave di prestazione

L'implementazione dell'osservabilità nel carico di lavoro inizia con la comprensione del suo stato e l'adozione di decisioni basate sui dati che riflettono i requisiti aziendali. Uno dei modi più efficaci per garantire l'allineamento tra le attività di monitoraggio e gli obiettivi aziendali è definire e monitorare gli indicatori chiave di prestazione (KPI).

Risultato desiderato: Pratiche di osservabilità efficienti e strettamente allineate agli obiettivi aziendali garantiscono che le attività di monitoraggio siano sempre al servizio di risultati aziendali tangibili.

Anti-pattern comuni:

- KPI non definiti: lavorare senza KPI chiari può portare ad attività di monitoraggio eccessive o insufficienti e alla perdita di segnali vitali.
- KPI statici: non riesaminare od ottimizzare i KPI man mano che il carico di lavoro o gli obiettivi aziendali si evolvono.
- Disallineamento: concentrarsi su metriche tecniche non direttamente correlate ai risultati aziendali o che sono più difficili da correlare ai problemi del mondo reale.

Vantaggi dell'adozione di questa best practice:

- Facilità di identificazione dei problemi: i KPI aziendali spesso evidenziano i problemi in modo più chiaro rispetto alle metriche tecniche. Un valore di un KPI aziendale che diminuisce permette di individuare un problema in modo più efficace rispetto alla valutazione di numerose metriche tecniche.
- Allineamento aziendale: assicura che le attività di monitoraggio supportino direttamente gli obiettivi aziendali.
- Efficienza: viene data la priorità alle risorse di monitoraggio e al focus sulle metriche che contano.
- Proattività: riconoscere e risolvere i problemi prima che abbiano implicazioni aziendali più ampie.

Livello di rischio associato se questa best practice non fosse adottata: alto

## Guida all'implementazione

Per definire in modo efficace i KPI del carico di lavoro:

1. Inizia con i risultati aziendali: prima di approfondire le metriche, comprendi i risultati aziendali desiderati. È stato rilevato un aumento delle vendite, un maggiore coinvolgimento degli utenti o tempi di risposta più rapidi?
2. Correla le metriche tecniche con gli obiettivi aziendali: non tutte le metriche tecniche hanno un impatto diretto sui risultati aziendali. Identifica quelli che hanno un impatto, anche se spesso è più immediato individuare un problema utilizzando un KPI aziendale.
3. Utilizza [Amazon CloudWatch](#): Utilizza CloudWatch per definire e monitorare le metriche che rappresentano i tuoi KPI.
4. Rivedi e aggiorna regolarmente i KPI: man mano che il carico di lavoro e la tua attività si evolvono, mantieni la pertinenza dei tuoi KPI.
5. Coinvolgi gli stakeholder: coinvolgi i team tecnici e aziendali nella definizione e revisione dei KPI.

Livello di impegno per il piano di implementazione: Medio

## Risorse

Best practice correlate:

- [the section called “OPS04-BP02 Implementazione della telemetria dell'applicazione”](#)
- [the section called “OPS04-BP03 Implementazione della telemetria dell'esperienza utente”](#)
- [the section called “OPS04-BP04 Implementazione della telemetria delle dipendenze”](#)
- [the section called “OPS04-BP05 Implementazione del tracciamento distribuito”](#)

Documenti correlati:

- [AWS Observability Best Practices](#)
- [CloudWatch User Guide](#)
- [AWS Observability Skill Builder Course](#)

Video correlati:

- [Developing an observability strategy](#)

Esempi correlati:

- [One Observability Workshop](#)

## OPS04-BP02 Implementazione della telemetria dell'applicazione

La telemetria dell'applicazione è la base su cui si fonda l'osservabilità del carico di lavoro. È fondamentale emettere dati di telemetria che offrano approfondimenti utili sullo stato dell'applicazione e sul raggiungimento degli obiettivi sia tecnici sia aziendali. Dalla risoluzione dei problemi alla misurazione dell'impatto di una nuova funzionalità fino all'allineamento con gli indicatori di prestazione chiave (KPI), la telemetria dell'applicazione garantisce informazioni su cui basare la creazione, il funzionamento e l'evoluzione del carico di lavoro.

Metriche, log e tracce costituiscono i tre pilastri principali dell'osservabilità. Questi operano come strumenti diagnostici che descrivono lo stato dell'applicazione. Nel tempo, aiutano a creare criteri di base e a identificare le anomalie. Tuttavia, per garantire l'allineamento tra le attività di monitoraggio e gli obiettivi aziendali, è fondamentale definire e monitorare i KPI. I KPI aziendali spesso facilitano l'identificazione dei problemi rispetto alle sole metriche tecniche.

Altri tipi di telemetria, come il monitoraggio degli utenti reali (RUM) e le transazioni sintetiche, completano queste origini dati primarie. Il RUM offre approfondimenti sulle interazioni degli utenti in tempo reale, mentre le transazioni sintetiche simulano i potenziali comportamenti degli utenti, aiutando a rilevare i colli di bottiglia prima che vengano riscontrati dagli utenti reali.

Risultato desiderato: ottieni approfondimenti utili sulle prestazioni del tuo carico di lavoro. Questi approfondimenti consentono di prendere decisioni proattive sull'ottimizzazione delle prestazioni, ottenere una maggiore stabilità del carico di lavoro, semplificare i processi CI/CD e utilizzare le risorse in modo efficace.

Anti-pattern comuni:

- **Osservabilità incompleta:** si trascura di incorporare l'osservabilità a ogni livello del carico di lavoro, con conseguenti punti ciechi che possono nascondere le prestazioni vitali del sistema e gli approfondimenti sul comportamento.
- **Visualizzazione frammentata dei dati:** quando i dati sono sparsi su più strumenti e sistemi, diventa difficile mantenere una visione olistica dello stato e delle prestazioni del carico di lavoro.

- Problemi segnalati dagli utenti: un segno della mancanza di un rilevamento proattivo dei problemi tramite telemetria e monitoraggio dei KPI aziendali.

Vantaggi dell'adozione di questa best practice:

- Processo decisionale informato: con gli approfondimenti ricavati dalla telemetria e dai KPI aziendali, puoi prendere decisioni basate sui dati.
- Migliore efficienza operativa: l'utilizzo delle risorse basato sui dati porta a un miglioramento dell'efficienza risparmiando sui costi.
- Maggiore stabilità del carico di lavoro: rilevamento e risoluzione più rapidi dei problemi con conseguente aumento dei tempi di attività.
- Processi CI/CD semplificati: gli approfondimenti ricavati dai dati di telemetria facilitano il perfezionamento dei processi e la distribuzione affidabile del codice.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

Per implementare la telemetria delle applicazioni per il carico di lavoro, utilizza i servizi AWS come [Amazon CloudWatch](#) e [AWS X-Ray](#). Amazon CloudWatch fornisce una suite completa di strumenti di monitoraggio, che consente di osservare le risorse e le applicazioni in ambienti AWS e on-premises. Raccoglie, tiene traccia e analizza le metriche, consolida e monitora i dati di log e risponde alle modifiche che interessano le risorse, migliorando la comprensione del funzionamento del carico di lavoro. Integrato con altri servizi, AWS X-Ray consente di tenere traccia, analizzare ed eseguire il debug delle applicazioni, offrendoti una comprensione approfondita del comportamento del tuo carico di lavoro. Grazie a funzionalità come mappe dei servizi, distribuzioni di latenza e tempistiche di tracciamento, AWS X-Ray fornisce approfondimenti sulle prestazioni del carico di lavoro e sui colli di bottiglia che lo interessano.

### Passaggi dell'implementazione

1. Identifica quali dati raccogliere: definisci le metriche, i log e le tracce essenziali che potrebbero offrire importanti approfondimenti sullo stato, le prestazioni e il comportamento del tuo carico di lavoro.
2. Implementa [l'agente CloudWatch](#): l'agente CloudWatch è fondamentale nel fornire metriche di sistema e dell'applicazione e log dal carico di lavoro e dall'infrastruttura sottostante. L'agente

CloudWatch può essere utilizzato anche per raccogliere tracce OpenTelemetry o X-Ray e inviarle a X-Ray.

3. Implementa il rilevamento delle anomalie per log e metriche: utilizza il [rilevamento delle anomalie CloudWatch Logs](#) e il [rilevamento delle anomalie delle metriche CloudWatch](#) per identificare automaticamente attività insolite nelle operazioni dell'applicazione. Questi strumenti utilizzano algoritmi di machine learning per rilevare e comunicare le anomalie, migliorando le capacità di monitoraggio e accelerando i tempi di risposta a potenziali interruzioni o minacce alla sicurezza. Configura queste funzionalità per gestire in modo proattivo lo stato e la sicurezza delle applicazioni.
4. Proteggi i dati di log sensibili: utilizza la [protezione dei dati Amazon CloudWatch Logs](#) per mascherare le informazioni sensibili all'interno dei log. Questa funzionalità aiuta a mantenere la privacy e la conformità con il rilevamento e il mascheramento automatici dei dati sensibili prima dell'accesso. Implementa il mascheramento dei dati per gestire e proteggere in modo sicuro i dettagli sensibili come le informazioni di identificazione personale (PII).
5. Definisci e monitora i KPI aziendali: stabilisci [metriche personalizzate](#) in linea con i [risultati aziendali](#).
6. Strumenta la tua applicazione con AWS X-Ray: oltre a implementare l'agente CloudWatch, è fondamentale [strumentare l'applicazione](#) per generare dati di tracciamento. Questo processo può fornire ulteriori approfondimenti sul comportamento e sulle prestazioni del carico di lavoro.
7. Standardizza la raccolta dei dati nell'applicazione: standardizza le procedure di raccolta dei dati nell'applicazione. L'uniformità aiuta a correlare e analizzare i dati, fornendo una visione completa del comportamento dell'applicazione.
8. Implementa l'osservabilità su più account: migliora l'efficienza del monitoraggio su più account Account AWS con [l'osservabilità su più account Amazon CloudWatch](#). Con questa funzionalità, puoi consolidare metriche, log e allarmi di diversi account in un'unica visualizzazione, semplificando la gestione e migliorando i tempi di risposta per i problemi identificati nell'ambiente AWS dell'organizzazione.
9. Analizza e agisci sui dati: una volta in atto la raccolta e la normalizzazione dei dati, utilizza [Amazon CloudWatch](#) per l'analisi di metriche e log e [AWS X-Ray](#) per l'analisi delle tracce. Tale analisi può fornire approfondimenti cruciali sullo stato, le prestazioni e il comportamento del carico di lavoro, guidando il processo decisionale.

Livello di impegno per il piano di deployment: elevato

## Risorse

Best practice correlate:

- [OPS04-BP01 Definizione dei KPI del carico di lavoro](#)
- [OPS04-BP03 Implementazione della telemetria dell'attività degli utenti](#)
- [OPS04-BP04 Implementazione della telemetria delle dipendenze](#)
- [OPS04-BP05 Implementazione della tracciabilità delle transazioni](#)

Documenti correlati:

- [AWS Observability Best Practices](#)
- [CloudWatch User Guide](#)
- [AWS X-Ray Developer Guide](#)
- [Strumentazione di sistemi distribuiti per visibilità operativa](#)
- [AWS Observability Skill Builder Course](#)
- [Quali sono le novità di Amazon CloudWatch](#)
- [Quali sono le novità di AWS X-Ray](#)

Video correlati:

- [AWS re:Invent 2022 - Observability best practices at Amazon](#)
- [AWS re:Invent 2022 - Developing an observability strategy](#)

Esempi correlati:

- [One Observability Workshop](#)
- [AWS Solutions Library: Application Monitoring with Amazon CloudWatch](#)

## OPS04-BP03 Implementazione della telemetria dell'esperienza utente

Acquisire informazioni approfondite sulle esperienze dei clienti e sulle interazioni con la tua applicazione è fondamentale. Il monitoraggio dell'utente reale (RUM) e le transazioni sintetiche sono strumenti molto efficaci per questo scopo. RUM fornisce dati sulle interazioni degli utenti reali, garantendo una prospettiva non filtrata della soddisfazione degli utenti, mentre le transazioni

sintetiche simulano le interazioni degli utenti, aiutando a rilevare potenziali problemi prima che essi abbiano un impatto sugli utenti reali.

Risultato desiderato: Una visione olistica dell'esperienza del cliente, il rilevamento proattivo dei problemi e l'ottimizzazione delle interazioni degli utenti per offrire esperienze digitali fluide.

Anti-pattern comuni:

- Applicazioni senza monitoraggio dell'utente reale (RUM):
  - rilevamento ritardato dei problemi: senza RUM, potresti non accorgerti di rallentamenti o problemi di prestazioni fino a quando non ricevi lamentele da parte degli utenti. Questo approccio reattivo può causare insoddisfazione nei clienti.
  - Mancanza di informazioni sull'esperienza utente: non utilizzare RUM significa perdere dati cruciali che mostrano come gli utenti reali interagiscono con l'applicazione, il che limita la tua capacità di ottimizzare l'esperienza utente.
- Applicazioni senza transazioni sintetiche:
  - Casi limite trascurati: le transazioni sintetiche consentono di testare percorsi e funzioni che potrebbero non essere utilizzati frequentemente dagli utenti tipici, ma che sono fondamentali per determinate funzioni aziendali. Senza di esse, questi percorsi potrebbero non funzionare correttamente e passare inosservati.
  - Verifica della presenza di problemi quando l'applicazione non viene utilizzata: i test sintetici regolari possono simulare situazioni in cui gli utenti reali non interagiscono attivamente con l'applicazione, garantendo che il sistema funzioni sempre correttamente.

Vantaggi dell'adozione di questa best practice:

- Rilevamento proattivo dei problemi: identifica e risolvi i problemi potenziali prima che abbiano un impatto sugli utenti reali.
- Esperienza utente ottimizzata: grazie al suo feedback continuo, RUM aiuta a perfezionare e migliorare l'esperienza utente complessiva.
- Informazioni approfondite sulle prestazioni del dispositivo e del browser: scopri come si comporta la tua applicazione in vari dispositivi e browser e implementa ulteriori ottimizzazioni.
- Flussi di lavoro aziendali convalidati: transazioni sintetiche regolari assicurano che le funzionalità principali e i percorsi critici siano operativi ed efficienti in maniera costante.
- Prestazioni delle applicazioni migliorate: sfrutta le informazioni approfondite raccolte dai dati degli utenti reali per migliorare la reattività e l'affidabilità delle applicazioni.

Livello di rischio associato se questa best practice non fosse adottata: alto

## Guida all'implementazione

Per eseguire la telemetria delle attività degli utenti sfruttando RUM e le transazioni sintetiche, AWS offre servizi come [Amazon CloudWatch RUM](#) e [Amazon CloudWatch Synthetics](#). Metriche, log e tracce, insieme ai dati sulle attività degli utenti, forniscono una visione completa dello stato operativo dell'applicazione e dell'esperienza utente.

### Passaggi dell'implementazione

1. Implementa Amazon CloudWatch RUM: integra la tua applicazione con CloudWatch RUM per raccogliere, analizzare e presentare dati relativi agli utenti reali.
  - a. Utilizza [la libreria JavaScript CloudWatch RUM](#) per integrare RUM con la tua applicazione.
  - b. Configura dashboard per visualizzare e monitorare i dati relativi agli utenti reali.
2. Configura CloudWatch Synthetics: crea canary o routine con script che simulano le interazioni degli utenti con la tua applicazione.
  - a. Definisci i flussi di lavoro e i percorsi critici delle applicazioni.
  - b. Progetta canary utilizzando [script di CloudWatch Synthetics](#) per simulare le interazioni degli utenti per questi percorsi.
  - c. Pianifica e monitora i canary affinché si attivino a intervalli specifici, in modo da garantire controlli costanti delle prestazioni.
3. Analizza e intervieni sui dati: Utilizza i dati provenienti da RUM e transazioni sintetiche per ottenere informazioni e adottare misure correttive quando vengono rilevate anomalie. Usa dashboard CloudWatch e allarmi per ottenere informazioni costanti.

Livello di impegno per il piano di implementazione: Medio

## Risorse

Best practice correlate:

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS04-BP02 Implementazione della telemetria dell'applicazione](#)
- [OPS04-BP04 Implementazione della telemetria delle dipendenze](#)
- [OPS04-BP05 Implementazione del tracciamento distribuito](#)

## Documenti correlati:

- [Amazon CloudWatch RUM Guide](#)
- [Amazon CloudWatch Synthetics Guide](#)

## Video correlati:

- [Optimize applications through end user insights with Amazon CloudWatch RUM](#)
- [AWS on Air ft. Real-User Monitoring for Amazon CloudWatch](#)

## Esempi correlati:

- [One Observability Workshop](#)
- [Repository Git per client Web Amazon CloudWatch RUM](#)
- [Using Amazon CloudWatch Synthetics to measure page load time](#)

## OPS04-BP04 Implementazione della telemetria delle dipendenze

La telemetria delle dipendenze è essenziale per monitorare lo stato e le prestazioni dei servizi e dei componenti esterni su cui si basa il carico di lavoro. Fornisce preziosi approfondimenti su reperibilità, timeout e altri eventi critici correlati alle dipendenze come DNS, database o API di terze parti.

Dotando l'applicazione di strumenti per generare metriche, log e tracce relative a queste dipendenze, acquisisci una comprensione più chiara dei potenziali colli di bottiglia, problemi di prestazioni o errori che potrebbero influire sul carico di lavoro.

Risultato desiderato: le dipendenze su cui si basa il carico di lavoro funzionano come previsto, consentendo di gestire i problemi in modo proattivo e garantendo prestazioni ottimali del carico di lavoro.

### Anti-pattern comuni:

- Scarsa attenzione alle dipendenze esterne: il focus è rivolto esclusivamente alle metriche interne dell'applicazione, trascurando quelle legate alle dipendenze esterne.
- Mancanza di monitoraggio proattivo: si attende che si verifichino problemi anziché monitorare costantemente lo stato e le prestazioni delle dipendenze.
- Monitoraggio isolato in comparti: si utilizzano strumenti di monitoraggio multipli ed eterogenei che possono portare a visioni dello stato delle dipendenze frammentate e incoerenti.

Vantaggi dell'adozione di questa best practice:

- Maggiore affidabilità del carico di lavoro: viene garantito che le dipendenze esterne siano costantemente disponibili e funzionino in modo ottimale.
- Rilevamento e risoluzione dei problemi più rapidi: identificazione e risoluzione proattiva dei problemi relativi alle dipendenze prima che influiscano sul carico di lavoro.
- Visione completa: acquisizione di una visione olistica dei componenti interni ed esterni che influenzano lo stato del carico di lavoro.
- Scalabilità del carico di lavoro migliorata: grazie alla comprensione dei limiti di scalabilità e delle caratteristiche prestazionali delle dipendenze esterne.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

Implementa la telemetria delle dipendenze iniziando con l'identificazione dei servizi, dell'infrastruttura e dei processi da cui dipende il carico di lavoro. Esegui una valutazione quantitativa delle condizioni ottimali nelle quali tali dipendenze funzionano come previsto e poi determina quali dati sono necessari per misurarle. Con queste informazioni, puoi creare dashboard e avvisi che forniscono approfondimenti ai tuoi team operativi sullo stato di tali dipendenze. Usa gli strumenti AWS per scoprire e quantificare gli impatti quando le dipendenze non riescono a fornire le prestazioni necessarie. Rivedi costantemente la tua strategia per tenere conto dei cambiamenti relativi a priorità, obiettivi e alle informazioni dettagliate acquisite.

### Passaggi dell'implementazione

Per implementare efficacemente la telemetria delle dipendenze:

1. Identifica le dipendenze esterne: collabora con le parti interessate per individuare le dipendenze esterne sulle quali si basa il carico di lavoro. Le dipendenze esterne possono comprendere servizi come database esterni, API di terze parti, percorsi di connettività di rete verso altri ambienti e servizi DNS. Il primo passo verso un'efficace telemetria delle dipendenze è acquisire una comprensione totale di quali esse siano.
2. Sviluppa una strategia di monitoraggio: una volta acquisito un quadro chiaro delle dipendenze esterne, progetta una strategia di monitoraggio ad hoc per esse. Trovare la strategia giusta implica comprendere le criticità di tutte le dipendenze, il loro comportamento previsto e gli eventuali accordi od obiettivi sul livello di servizio associato (SLA o SLT). Imposta avvisi proattivi che ti informino riguardo a cambiamenti di stato o deviazioni delle prestazioni.

3. Usa il [monitoraggio della rete](#): utilizza il [monitoraggio di Internet](#) e il [monitoraggio della rete](#) che forniscono approfondimenti completi sulle condizioni globali di Internet e della rete. Questi strumenti consentono di comprendere e rispondere alle interruzioni, ai malfunzionamenti o al degrado delle prestazioni che influiscono sulle dipendenze esterne.
4. Non perdere alcun aggiornamento con [AWS Health Dashboard](#): fornisce avvisi e indicazioni per la correzione qualora AWS sia interessato da eventi che potrebbero influire sui servizi.
  - a. Monitora [gli eventi AWS Health con le regole Amazon EventBridge](#) o integra a livello di programmazione l'API AWS Health per automatizzare le azioni quando ricevi eventi AWS Health. Possono essere azioni generali, come l'invio di tutti i messaggi pianificati sugli eventi del ciclo di vita a un'interfaccia di chat, oppure azioni specifiche, come l'avvio di un flusso di lavoro in uno strumento di gestione dei servizi IT.
  - b. Se usi AWS Organizations, [aggrega gli eventi AWS Health](#) tra gli account.
5. Strumenta l'applicazione con [AWS X-Ray](#): AWS X-Ray fornisce approfondimenti sulle prestazioni delle applicazioni e delle relative dipendenze sottostanti. La tracciatura delle richieste dall'inizio alla fine ti permette di identificare colli di bottiglia o guasti nei servizi o nei componenti esterni su cui si basa l'applicazione.
6. Usa [Amazon DevOps Guru](#): questo servizio basato sul machine learning identifica i problemi operativi, prevede quando potrebbero verificarsi problemi critici e consiglia azioni specifiche da intraprendere. Fornisce un supporto prezioso per acquisire approfondimenti sulle dipendenze e assicurarsi che queste non siano la fonte di problemi operativi.
7. Monitora regolarmente: monitora le metriche e i log relativi alle dipendenze esterne in maniera costante. Imposta avvisi per comportamenti imprevisti o prestazioni ridotte.
8. Convalida dopo le modifiche: ogni volta che una dipendenza esterna è interessata da un aggiornamento o una modifica, convalidane le prestazioni e verifica che queste siano in linea con i requisiti dell'applicazione.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS04-BP01 Definizione dei KPI del carico di lavoro](#)
- [OPS04-BP02 Implementazione della telemetria dell'applicazione](#)
- [OPS04-BP03 Implementazione della telemetria dell'attività degli utenti](#)

- [OPS04-BP05 Implementazione della tracciabilità delle transazioni](#)
- [OP08-BP04 Creare avvisi fruibili](#)

Documenti correlati:

- [Amazon Personal AWS Health Dashboard User Guide](#)
- [AWS Internet Monitor User Guide](#)
- [AWS X-Ray Developer Guide](#)
- [AWS DevOps Guru User Guide](#)

Video correlati:

- [Visibility into how internet issues impact app performance](#)
- [Introduction to Amazon DevOps Guru](#)
- [Manage resource lifecycle events at scale with AWS Health](#)

Esempi correlati:

- [Gaining operational insights with AIOps using Amazon DevOps Guru](#)
- [AWS Health Aware](#)
- [Using Tag-Based Filtering to Manage AWS Health Monitoring and Alerting at Scale](#)

## OPS04-BP05 Implementazione del tracciamento distribuito

Il tracciamento distribuito offre un modo per monitorare e visualizzare le richieste mentre attraversano vari componenti di un sistema distribuito. Acquisendo i dati di tracciamento da più fonti e analizzandoli in una vista unificata, i team possono comprendere meglio il flusso delle richieste, in quali punti sono presenti colli di bottiglia e dove devono concentrare gli sforzi di ottimizzazione.

Risultato desiderato: Una visione olistica del flusso delle richieste nel tuo sistema distribuito, che ti permette di ottenere un debug preciso, prestazioni ottimizzate e migliori esperienze utente.

Anti-pattern comuni:

- **Strumentazione incoerente:** non tutti i servizi in un sistema distribuito sono dotati di strumentazione per il monitoraggio.

- Ignorare la latenza: concentrarsi solo sugli errori e non considerare la latenza o il graduale deterioramento delle prestazioni.

Vantaggi dell'adozione di questa best practice:

- Panoramica completa del sistema: visualizzazione dell'intero percorso delle richieste, dall'ingresso all'uscita.
- Debug avanzato: identificazione rapida dei punti in cui si verificano guasti o problemi di prestazioni.
- Esperienza utente migliorata: monitoraggio e ottimizzazione in base ai dati effettivi dell'utente, garantendo che il sistema soddisfi le esigenze del mondo reale.

Livello di rischio associato se questa best practice non fosse adottata: alto

## Guida all'implementazione

Inizia identificando tutti gli elementi del carico di lavoro che richiedono strumentazione. Una volta presi in considerazione tutti i componenti, sfrutta strumenti come AWS X-Ray e OpenTelemetry per raccogliere dati di tracciamento da analizzare con strumenti come X-Ray e Amazon CloudWatchServiceLens Map. Effettua revisioni periodiche insieme agli sviluppatori e integra queste discussioni con strumenti come Amazon DevOps Guru, X-Ray Analytics e X-Ray Insights per ottenere risultati più approfonditi. Imposta avvisi basati sui dati di tracciamento per notificare quando i risultati sono a rischio, come definito nel piano di monitoraggio del carico di lavoro.

### Passaggi dell'implementazione

Per implementare il tracciamento distribuito in modo efficace:

1. Adotta [AWS X-Ray](#): implementa X-Ray nella tua applicazione per ottenere informazioni dettagliate sul suo comportamento, comprenderne le prestazioni e individuare i punti critici. Utilizza X-Ray Insights per l'analisi automatica dei tracciamenti.
2. Dota i tuoi servizi di strumenti: verifica che tutti i servizi, dalle funzioni [AWS Lambda](#) alle [istanze EC2](#), siano in grado di inviare dati di tracciamento. Più servizi dotati di strumentazione, più chiara sarà la visione end-to-end.
3. Incorpora [il monitoraggio dell'utente reale tramite CloudWatch](#) e [il monitoraggio sintetico](#): integra il monitoraggio dell'utente reale (RUM) e il monitoraggio sintetico con X-Ray. Ciò ti consente di acquisire esperienze utenti del mondo reale e simulare le interazioni degli utenti per identificare potenziali problemi.

4. Utilizza [l'agente CloudWatch](#): l'agente può inviare dati di tracciamento da X-Ray o da OpenTelemetry, permettendoti di raccogliere informazioni più approfondite.
5. Utilizza [Amazon DevOps Guru](#): DevOps Guru utilizza dati provenienti da X-Ray, CloudWatch, AWS Config e AWS CloudTrail per fornire suggerimenti fruibili.
6. Analizza le tracce: esamina regolarmente i dati di tracciamento per individuare schemi, anomalie o colli di bottiglia che possono influire sulle prestazioni dell'applicazione.
7. Imposta avvisi: configura avvisi in [CloudWatch](#) per segnalare schemi insoliti o latenze prolungate, il che ti permette di effettuare una risoluzione proattiva dei problemi.
8. Miglioramento continuo: riesamina la tua strategia di tracciamento man mano che aggiungi o modifichi servizi per acquisire tutti i punti dati pertinenti.

Livello di impegno per il piano di implementazione: Medio

## Risorse

Best practice correlate:

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS04-BP02 Implementazione della telemetria dell'applicazione](#)
- [OPS04-BP03 Implementazione della telemetria dell'esperienza utente](#)
- [OPS04-BP04 Implementazione della telemetria delle dipendenze](#)

Documenti correlati:

- [Guida per gli sviluppatori AWS X-Ray](#)
- [Amazon CloudWatch agent User Guide](#)
- [Amazon DevOps Guru User Guide](#)

Video correlati:

- [Use AWS X-Ray Insights](#)
- [AWS on Air ft. Observability: Amazon CloudWatch and AWS X-Ray](#)

Esempi correlati:

- [Instrumenting your Application with AWS X-Ray](#)

## Progettazione delle operazioni

Adotta strategie che migliorino il flusso delle modifiche in produzione e che favoriscano la rifattorizzazione, il feedback veloce sulla qualità e la correzione di errori. Tali prassi accelerano l'ingresso in produzione delle modifiche vantaggiose, contengono i problemi che si sono diffusi e permettono una rapida identificazione e risoluzione dei problemi introdotti attraverso le attività di implementazione.

In AWS, puoi vedere il tuo carico di lavoro completo (applicazioni, infrastruttura, policy, governance e operazioni) in forma di codice. Tutti gli elementi possono essere definiti al suo interno e aggiornati tramite codice. Ciò significa che puoi applicare la medesima disciplina di progettazione che utilizzi per il codice dell'applicazione a ciascun elemento del tuo stack.

### Best practice

- [OPS05-BP01 Utilizzo del controllo delle versioni](#)
- [OPS05-BP02 Test e convalida delle modifiche](#)
- [OPS05-BP03 Utilizzo di sistemi di gestione delle configurazioni](#)
- [OPS05-BP04 Utilizzo di sistemi di gestione della compilazione e implementazione](#)
- [OPS05-BP05 Esecuzione della gestione delle patch](#)
- [OPS05-BP06 Condivisione degli standard di progettazione](#)
- [OPS05-BP07 Implementazione di prassi per migliorare la qualità del codice](#)
- [OPS05-BP08 Utilizzo di più ambienti](#)
- [OPS05-BP09 Applicazione di modifiche frequenti, minime e reversibili](#)
- [OPS05-BP10 Automazione completa dell'integrazione e della distribuzione](#)

### OPS05-BP01 Utilizzo del controllo delle versioni

Utilizza il controllo delle versioni per attivare il monitoraggio di modifiche e rilasci.

Molti servizi AWS offrono funzionalità di controllo delle versioni. Utilizza una revisione o un sistema di controllo del codice sorgente come [AWS CodeCommit](#) per gestire il codice e altri artefatti, come i modelli [AWS CloudFormation](#) controllati dalla versione della tua infrastruttura.

Risultato desiderato: I tuoi team collaborano alla gestione del codice. Una volta unito, il codice è coerente e nessuna modifica viene persa. Gli errori possono essere facilmente ripristinati mediante il corretto controllo delle versioni.

Anti-pattern comuni:

- Hai sviluppato e archiviato il codice sulla workstation. Si è verificato un errore di archiviazione non recuperabile sulla workstation e il codice è andato perso.
- Dopo aver sovrascritto il codice esistente con le modifiche, riavvii l'applicazione e non è più utilizzabile. Non è possibile ripristinare la modifica.
- Hai un blocco di scrittura su un file di report che deve essere modificato da altri utenti. Ti contattano per chiederti di smettere di utilizzarlo in modo che possano completare le loro attività.
- Il team di ricerca ha lavorato a un'analisi dettagliata che definisce il tuo lavoro futuro. Qualcuno ha salvato accidentalmente la lista della spesa nel report finale. Non puoi ripristinare la modifica e devi ricreare il report.

Vantaggi dell'adozione di questa best practice: Grazie alle funzionalità di controllo delle versioni, puoi ripristinare facilmente gli stati validi noti e le versioni precedenti e limitare il rischio di perdita degli asset.

Livello di rischio associato se questa best practice non fosse adottata: alto

## Guida all'implementazione

Mantieni gli asset in repository con controllo delle versioni. In questo modo si supporta il monitoraggio delle modifiche, la distribuzione di nuove versioni, il rilevamento delle modifiche apportate alle versioni esistenti e il ripristino delle versioni precedenti, ad esempio il rollback a uno stato corretto noto in caso di errore. Integra nelle tue procedure le funzionalità di controllo delle versioni dei sistemi di gestione delle configurazioni.

## Risorse

Best practice correlate:

- [OPS05-BP04 Utilizzo di sistemi di gestione della compilazione e implementazione](#)

Documenti correlati:

- [Che cos'è AWS CodeCommit?](#)

Video correlati:

- [Introduzione ad AWS CodeCommit](#)

## OPS05-BP02 Test e convalida delle modifiche

Ogni modifica apportata deve essere testata per evitare errori in produzione. Questa best practice si concentra sulla verifica delle modifiche dal controllo di versione alla creazione dell'artefatto. Oltre alle modifiche al codice dell'applicazione, i test dovrebbero includere l'infrastruttura, la configurazione, i controlli di sicurezza e le procedure operative. I test assumono molte forme, dai test unitari all'analisi dei componenti software (SCA). Spostando i test più a sinistra nel processo di integrazione e consegna del software ottieni una maggiore certezza della qualità degli artefatti.

L'organizzazione deve sviluppare standard di test per tutti gli artefatti software. I test automatizzati riducono la fatica ed evitano gli errori dei test manuali. I test manuali potrebbero essere necessari in alcuni casi. Gli sviluppatori devono avere accesso ai risultati dei test automatizzati per creare cicli di feedback che migliorino la qualità del software.

Risultato desiderato: le modifiche software vengono testate prima del rilascio. Gli sviluppatori hanno accesso ai risultati dei test e alle convalide. La tua organizzazione ha uno standard per i test che applica a tutte le modifiche software.

Anti-pattern comuni:

- Implementi una nuova modifica software senza test. Non funziona in produzione e genera un'interruzione.
- I nuovi gruppi di sicurezza vengono implementati con AWS CloudFormation senza essere testati in un ambiente di pre-produzione. I gruppi di sicurezza rendono la tua app irraggiungibile per i clienti.
- Un metodo viene modificato, ma non ci sono test di unità. Il software ha esito negativo quando viene implementato in produzione.

Vantaggi dell'adozione di questa best practice: si riduce la percentuale di modifiche non riuscite nelle implementazioni del software. La qualità del software viene migliorata. Gli sviluppatori hanno una maggiore consapevolezza della fattibilità del loro codice. Le policy di sicurezza possono essere implementate in maniera affidabile per supportare la conformità dell'organizzazione. Le modifiche all'infrastruttura, come gli aggiornamenti automatici delle politiche di scaling, vengono testate in anticipo per soddisfare le esigenze del traffico.

Livello di rischio associato se questa best practice non fosse adottata: Elevato

## Guida all'implementazione

I test vengono eseguiti su tutte le modifiche, dal codice dell'applicazione all'infrastruttura, come parte della pratica di integrazione continua. I risultati dei test vengono pubblicati in modo che gli sviluppatori abbiano un feedback rapido. La tua organizzazione ha uno standard per i test che applica a tutte le modifiche software.

Usa la potenza dell'IA generativa con Amazon Q Developer per migliorare la produttività degli sviluppatori e la qualità del codice. Amazon Q Developer include la generazione di suggerimenti di codice (basati su modelli linguistici di grandi dimensioni), la produzione di unit test (comprese le condizioni limite) e il miglioramento della sicurezza del codice tramite il rilevamento e la correzione delle vulnerabilità di sicurezza.

### Esempio del cliente

Nell'ambito della pipeline di integrazione continua, AnyCompany Retail esegue diversi tipi di test su tutti gli artefatti software. Praticano lo sviluppo guidato dai test, per cui tutto il software è dotato di test unitari. Una volta creato l'artefatto, eseguono test end-to-end. Al termine di questa prima serie di test, viene eseguita una scansione statica della sicurezza dell'applicazione, alla ricerca di vulnerabilità note. Gli sviluppatori ricevono messaggi al superamento di ciascun gate di test. Una volta completati tutti i test, l'artefatto software viene archiviato in un repository di artefatti.

### Passaggi dell'implementazione

1. Collaborare con le parti interessate dell'organizzazione per sviluppare uno standard di test per gli artefatti software. Quali test standard devono superare tutti gli artefatti? Ci sono requisiti di conformità o di governance che devono essere inclusi nella copertura dei test? Devi condurre test di qualità del codice? Quando i test sono terminati, chi deve esserne a conoscenza?
  1. L'[architettura di riferimento della pipeline di distribuzione AWS](#) contiene un elenco autorevole dei tipi di test che possono essere condotti su artefatti software come parte di una pipeline di integrazione.
2. Fornisci la tua applicazione dei test necessari in base allo standard di test del software. Ogni set di test deve essere completato in meno di dieci minuti. I test devono essere eseguiti come parte della pipeline di integrazione.
  - a. Usa [Amazon Q Developer](#), uno strumento di IA generativa che consente di creare casi di unit test (comprese le condizioni limite), generare funzioni utilizzando codice e commenti e implementare gli algoritmi noti.

- b. Usa [Amazon CodeGuru Reviewer](#) per testare il codice dell'applicazione e trovare eventuali difetti.
- c. Puoi usare [AWS CodeBuild](#) per condurre i test su artefatti software.
- d. [AWS CodePipeline](#) può organizzare i tuoi test software in una pipeline.

## Risorse

### Best practice correlate:

- [OPS05-BP01 Utilizzo del controllo delle versioni](#)
- [OPS05-BP06 Condivisione degli standard di progettazione](#)
- [OPS05-BP07 Implementazione di prassi per migliorare la qualità del codice](#)
- [OPS05-BP10 Automazione completa dell'integrazione e dell'implementazione](#)

### Documenti correlati:

- [Adopt a test-driven development approach](#)
- [Accelerate your Software Development Lifecycle with Amazon Q](#)
- [Amazon Q Developer, now generally available, includes previews of new capabilities to reimagine developer experience](#)
- [The Ultimate Cheat Sheet for Using Amazon Q Developer in Your IDE](#)
- [Shift-Left Workload, leveraging AI for Test Creation](#)
- [Amazon Q Developer Center](#)
- [10 ways to build applications faster with Amazon CodeWhisperer](#)
- [Looking beyond code coverage with Amazon CodeWhisperer](#)
- [Best Practices for Prompt Engineering with Amazon CodeWhisperer](#)
- [Automated AWS CloudFormation Testing Pipeline with TaskCat and CodePipeline](#)
- [Building end-to-end AWS DevSecOps CI/CD pipeline with open source SCA, SAST, and DAST tools](#)
- [Getting started with testing serverless applications](#)
- [My CI/CD pipeline is my release captain](#)
- [Practicing Continuous Integration and Continuous Delivery on AWS Whitepaper](#)

### Video correlati:

- [Implement an API with Amazon Q Developer Agent for Software Development](#)
- [Installing, Configuring, & Using Amazon Q Developer with JetBrains IDEs \(How-to\)](#)
- [Mastering the art of Amazon CodeWhisperer - YouTube playlist](#)
- [AWS re:Invent 2020: Testable infrastructure: Integration testing on AWS](#)
- [AWS Summit ANZ 2021 - Driving a test-first strategy with CDK and test driven development](#)
- [Testing Your Infrastructure as Code with AWS CDK](#)

### Risorse correlate:

- [Building applications using generative AI with Amazon CodeWhisperer](#)
- [Amazon CodeWhisperer Workshop](#)
- [AWS Deployment Pipeline Reference Architecture - Application](#)
- [AWS Kubernetes DevSecOps Pipeline](#)
- [Policy as Code Workshop – Test Driven Development](#)
- [Run unit tests for a Node.js application from GitHub by using AWS CodeBuild](#)
- [Use Serverspec for test-driven development of infrastructure code](#)

### Servizi correlati:

- [Amazon Q Developer](#)
- [Amazon CodeGuru Reviewer](#)
- [AWS CodeBuild](#)
- [AWS CodePipeline](#)

## OPS05-BP03 Utilizzo di sistemi di gestione delle configurazioni

L'utilizzo di sistemi di gestione delle configurazioni permette di effettuare modifiche alle stesse e tenerne traccia. Questi sistemi riducono gli errori causati dai processi manuali e il livello di impegno richiesto per la distribuzione delle modifiche.

Durante l'inizializzazione di una risorsa, la gestione delle configurazioni statiche consente di impostare valori che dovrebbero rimanere coerenti per tutta la vita utile della risorsa. Ne sono

alcuni esempi l'azione di configurare un server web o applicativo su un'istanza oppure di definire la configurazione di un servizio AWS nella [AWS Management Console](#) o tramite la [AWS CLI](#).

Al momento dell'inizializzazione, la gestione delle configurazioni dinamiche consente di impostare valori che possono cambiare nel corso della vita utile di una risorsa. Ad esempio è possibile impostare un interruttore funzionale in grado di attivare una funzionalità nel codice tramite una modifica della configurazione, oppure modificare il livello di dettaglio del log durante un incidente per acquisire un maggior numero di dati e cambiarlo in seguito per tornare al livello di dettaglio precedente, risparmiando così in numero di log e nei relativi costi.

In AWS, è possibile utilizzare [AWS Config](#) per monitorare in modo continuo le configurazioni delle risorse AWS [tra i diversi account e regioni](#). Questa soluzione aiuta a tenere traccia della cronologia delle configurazioni, a capire che effetto avrebbe la modifica di una configurazione sulle altre risorse e a verificarle rispetto alle configurazioni previste o desiderate tramite [Regole di AWS Config](#) e [pacchetti di conformità AWS Config](#).

Se sulle applicazioni in esecuzione su istanze Amazon EC2, AWS Lambda, container, funzioni serverless, applicazioni mobili o dispositivi IoT sono attive configurazioni dinamiche, è possibile utilizzare [AWS AppConfig](#) per configurarle, convalidarle, implementarle e monitorarle nei tuoi ambienti.

In AWS, puoi creare pipeline di integrazione continua/distribuzione continua (CI/CD) utilizzando servizi come gli [Strumenti per sviluppatori in AWS](#) (ad esempio: [AWS CodeCommit](#), [AWS CodeBuild](#), [AWS CodePipeline](#), [AWS CodeDeploy](#) e [AWS CodeStar](#)).

Risultato desiderato: Puoi configurare, convalidare e implementare come parte della tua pipeline di integrazione continua e di distribuzione continua (CI/CD). Esegui il monitoraggio per verificare che le configurazioni siano corrette. Ciò riduce al minimo l'impatto sugli utenti finali e sui clienti.

Anti-pattern comuni:

- Aggiorni manualmente la configurazione del server Web all'interno del parco istanze e un certo numero di server non risponde a causa di errori di aggiornamento.
- Aggiorni manualmente il parco istanze del server applicazioni nel corso di molte ore. L'incoerenza nella configurazione durante la modifica causa comportamenti imprevisti.
- Qualcuno ha aggiornato i tuoi gruppi di sicurezza e i server Web non sono più accessibili. Senza sapere cosa è stato modificato, dedichi molto tempo a esaminare il problema prolungando il tempo necessario per il ripristino.

- Avvii una configurazione di riproduzione in produzione tramite CI/CD senza una convalida. Esponi utenti e clienti a dati e servizi errati.

Vantaggi dell'adozione di questa best practice: L'adozione di sistemi di gestione della configurazione riduce il livello di impegno necessario per apportare e tenere traccia delle modifiche e la frequenza degli errori causati dalle procedure manuali. I sistemi di gestione della configurazione forniscono garanzie per quanto riguarda la governance, la conformità e i requisiti normativi.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

I sistemi di gestione della configurazione vengono utilizzati per tenere traccia e implementare le modifiche nelle configurazioni delle applicazioni e degli ambienti. I sistemi di gestione della configurazione vengono utilizzati anche per ridurre gli errori causati dai processi manuali, rendere le modifiche alla configurazione ripetibili e verificabili e per ridurre il livello di impegno.

### Passaggi dell'implementazione

1. Identifica i proprietari della configurazione.
  - a. Metti a conoscenza i proprietari delle configurazioni di qualsiasi esigenza di conformità, governance o normativa.
2. Identifica gli elementi e i risultati della configurazione.
  - a. Gli elementi di configurazione sono tutte le configurazioni ambientali e dell'applicazione interessate da un'implementazione all'interno della pipeline CI/CD.
  - b. I risultati finali includono criteri di successo, convalide e aspetti da monitorare.
3. Seleziona gli strumenti per la gestione della configurazione in base ai requisiti aziendali e alla pipeline di distribuzione.
4. Per modifiche significative alla configurazione, prendi in considerazione le implementazioni ponderate, ad esempio le implementazioni canary, per ridurre al minimo l'impatto di configurazioni errate.
5. Integra la gestione della configurazione nella tua pipeline CI/CD.
6. Convalida tutte le modifiche inserite.

## Risorse

Best practice correlate:

- [OPS06-BP01 Preparazione di un piano in caso di esito negativo delle modifiche](#)
- [OPS06-BP02 Implementazioni dei test](#)
- [OPS06-BP03 Utilizza strategie di deployment sicure](#)
- [OPS06-BP04 Automazione dei test e del rollback](#)

Documenti correlati:

- [AWS Control Tower](#)
- [AWS Landing Zone Accelerator](#)
- [AWS Config](#)
- [What is AWS Config?](#)
- [AWS AppConfig](#)
- [What is AWS CloudFormation?](#)
- [Strumenti per sviluppatori in AWS](#)

Video correlati:

- [AWS re:Invent 2022 - Proactive governance and compliance for AWS workloads](#)
- [AWS re:Invent 2020: Achieve compliance as code using AWS Config](#)
- [Manage and Deploy Application Configurations with AWS AppConfig](#)

## OPS05-BP04 Utilizzo di sistemi di gestione della compilazione e implementazione

Utilizza sistemi di gestione della creazione e distribuzione. Questi sistemi riducono gli errori causati dai processi manuali e il livello di impegno richiesto per la distribuzione delle modifiche.

In AWS, puoi compilare pipeline di integrazione continua/implementazione continua (CI/CD) utilizzando servizi come gli [Strumenti per sviluppatori in AWS](#) (ad esempio, AWS CodeCommit, [AWS CodeBuild](#), [AWS CodePipeline](#), [AWS CodeDeploy](#) e [AWS CodeStar](#)).

Risultato desiderato: I sistemi di gestione della costruzione e dell'implementazione supportano il sistema di distribuzione e integrazione continua (CI/CD) dell'organizzazione, che fornisce funzionalità per automatizzare rollout sicuri con le configurazioni corrette.

## Anti-pattern comuni:

- Dopo aver compilato il codice nel sistema di sviluppo, copi il file eseguibile nei sistemi di produzione e questo non si avvia. I file di log locali indicano che l'operazione è risultata impossibile a causa della mancanza di dipendenze.
- Hai creato l'applicazione con nuove funzionalità nel tuo ambiente di sviluppo e fornisci il codice per eseguire il controllo qualità (QA). Il controllo qualità non riesce perché mancano asset statici.
- Venerdì, dopo un notevole sforzo, hai creato l'applicazione manualmente nel tuo ambiente di sviluppo, incluse le nuove funzionalità codificate. Lunedì non sei in grado di ripetere le fasi che ti hanno consentito di creare correttamente la tua applicazione.
- Esegui i test creati per la nuova versione. Quindi passi la settimana successiva a configurare un ambiente di test ed eseguire tutti i test di integrazione esistenti seguiti dai test delle prestazioni. Il nuovo codice ha un impatto inaccettabile sulle prestazioni e deve essere risviluppato e quindi ritestato.

Vantaggi dell'adozione di questa best practice: Fornendo meccanismi per gestire le attività di compilazione e distribuzione, riduci il livello di impegno necessario per eseguire attività ripetitive, consenti ai membri del team di concentrarsi liberamente sulle loro attività creative di valore elevato e limiti l'introduzione di errori derivanti da procedure manuali.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

I sistemi di gestione della creazione e implementazione vengono utilizzati per tenere traccia e implementare le modifiche, ridurre gli errori causati dai processi manuali e diminuire il livello di impegno richiesto per le implementazioni sicure. Automatizza completamente la pipeline di integrazione e distribuzione dal check-in del codice fino alle fasi di creazione, test, distribuzione e convalida. Ciò riduce il lead time e i costi, incoraggia una maggiore frequenza delle modifiche, riduce il livello di impegno e aumenta la collaborazione.

## Passaggi dell'implementazione

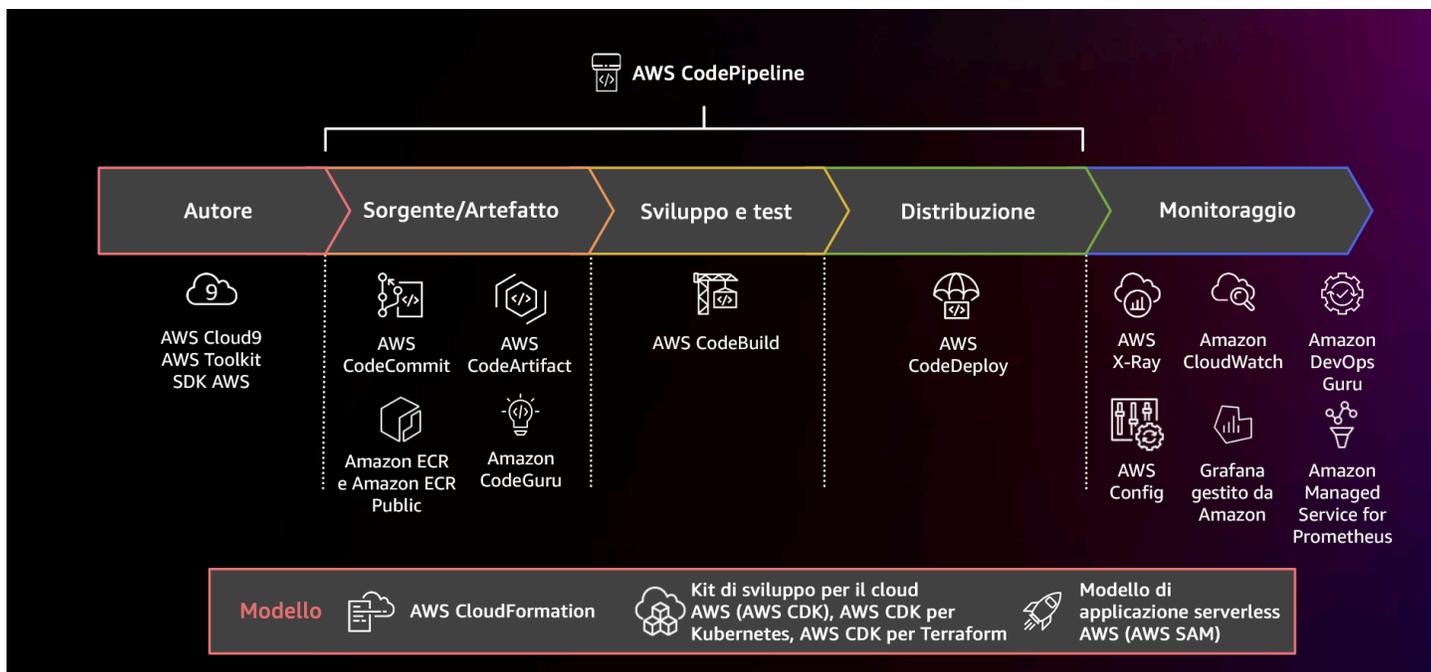


Diagramma che mostra una pipeline CI/CD che utilizza AWS CodePipeline e servizi correlati

1. Utilizza AWS CodeCommit per verificare la versione, archiviare e gestire risorse come documenti, codice sorgente e file binari.
2. Utilizza CodeBuild per compilare il codice sorgente, eseguire test delle unità e produrre artefatti pronti per l'implementazione.
3. Utilizza CodeDeploy come servizio di implementazione per automatizzare l'implementazione delle applicazioni su istanze [Amazon EC2](#), istanze on-premise, [funzioni AWS Lambda serverless](#) [Amazon ECS](#).
4. Monitora le tue implementazioni.

## Risorse

Best practice correlate:

- [OPS06-BP04 Automazione dei test e del rollback](#)

Documenti correlati:

- [Strumenti per sviluppatori in AWS](#)

- [Che cos'è AWS CodeCommit?](#)
- [Che cos'è AWS CodeBuild?](#)
- [AWS CodeBuild](#)
- [Che cos'è AWS CodeDeploy?](#)

Video correlati:

- [AWS re:Invent 2022 - AWS Well-Architected best practices for DevOps on AWS](#)

## OPS05-BP05 Esecuzione della gestione delle patch

La gestione delle patch consente di ottenere funzionalità, risolvere problemi e rispettare i requisiti di governance. Automatizza la gestione delle patch per ridurre gli errori causati dai processi manuali, dimensionare e ridurre il livello di impegno richiesto per applicare le patch.

La gestione delle patch e delle vulnerabilità fa parte delle attività di gestione dei rischi e dei vantaggi. È preferibile disporre di infrastrutture immutabili e distribuire carichi di lavoro in stati noti verificati. Se ciò non è realizzabile, l'applicazione di patch sul posto è l'alternativa.

[Amazon EC2 Image Builder](#) fornisce pipeline per l'aggiornamento di immagini AMI. Come parte della gestione delle patch, prendi in considerazione l'utilizzo di [Amazon Machine Image](#) (AMI) con una [pipeline di immagini AMI](#) o immagini del container con una [pipeline di immagini Docker](#). Inoltre, puoi utilizzare AWS Lambda, che fornisce modelli per [runtime personalizzati e librerie aggiuntive](#) per eliminare le vulnerabilità.

È consigliabile gestire gli aggiornamenti alle [Amazon Machine Image](#) per immagini Linux o Windows Server utilizzando [Amazon EC2 Image Builder](#). Puoi utilizzare [Amazon Elastic Container Registry \(Amazon ECR\)](#) con la pipeline esistente per gestire le immagini Amazon ECS ed Amazon EKS. Lambda include [funzionalità di gestione delle versioni](#).

L'applicazione di patch non deve essere eseguita sui sistemi di produzione senza prima eseguire test in un ambiente sicuro. Le patch devono essere applicate solo se supportano risultati operativi o aziendali. In AWS, è possibile utilizzare [Gestione patch di AWS Systems Manager](#) per automatizzare il processo di applicazione di patch ai sistemi gestiti e pianificare l'attività utilizzando le [finestre di manutenzione di Systems Manager](#).

Risultato desiderato: Le immagini AMI e dei container sono aggiornate, dotate di patch e pronte per il lancio. È possibile tenere traccia dello stato di tutte le immagini implementate e conoscere

la conformità delle patch. Puoi eseguire report sullo stato attuale e disporre di un processo per soddisfare le tue esigenze di conformità.

Anti-pattern comuni:

- Ti viene assegnato il compito di applicare tutte le nuove patch di sicurezza entro 2 ore, il che provoca più interruzioni a causa dell'incompatibilità dell'applicazione con le patch.
- Una libreria senza patch comporta conseguenze indesiderate in quanto parti sconosciute utilizzano vulnerabilità al suo interno per accedere al carico di lavoro.
- L'applicazione di patch agli ambienti per sviluppatori viene eseguita automaticamente senza avvisare gli sviluppatori. Gli sviluppatori ti inviano più reclami perché il loro ambiente non funziona come previsto.
- Non hai applicato patch al software pronto all'uso commerciale su un'istanza persistente. Quando hai problemi con il software e contatti il fornitore, questo ti informa che la versione non è supportata e che devi applicare le patch a un livello specifico per ricevere assistenza.
- Una patch rilasciata di recente per il software di crittografia utilizzato offre miglioramenti significativi in termini di prestazioni. Il sistema privo di patch presenta problemi di prestazioni che rimangono in vigore a causa della mancata applicazione di patch.
- Ricevi una notifica di una vulnerabilità zero-day che richiede una correzione di emergenza; quindi devi applicare manualmente le patch a tutti i tuoi ambienti.

Vantaggi dell'adozione di questa best practice: Stabilendo un processo di gestione delle patch, inclusi i criteri per l'applicazione di patch e la metodologia di distribuzione tra gli ambienti, sarai in grado di dimensionare e generare report sui livelli di patch. Ciò fornisce garanzie sull'applicazione delle patch di sicurezza e una chiara visibilità sullo stato delle correzioni note in atto. Ciò incoraggia l'adozione delle caratteristiche e funzionalità desiderate, aiuta a eliminare rapidamente i problemi e a mantenere la conformità alla governance. Implementa sistemi di gestione delle patch e automazione per ridurre il livello di impegno per distribuire le patch e limitare gli errori causati dai processi manuali.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Applica patch ai sistemi per correggere gli errori, ottenere le funzionalità o le capacità desiderate e assicurare la conformità alle policy di governance e ai requisiti di supporto del vendor. Nei sistemi immutabili, distribuisci con il set di patch appropriato per raggiungere il risultato desiderato.

Automatizza il meccanismo di gestione delle patch per ridurre il tempo necessario per applicare le patch, evitare gli errori causati dai processi manuali e diminuire il livello di impegno richiesto per applicare le patch.

## Passaggi dell'implementazione

Per Amazon EC2 Image Builder:

1. specifica i dettagli della pipeline utilizzando Amazon EC2 Image Builder:
  - a. Crea una pipeline di immagini e assegnale un nome
  - b. Definisci la pianificazione e il fuso orario della pipeline
  - c. Configura eventuali dipendenze
2. Scegli una ricetta:
  - a. Seleziona una ricetta esistente o creane una nuova
  - b. Seleziona il tipo di immagine
  - c. Assegna un nome e una versione alla tua ricetta
  - d. Seleziona l'immagine di base
  - e. Aggiungi componenti di compilazione e inseriscili nel registro di destinazione
3. Facoltativo: definisci la configurazione dell'infrastruttura.
4. Facoltativo: definisci le impostazioni di configurazione.
5. Revisiona le impostazioni.
6. Mantieni il livello di igiene delle ricette a livelli ottimali.

Per Gestione patch di Systems Manager:

1. Crea una patch di base.
2. Seleziona un metodo per le operazioni di definizione del percorso.
3. Abilita il report e la scansione della conformità.

## Risorse

Best practice correlate:

- [OPS06-BP04 Automazione dei test e del rollback](#)

## Documenti correlati:

- [What is Amazon EC2 Image Builder](#)
- [Create an image pipeline using the Amazon EC2 Image Builder](#)
- [Create a container image pipeline](#)
- [Gestione patch di AWS Systems Manager](#)
- [Working with Patch Manager](#)
- [Working with patch compliance reports](#)
- [Strumenti per sviluppatori in AWS](#)

## Video correlati:

- [CI/CD per applicazioni serverless su AWS](#)
- [Progettare nell'ottica Ops](#)

## Esempi correlati:

- [Well-Architected Labs: Inventario e gestione delle patch](#)
- [AWS Systems Manager Patch Manager tutorials](#)

## OPS05-BP06 Condivisione degli standard di progettazione

Condividi le best practice con i team per incrementare la consapevolezza e potenziare al massimo i vantaggi delle attività di sviluppo. Documentale e mantienile aggiornate di pari passo con l'evoluzione dell'architettura. Se nella tua organizzazione vengono applicati standard condivisi, è fondamentale che esistano meccanismi per richiedere aggiunte, modifiche ed eccezioni agli standard. Senza questa opzione, gli standard diventano un ostacolo per l'innovazione.

Risultato desiderato: Gli standard di progettazione vengono condivisi fra team nelle organizzazioni. Vengono documentati e tenuti aggiornati in base all'evoluzione delle best practice.

## Anti-pattern comuni:

- Due team di sviluppo hanno creato ciascuno un servizio di autenticazione utente. Gli utenti devono mantenere un set separato di credenziali per ogni parte del sistema a cui vogliono accedere.
- Ogni team gestisce la propria infrastruttura. Un nuovo requisito di conformità impone una modifica all'infrastruttura e ogni team la implementa in modo diverso.

Vantaggi dell'adozione di questa best practice: L'uso di standard condivisi incoraggia l'applicazione di best practice e permette di ottenere i massimi vantaggi dalle attività di sviluppo. La documentazione e l'aggiornamento degli standard di progettazione tengono l'organizzazione al passo con le best practice e i requisiti di sicurezza e conformità.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Condividi le best practice, gli standard di progettazione, gli elenchi di controllo, le procedure operative, le linee guida e i requisiti di governance esistenti tra team diversi. Definisci procedure per richiedere modifiche, aggiunte ed eccezioni agli standard di progettazione per supportare il miglioramento e l'innovazione. Rendi noto ai team il contenuto pubblicato. Predisponi un meccanismo per mantenere aggiornati gli standard di progettazione in base all'emergere di nuove best practice.

### Esempio del cliente

AnyCompany Retail ha un team interfunzionale che crea modelli di architettura software. Questo team crea l'architettura con conformità e governance integrate. I team che adottano gli standard condivisi traggono vantaggio dall'integrazione di conformità e governance. Possono creare rapidamente soluzioni sulla base degli standard di progettazione. Il team responsabile dell'architettura si incontra ogni trimestre per valutare i modelli architetturali e aggiornarli, se necessario.

### Passaggi dell'implementazione

1. Identifica un team interfunzionale responsabile dello sviluppo e dell'aggiornamento degli standard di progettazione. Questo team collaborerà con gli stakeholder in tutta l'organizzazione per sviluppare standard di progettazione, procedure operative, elenchi di controllo, linee guida e requisiti di governance. Documenta gli standard di progettazione e condividili internamente all'organizzazione.
  - a. [AWS Service Catalog](#) può aiutarti a creare portfolio che rappresentano gli standard di progettazione usando il modello Infrastruttura come codice (IaC). Puoi condividere portfolio tra più account.
2. Predisponi un meccanismo per mantenere aggiornati gli standard di progettazione man mano che vengono identificate nuove best practice.
3. Se gli standard di progettazione vengono applicati a livello centrale, definisci un processo per richiedere modifiche, aggiornamenti ed eccezioni.

Livello di impegno per il piano di implementazione: medio. Lo sviluppo di un processo per creare e condividere standard di progettazione può richiedere il coordinamento e la cooperazione con gli stakeholder in tutta l'organizzazione.

## Risorse

Best practice correlate:

- [OPS01-BP03 Valutazione dei requisiti di governance](#) – I requisiti di governance influiscono sugli standard di progettazione.
- [OPS01-BP04 Valutazione dei requisiti di conformità](#) – La conformità è un fattore essenziale nella creazione di standard di progettazione.
- [OPS07-BP02 Revisione costante della prontezza operativa](#) – Gli elenchi di controllo della prontezza operativa sono un meccanismo per implementare standard di progettazione durante la progettazione del carico di lavoro.
- [OPS11-BP01 Definizione di un processo per il miglioramento continuo](#) – L'aggiornamento degli standard di progettazione contribuisce a un miglioramento continuo.
- [OPS11-BP04 Gestione delle informazioni](#) – Nell'ambito della procedura di gestione delle informazioni, documenta e condividi gli standard di progettazione.

Documenti correlati:

- [Automate AWS Backups with AWS Service Catalog](#)
- [AWS Service Catalog Account Factory-Enhanced](#)
- [Expedia Group crea un'offerta Database as a Service \(DBaaS\) usando il AWS Service Catalog](#)
- [Mantenimento della visibilità sull'uso di modelli architetturali cloud](#)
- [Simplify sharing your AWS Service Catalog portfolios in an AWS Organizations setup](#)

Video correlati:

- [AWS Service Catalog – Getting Started](#)
- [AWS re:Invent 2020: Manage your AWS Service Catalog portfolios like an expert](#)

Esempi correlati:

- [AWS Service Catalog Reference Architecture](#)

- [AWS Service Catalog Workshop](#)

Servizi correlati:

- [AWS Service Catalog](#)

## OPS05-BP07 Implementazione di prassi per migliorare la qualità del codice

Implementa prassi per migliorare la qualità del codice e ridurre al minimo i difetti. Alcuni esempi includono sviluppo basato su test, revisioni del codice, adozione degli standard e programmazione in coppia. Inserisci queste prassi nel processo di integrazione continua e distribuzione continua.

Risultato desiderato: l'organizzazione usa best practice come le revisioni del codice e la programmazione in coppia per migliorare la qualità del codice. Sviluppatori e operatori adottano le best practice per la qualità del codice nell'ambito del ciclo di vita di sviluppo del software.

Anti-pattern comuni:

- Commit del codice nel ramo principale dell'applicazione senza alcuna revisione. In questo modo, la modifica viene automaticamente implementata nell'ambiente di produzione e causa un'interruzione.
- Sviluppo di una nuova applicazione senza unit test, test end-to-end o test di integrazione. Non è possibile in alcun modo testare l'applicazione prima dell'implementazione.
- I team apportano modifiche manuali nell'ambiente di produzione per gestire gli errori. Le modifiche non vengono sottoposte a test o revisioni del codice, né vengono acquisite o registrate durante i processi di integrazione continua e distribuzione continua.

Vantaggi dell'adozione di questa best practice: l'adozione delle procedure per migliorare la qualità del codice ti consente di ridurre al minimo i problemi di produzione. La qualità del codice facilita l'uso delle best practice, come la programmazione in coppia, le revisioni del codice e l'implementazione di strumenti di produttività basati sull'IA.

Livello di rischio associato se questa best practice non fosse adottata: medio

### Guida all'implementazione

Implementa prassi per migliorare la qualità del codice in modo da ridurre gli errori prima dell'implementazione. Usa prassi come lo sviluppo basato su test, le revisioni del codice e la programmazione in coppia per migliorare la qualità dello sviluppo.

Usa la potenza dell'IA generativa con Amazon Q Developer per migliorare la produttività degli sviluppatori e la qualità del codice. Amazon Q Developer include la generazione di suggerimenti di codice (basati su modelli linguistici di grandi dimensioni), la produzione di unit test (comprese le condizioni limite) e il miglioramento della sicurezza del codice tramite il rilevamento e la correzione delle vulnerabilità di sicurezza.

### Esempio del cliente

AnyCompany Retail adotta diverse prassi per migliorare la qualità del codice. L'azienda ha adottato lo sviluppo basato su test come standard per la scrittura di applicazioni. Per alcune nuove funzionalità, gli sviluppatori eseguiranno la programmazione in coppia durante uno sprint. Ogni richiesta pull viene sottoposta a una revisione del codice da parte di uno sviluppatore senior prima di essere integrata e implementata.

### Passaggi dell'implementazione

1. Adotta prassi per la qualità del codice come lo sviluppo basato su test, le revisioni del codice e la programmazione in coppia nel processo di integrazione continua e distribuzione continua. Usa queste tecniche per migliorare la qualità del software.
  - a. Utilizza [Amazon Q Developer](#), uno strumento di IA generativa che consente di creare casi di unit test (comprese le condizioni limite), generare funzioni utilizzando codice e commenti, implementare gli algoritmi noti, rilevare violazioni delle policy di sicurezza e vulnerabilità nel codice, rilevare segreti, analizzare l'infrastructure as code (IaC), documentare il codice e apprendere più rapidamente librerie di codici di terze parti.
  - b. [Amazon CodeGuru Reviewer](#) può fornire suggerimenti di programmazione per il codice Java e Python tramite il machine learning.
  - c. Puoi creare ambienti di sviluppo condivisi con [AWS Cloud9](#) in cui collaborare allo sviluppo del codice.

Livello di impegno per il piano di implementazione: medio. Esistono molti modi per implementare questa best practice, ma la realizzazione dell'adozione da parte dell'organizzazione può essere problematica.

### Risorse

Best practice correlate:

- [OPS05-BP02 Test e convalida delle modifiche](#)

- [OPS05-BP06 Condivisione degli standard di progettazione](#)

#### Documenti correlati:

- [Adopt a test-driven development approach](#)
- [Accelerate your Software Development Lifecycle with Amazon Q](#)
- [Amazon Q Developer, now generally available, includes previews of new capabilities to reimagine developer experience](#)
- [The Ultimate Cheat Sheet for Using Amazon Q Developer in Your IDE](#)
- [Shift-Left Workload, leveraging AI for Test Creation](#)
- [Amazon Q Developer Center](#)
- [10 ways to build applications faster with Amazon CodeWhisperer](#)
- [Looking beyond code coverage with Amazon CodeWhisperer](#)
- [Best Practices for Prompt Engineering with Amazon CodeWhisperer](#)
- [Agile Software Guide](#)
- [My CI/CD pipeline is my release captain](#)
- [Automate code reviews with Amazon CodeGuru Reviewer](#)
- [Adopt a test-driven development approach](#)
- [How DevFactory builds better applications with Amazon CodeGuru](#)
- [On Pair Programming](#)
- [RENGA Inc. automates code reviews with Amazon CodeGuru](#)
- [The Art of Agile Development: Test-Driven Development](#)
- [Why code reviews matter \(and actually save time!\)](#)

#### Video correlati:

- [Implement an API with Amazon Q Developer Agent for Software Development](#)
- [Installing, Configuring, & Using Amazon Q Developer with JetBrains IDEs \(How-to\)](#)
- [Mastering the art of Amazon CodeWhisperer - YouTube playlist](#)
- [AWS re:Invent 2020: Continuous improvement of code quality with Amazon CodeGuru](#)
- [AWS Summit ANZ 2021 - Driving a test-first strategy with CDK and test driven development](#)

## Servizi correlati:

- [Amazon Q Developer](#)
- [Amazon CodeGuru Reviewer](#)
- [Amazon CodeGuru Profiler](#)
- [AWS Cloud9](#)

## OPS05-BP08 Utilizzo di più ambienti

Utilizza ambienti multipli per sperimentare, sviluppare e testare il carico di lavoro. Applica livelli crescenti di controlli man mano che gli ambienti si avvicinano alla fase di produzione per avere la certezza che il carico di lavoro funzioni come previsto una volta implementato.

Risultato desiderato: Disponi di più ambienti che riflettono le tue esigenze di conformità e governance. Testi e promuovi il codice negli ambienti lungo il tuo percorso verso la produzione.

### Anti-pattern comuni:

- Stai sviluppando in un ambiente di sviluppo condiviso e un altro sviluppatore sovrascrive le tue modifiche al codice.
- I controlli di sicurezza restrittivi nell'ambiente di sviluppo condiviso impediscono di sperimentare nuovi servizi e funzionalità.
- Esegui test di carico sui tuoi sistemi di produzione e causa un'interruzione per i tuoi utenti.
- Si è verificato un errore critico che ha causato la perdita di dati nella produzione. Nel tuo ambiente di produzione tenti di ricreare le condizioni che portano alla perdita di dati in modo da poter identificare come si è verificata e impedire che si ripeta. Per evitare un'ulteriore perdita di dati durante il test, devi rendere l'applicazione non disponibile per i tuoi utenti.
- Stai operando un servizio multi-tenant e non sei in grado di supportare la richiesta di un cliente per un ambiente dedicato.
- Ogni volta che esegui un test, lo fai nel tuo ambiente di produzione.
- Ritieni che la semplicità di un singolo ambiente prevalga sulla portata dell'impatto che possono avere modifiche all'interno dell'ambiente.

Vantaggi dell'adozione di questa best practice: puoi supportare più ambienti di sviluppo, test e produzione simultanei senza creare conflitti tra sviluppatori o community di utenti.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Utilizza più ambienti e fornisci agli sviluppatori ambienti di sperimentazione (sandbox) con controlli minimi per incoraggiare la sperimentazione. Fornisci ambienti di sviluppo individuali per facilitare il lavoro in parallelo, incrementando l'agilità dello sviluppo. Implementa controlli più rigorosi negli ambienti che si avvicinano alla produzione per consentire agli sviluppatori di innovare. Utilizza l'approccio Infrastructure-as-Code e sistemi di gestione delle configurazioni per distribuire ambienti configurati in modo coerente con i controlli presenti in produzione per assicurare che i sistemi funzionino nel modo previsto quando vengono distribuiti. Quando gli ambienti non vengono utilizzati, disattivali per evitare costi associati alle risorse inattive, ad esempio i sistemi di sviluppo nelle ore serali e nei fine settimana. Durante i test di carico, è necessario implementare ambienti equivalenti a quelli di produzione per migliorare la validità dei risultati.

## Risorse

Documenti correlati:

- [Pianificatore di istanze su AWS](#)
- [Che cos'è AWS CloudFormation?](#)

## OPS05-BP09 Applicazione di modifiche frequenti, minime e reversibili

Le modifiche frequenti, minime e reversibili riducono la portata e l'impatto di una modifica. Le modifiche frequenti, minime e reversibili, se effettuate utilizzando congiuntamente sistemi di gestione delle modifiche, di gestione della configurazione e di compilazione e distribuzione, riducono la portata e l'impatto di una modifica. Questo si traduce in una risoluzione dei problemi più efficace, accelerando la correzione e mantenendo la possibilità di rollback delle modifiche.

Anti-pattern comuni:

- Distribuisce una nuova versione della tua applicazione ogni trimestre con una finestra di modifica, il che comporta la disattivazione di un servizio di base.
- Spesso apporti modifiche allo schema del database senza che ne venga tenuta traccia nei sistemi di gestione.
- Esegui aggiornamenti manuali sul posto, sovrascrivendo le installazioni e le configurazioni esistenti, senza avere un chiaro piano di rollback.

Vantaggi dell'adozione di questa best practice: Le attività di sviluppo sono più rapide grazie all'implementazione frequente di modifiche minime. Quando le modifiche sono minime, è molto più semplice identificare se hanno conseguenze indesiderate e, in tal caso, ripristinare la condizione precedente. Quando le modifiche sono reversibili, il rischio di implementare le modifiche è minore in quanto il ripristino è semplificato. Il processo di modifica comporta un rischio ridotto e l'impatto di una modifica non corretta è ridotto.

Livello di rischio associato se questa best practice non fosse adottata: basso

## Guida all'implementazione

Applica modifiche frequenti, minime e reversibili per ridurre la portata e l'impatto di una modifica. In questo modo si semplifica la risoluzione dei problemi, si velocizza la correzione ed è possibile eseguire il rollback di una modifica. Inoltre, aggiunge più rapidamente valore al business.

## Risorse

Best practice correlate:

- [OPS05-BP03 Utilizzo di sistemi di gestione delle configurazioni](#)
- [OPS05-BP04 Utilizzo di sistemi di gestione della compilazione e implementazione](#)
- [OPS06-BP04 Automazione dei test e del rollback](#)

Documenti correlati:

- [Implementazione di microservizi in AWS](#)
- [Microservices - Observability](#)

## OPS05-BP10 Automazione completa dell'integrazione e della distribuzione

Automatizza la creazione, la distribuzione e il test del carico di lavoro. Questo riduce gli errori causati dai processi manuali e l'impegno necessario per distribuire le modifiche.

Applica i metadati utilizzando i [tag delle risorse](#) e [AWS Resource Groups](#) seguendo una [strategia di applicazione dei tag](#) per agevolare l'identificazione delle risorse. Applica tag alle risorse per organizzare, monitorare i costi e controllare gli accessi e ottimizza l'esecuzione delle attività operative automatizzate.

Risultato desiderato: Chi si occupa di sviluppo utilizza strumenti per distribuire codice ed effettuare la promozione a produzione. Gli sviluppatori non devono effettuare il login alla AWS Management Console per fornire gli aggiornamenti. Esiste un audit trail completo di modifiche e configurazioni che soddisfa le esigenze di governance e conformità. I processi sono ripetibili e standardizzati tra i team. Gli sviluppatori sono liberi di concentrarsi sullo sviluppo e sui rilasci del codice, aumentando la produttività.

Anti-pattern comuni:

- Venerdì termini la creazione del nuovo codice per il ramo delle funzionalità. Lunedì, dopo aver eseguito gli script di test di qualità del codice e tutti gli script dei test di unità, effettui il check-in del codice per il prossimo rilascio programmato.
- Ti verrà assegnato di codificare una correzione per un problema critico che interessa un numero elevato di clienti nella produzione. Dopo aver testato la correzione, esegui il commit del codice e richiedi via e-mail alla gestione delle modifiche l'approvazione per implementarlo in produzione.
- In qualità di sviluppatore, accedi alla AWS Management Console per creare un nuovo ambiente di sviluppo utilizzando metodi e sistemi non standard.

Vantaggi dell'adozione di questa best practice: Implementando sistemi di gestione automatizzati di compilazione e implementazione, si riduce il numero di errori causati dai processi manuali e lo sforzo di distribuire le modifiche aiutando i membri del team a concentrarsi sull'offerta di valore aggiunto. Maggiore velocità di consegna man mano che procedi verso la promozione a produzione.

Livello di rischio associato se questa best practice non fosse adottata: basso

## Guida all'implementazione

Utilizza i sistemi di gestione della compilazione e implementazione per tenere traccia e implementare le modifiche, ridurre gli errori causati dai processi manuali e ridurre il livello di impegno richiesto. Automatizza completamente la pipeline di integrazione e distribuzione dal check-in del codice fino alle fasi di creazione, test, distribuzione e convalida. In questo modo è possibile diminuire il lead time, incoraggiare una maggiore frequenza di modifica, ridurre il livello di impegno e accelerare il time-to-market, il che si traduce in una maggiore produttività e in un aumento della sicurezza del codice man mano che procedi con la promozione verso la produzione.

## Risorse

Best practice correlate:

- [OPS05-BP03 Utilizzo di sistemi di gestione delle configurazioni](#)
- [OPS05-BP04 Utilizzo di sistemi di gestione della compilazione e implementazione](#)

Documenti correlati:

- [Che cos'è AWS CodeBuild?](#)
- [Che cos'è AWS CodeDeploy?](#)

Video correlati:

- [AWS re\Invent 2022 - AWS Well-Architected best practices for DevOps on AWS](#)

## Riduzione dei rischi di distribuzione

Adotta prassi che consentano di fornire un feedback rapido sulla qualità e permettano un ripristino veloce dalle modifiche che non hanno i risultati previsti. L'uso di queste prassi consente di mitigare l'impatto dei problemi introdotti attraverso la distribuzione delle modifiche.

La progettazione del tuo carico di lavoro dovrebbe includere il modo in cui sarà distribuito, aggiornato e utilizzato. Implementerai pratiche di progettazione che permettono di ridurre difetti e trovare soluzioni in modo rapido e sicuro.

Best practice

- [OPS06-BP01 Preparazione di un piano in caso di esito negativo delle modifiche](#)
- [OPS06-BP02 Implementazioni dei test](#)
- [OPS06-BP03 Utilizza strategie di deployment sicure](#)
- [OPS06-BP04 Automazione dei test e del rollback](#)

### OPS06-BP01 Preparazione di un piano in caso di esito negativo delle modifiche

Pianifica il ripristino di uno stato corretto noto o la correzione nell'ambiente di produzione nel caso in cui l'implementazione generi un risultato indesiderato. Disporre di una politica per stabilire un piano di questo tipo aiuta tutti i team a sviluppare strategie di ripristino dalle modifiche con esito negativo. Alcune strategie di esempio sono le fasi di deployment e rollback, le politiche di modifica,

i flag di funzionalità, l'isolamento del traffico e lo spostamento del traffico. Una singola release può includere più modifiche ai componenti correlati. La strategia dovrebbe fornire la capacità di resistere o ripristinare in caso di guasto generato da qualsiasi modifica dei componenti.

Risultato desiderato: hai preparato un piano di ripristino dettagliato per la modifica in caso di fallimento. Inoltre, hai ridotto le dimensioni della release per ridurre al minimo il potenziale impatto su altri componenti del carico di lavoro. Di conseguenza, hai ridotto l'impatto aziendale abbreviando i potenziali tempi di inattività causati da una modifica non riuscita e aumentando la flessibilità e l'efficienza dei tempi di ripristino.

Anti-pattern comuni:

- Hai eseguito un deployment e l'applicazione è diventata instabile, ma sembra che ci siano utenti attivi sul sistema. Devi decidere se eseguire il rollback della modifica e influire sugli utenti attivi o aspettare di eseguire il rollback della modifica, sapendo che gli utenti potranno essere comunque influenzati.
- Dopo aver apportato una modifica di routine, i nuovi ambienti sono accessibili, ma una delle sottoreti è diventata irraggiungibile. Devi decidere se eseguire il rollback di tutto o provare a correggere il problema della sottorete inaccessibile. Mentre prendi tale decisione, la sottorete rimane irraggiungibile.
- I tuoi sistemi non sono progettati in modo da consentire loro di essere aggiornati con release più piccole. Di conseguenza, è difficile annullare tali modifiche di massa (bulk changes) durante un deployment conclusosi con esito negativo.
- Non utilizzi l'Infrastruttura come codice (IaC) e hai apportato aggiornamenti manuali all'infrastruttura che hanno portato a configurazioni indesiderate. Non è possibile tracciare e ripristinare in modo efficace le modifiche manuali.
- Poiché non hai misurato l'aumento della frequenza dei deployment, il tuo team non è incentivato a ridurre le dimensioni delle modifiche e a migliorare i piani di rollback per ogni modifica, con conseguente aumento dei rischi e dei tassi di fallimento.
- Non misura la durata totale di un'interruzione causata da modifiche con esito negativo. Il tuo team non è in grado di stabilire le priorità e migliorare il processo di deployment e l'efficacia del piano di ripristino.

Vantaggi dell'adozione di questa best practice: Avere un piano per il ripristino dopo le modifiche non riuscite riduce al minimo il tempo medio di ripristino (MTTR) e riduce l'impatto sull'azienda.

Livello di rischio associato se questa best practice non fosse adottata: alto

## Guida all'implementazione

Una politica e una pratica coerenti e documentate adottate dai team di rilascio consentono a un'organizzazione di pianificare cosa dovrebbe succedere in caso di modifiche con esito negativo. In circostanze specifiche la politica dovrebbe consentire la possibilità di apportare correzioni per garantire la prosecuzione del processo. In entrambe le situazioni, un piano di correzione (fix forward) o ripristino (rollback) deve essere ben documentato e testato prima dell'implementazione nei sistemi di produzione live, in modo da ridurre al minimo il tempo necessario per ripristinare una modifica.

### Passaggi dell'implementazione

1. Documenta le politiche che richiedono ai team di disporre di piani efficaci per invertire le modifiche entro un periodo di tempo specificato.
  - a. Le politiche devono specificare quando è consentita una situazione di applicazione di correzioni per garantire la prosecuzione del processo.
  - b. Richiedi un piano di rollback documentato che sia accessibile a tutti i soggetti coinvolti.
  - c. Specifica i requisiti per il rollback (ad esempio, quando si rileva che sono state implementate modifiche non autorizzate).
2. Analizza il livello di impatto di tutte le modifiche relative a ciascun componente di un carico di lavoro.
  - a. Consenti che le modifiche ripetibili siano standardizzate, basate su modelli e preautorizzate se seguono un flusso di lavoro coerente che applica le politiche di modifica.
  - b. Riduci il potenziale impatto di qualsiasi modifica riducendone le dimensioni, in modo che il ripristino richieda meno tempo e abbia un impatto minore sulle attività aziendali.
  - c. Assicurati che le procedure di rollback riportino il codice allo stato corretto noto per evitare incidenti, ove possibile.
3. Integra strumenti e flussi di lavoro per applicare le tue politiche in modo programmatico.
4. Rendi visibili i dati sulle modifiche agli altri responsabili di carichi di lavoro per migliorare la velocità di diagnosi di eventuali modifiche con esito negativo che non possono essere ripristinate.
  - a. Misura il successo di questa pratica utilizzando dati di modifica visibili e identifica miglioramenti iterativi.
5. Utilizza gli strumenti di monitoraggio per verificare il successo o il fallimento di un deployment per accelerare il processo decisionale sul rollback.
6. Misura la durata dell'interruzione durante una modifica con esito negativo per migliorare continuamente i tuoi piani di ripristino.

Livello di impegno per il piano di implementazione: Medio

## Risorse

Best practice correlate:

- [OPS06-BP04 Automazione dei test e del rollback](#)

Documenti correlati:

- [AWS Builders Library | Ensuring Rollback Safety During Deployments](#)
- [AWS Whitepaper | Change Management in the Cloud](#)

Video correlati:

- [re:Invent 2019 | Amazon's approach to high-availability deployment](#)

## OPS06-BP02 Implementazioni dei test

Testa le procedure di rilascio in pre-produzione utilizzando la stessa configurazione di deployment, i controlli di sicurezza, i passaggi e le procedure utilizzati nell'ambiente di produzione. Verifica che tutte le fasi implementate siano state completate come previsto, ad esempio l'ispezione di file, configurazioni e servizi. Verifica ulteriormente tutte le modifiche con test funzionali, di integrazione e di carico, oltre ad attivare tutte le attività di monitoraggio come i controlli dell'integrità. Eseguendo questi test, è possibile identificare tempestivamente i problemi di deployment con l'opportunità di pianificarli e mitigarli prima del passaggio nell'ambiente di produzione.

Puoi creare ambienti paralleli temporanei per testare ogni modifica. Automatizza il deployment degli ambienti di test utilizzando l'Infrastruttura come codice (IaC) per ridurre la quantità di lavoro necessaria e garantire stabilità, coerenza e una distribuzione più rapida delle funzionalità.

Risultato desiderato: La tua organizzazione adotta una cultura di sviluppo che include il test delle implementazioni. Ciò garantisce che i team siano concentrati sulla realizzazione di valore aziendale anziché sulla gestione delle release. I team vengono coinvolti fin dall'identificazione dei rischi di deployment per determinare il percorso di mitigazione appropriato.

Anti-pattern comuni:

- Durante le release di produzione, le implementazioni non testate causano problemi frequenti che richiedono una risoluzione mirata e l'escalation.
- La tua release contiene porzioni di Infrastruttura come codice (IaC) che aggiornano le risorse esistenti. Non sei sicuro che l'IaC funzionerà correttamente e non avrà un impatto sulle risorse.
- Viene implementata una nuova funzionalità interessante nella tua applicazione. Non funziona come previsto e non c'è visibilità finché non viene segnalata dagli utenti interessati.
- I certificati vengono aggiornati. Si installano accidentalmente i certificati sui componenti sbagliati, il che non viene rilevato e influisce sui visitatori poiché non è possibile stabilire una connessione sicura al sito web.

Vantaggi dell'adozione di questa best practice: Test approfonditi in fase di pre-produzione delle procedure di implementazione e delle modifiche da queste introdotte riducono al minimo il potenziale impatto sulla produzione causato dalle fasi di implementazione. Ciò aumenta la fiducia durante il rilascio in produzione e riduce al minimo la necessità di supporto operativo senza rallentare la velocità di distribuzione delle modifiche apportate.

Livello di rischio associato se questa best practice non fosse adottata: alto

## Guida all'implementazione

Testare il processo di deployment è importante quanto testare le modifiche derivanti dal deployment. Ciò può essere ottenuto testando le fasi di deployment in un ambiente di pre-produzione che rispecchi il più fedelmente possibile quello di produzione. I problemi più comuni, come fasi di deployment incomplete o contenenti errori o configurazioni errate, possono essere individuati di conseguenza prima di passare all'ambiente di produzione. Inoltre, è possibile testare le fasi di ripristino.

### Esempio del cliente

Nell'ambito della sua pipeline di integrazione continua e distribuzione continua (CI/CD), AnyCompany Retail esegue i passaggi definiti necessari per rilasciare aggiornamenti dell'infrastruttura e del software per i propri clienti in un ambiente simile a quello di produzione. La pipeline comprende controlli preliminari per rilevare il "drift" (il rilevamento delle modifiche alle risorse eseguite al di fuori dell'IaC) nelle risorse prima del deployment, nonché per convalidare le azioni che l'IaC intraprende al suo avvio. Convalida le fasi di deployment, ad esempio la verifica che determinati file e configurazioni siano presenti e che i servizi siano in esecuzione e rispondano correttamente ai controlli di integrità sull'host locale, prima di effettuare nuovamente la registrazione sul sistema di bilanciamento del

carico. Inoltre, tutte le modifiche attivano una serie di test automatici, come test funzionali, di sicurezza, di regressione, di integrazione e di carico.

## Passaggi dell'implementazione

1. Esegui controlli di pre-installazione per rispecchiare l'ambiente di pre-produzione in produzione.
  - a. utilizza [il rilevamento della deviazione](#) per rilevare quando le risorse sono state modificate all'esterno di AWS CloudFormation.
  - b. utilizza [i set di modifiche](#) per verificare che l'intento dell'aggiornamento dello stack corrisponda alle azioni intraprese da AWS CloudFormation quando viene avviato il set di modifiche.
2. Ciò attiva una fase di approvazione manuale in [AWS CodePipeline](#) per autorizzare l'implementazione nell'ambiente di preproduzione.
3. Utilizza configurazioni di implementazione come [file AWS CodeDeploy AppSpec](#) per definire le fasi di implementazione e convalida.
4. Ove applicabile, [integra AWS CodeDeploy con altri servizi AWS](#) o [integra AWS CodeDeploy con prodotti e servizi dei partner](#).
5. [Monitora le implementazioni](#) usando le notifiche di eventi Amazon CloudWatch, AWS CloudTrail e Amazon SNS.
6. Esegui test automatici post-deployment, inclusi test funzionali, di sicurezza, di regressione, di integrazione e di carico.
7. [Risoluzione dei problemi](#) di implementazione.
8. La corretta convalida dei passaggi precedenti dovrebbe attivare un flusso di lavoro di approvazione manuale per autorizzare l'implementazione nell'ambiente di produzione.

Livello di impegno per il piano di implementazione: alto

## Risorse

Best practice correlate:

- [OPS05-BP02 Test e convalida delle modifiche](#)

Documenti correlati:

- [AWS Builders' Library | Automating safe, hands-off deployments | Test Deployments](#)
- [AWS Whitepaper | Practicing Continuous Integration and Continuous Delivery on AWS](#)

- [The Story of Apollo - Amazon's Deployment Engine](#)
- [Come eseguire test e debug con AWS CodeDeploy in locale prima di distribuire il codice](#)
- [Integrating Network Connectivity Testing with Infrastructure Deployment](#)

Video correlati:

- [re:Invent 2020 | Testing software and systems at Amazon](#)

Esempi correlati:

- [Tutorial | Deploy and Amazon ECS service with a validation test](#)

## OPS06-BP03 Utilizza strategie di deployment sicure

I roll-out sicuri della produzione controllano il flusso di modifiche vantaggiose con l'obiettivo di ridurre al minimo l'impatto percepito di tali modifiche sui clienti. I controlli di sicurezza forniscono meccanismi di ispezione per convalidare i risultati desiderati e limitare l'ambito di impatto derivante da eventuali difetti introdotti dalle modifiche o da errori di deployment. I roll-out sicuri possono includere strategie come feature-flags, one-box, roll-out (release canary), immutabili, suddivisioni del traffico e deployment blu/verdi.

Risultato desiderato: l'organizzazione utilizza un sistema di distribuzione e integrazione continua (CI/CD) che fornisce funzionalità per automatizzare roll-out sicuri. I team sono tenuti a utilizzare strategie di roll-out sicure appropriate.

Anti-pattern comuni:

- Distribuisci una modifica non riuscita a tutta la produzione contemporaneamente. Di conseguenza, tutti i clienti vengono colpiti contemporaneamente.
- Un difetto introdotto in un deployment simultaneo su tutti i sistemi richiede una release di emergenza. La correzione per tutti i clienti richiede diversi giorni.
- La gestione della release di produzione richiede la pianificazione e la partecipazione di diversi team. Ciò limita la tua capacità di aggiornare frequentemente le funzionalità per i tuoi clienti.
- Esegui un deployment variabile modificando i sistemi esistenti. Dopo aver scoperto che la modifica non è andata a buon fine, devi modificare nuovamente i sistemi per ripristinare la versione precedente estendendo il tempo di ripristino.

Vantaggi dell'adozione di questa best practice: Le implementazioni automatizzate bilanciano la velocità dei roll-out con la fornitura costante di modifiche vantaggiose per i clienti. La limitazione dell'impatto previene costosi errori di deployment e massimizza la capacità dei team di rispondere in modo efficiente ai guasti.

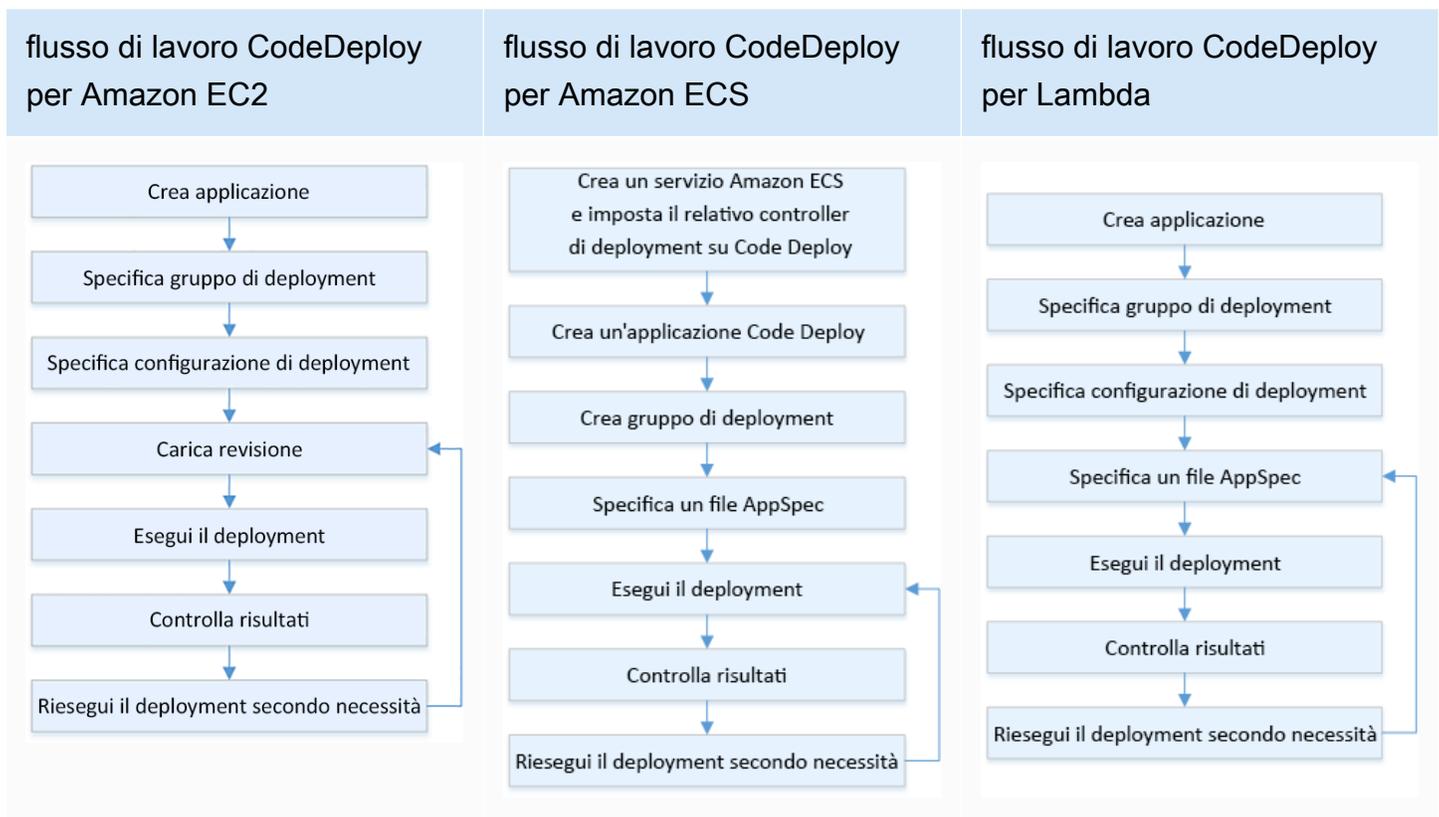
Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Gli errori della distribuzione continua possono portare a una ridotta disponibilità del servizio e a esperienze dei clienti negative. Per massimizzare il tasso di deployment di successo, implementa i controlli di sicurezza nel processo di rilascio end-to-end per ridurre al minimo gli errori di deployment, con l'obiettivo di raggiungere il traguardo di zero errori.

### Esempio del cliente

La missione di AnyCompany Retail è raggiungere deployment con tempi di inattività minimi o pari a zero, il che significa che non vi deve essere alcun impatto percepibile dagli utenti durante i deployment. A tal fine, l'azienda ha stabilito modelli di deployment (vedere il seguente diagramma del flusso di lavoro) come roll-out e deployment blu/verdi. Tutti i team adottano uno o più di questi modelli nella loro pipeline CI/CD.



## Passaggi dell'implementazione

1. Utilizza un flusso di lavoro di approvazione per avviare la sequenza delle fasi di roll-out della produzione al momento della promozione alla produzione.
2. Utilizza un sistema di implementazione automatizzato come [AWS CodeDeploy](#). AWS CodeDeploy [opzioni di implementazione](#) include le implementazioni locali (in-place) per EC2/on-premise e le implementazioni blu/verdi per EC2/on-premise AWS Lambda e Amazon ECS (vedi il diagramma del flusso di lavoro precedente).
  - a. Ove applicabile, [integra AWS CodeDeploy con altri servizi AWS](#) o [integra AWS CodeDeploy con prodotti e servizi dei partner](#).
3. Utilizza implementazioni blu/verdi per database come [Amazon Aurora](#) e [Amazon RDS](#).
4. [Monitor deployments](#) using Amazon CloudWatch, AWS CloudTrail, and Amazon Simple Notification Service (Amazon SNS) event notifications.
5. Esegui test automatici post-implementazione, inclusi test funzionali, di sicurezza, di regressione, di integrazione e di carico.
6. [Risoluzione dei problemi](#) di implementazione.

Livello di impegno per il piano di implementazione: Medio

## Risorse

Best practice correlate:

- [OPS05-BP02 Test e convalida delle modifiche](#)
- [OPS05-BP09 Applicazione di modifiche frequenti, minime e reversibili](#)
- [OPS05-BP10 Automazione completa dell'integrazione e della distribuzione](#)

Documenti correlati:

- [AWS Builders Library | Automating safe, hands-off deployments | Production deployments](#)
- [AWS Builders Library | My CI/CD pipeline is my release captain | Safe, automatic production releases](#)
- [AWS Whitepaper | Practicing Continuous Integration and Continuous Delivery on AWS | Deployment methods](#)
- [Guida per l'utente di AWS CodeDeploy](#)

- [Utilizzo di configurazioni di distribuzione in AWS CodeDeploy](#)
- [Configurazione della distribuzione di una release Canary di API Gateway](#)
- [Amazon ECS Deployment Types](#)
- [Fully Managed Blue/Green Deployments in Amazon Aurora and Amazon RDS](#)
- [Distribuzioni blu/verde con AWS Elastic Beanstalk](#)

Video correlati:

- [re:Invent 2020 | Hands-off: Automating continuous delivery pipelines at Amazon](#)
- [re:Invent 2019 | Amazon's Approach to high-availability deployment](#)

Esempi correlati:

- [Prova un'implementazione blu/verde di esempio in AWS CodeDeploy](#)
- [Workshop | Building CI/CD pipelines for Lambda canary deployments using AWS CDK](#)
- [Workshop | Blue/Green and Canary Deployment for EKS and ECS](#)
- [Workshop | Building a Cross-account CI/CD Pipeline](#)

## OPS06-BP04 Automazione dei test e del rollback

Per aumentare la velocità, l'affidabilità e la sicurezza del processo di deployment, rendi disponibile una strategia per le funzionalità di test e rollback automatizzate negli ambienti di pre-produzione e produzione. Automatizza i test durante il deployment in produzione per simulare le interazioni umane e di sistema che verificano le modifiche implementate. Automatizza il rollback per tornare rapidamente allo stato precedente corretto noto. Il rollback deve essere avviato automaticamente in condizioni predefinite, ad esempio quando il risultato desiderato della modifica non viene raggiunto o quando il test automatico fallisce. L'automazione di queste due attività migliora la percentuale di successo dei deployment, riduce al minimo i tempi di ripristino e riduce il potenziale impatto sulle attività aziendali.

Risultato desiderato: I test automatici e le strategie di rollback sono integrati nella pipeline di integrazione continua e distribuzione continua (CI/CD). Il monitoraggio è in grado di eseguire la convalida in base ai criteri di successo e avviare il rollback automatico in caso di errore. Ciò riduce al minimo l'impatto sugli utenti finali e sui clienti. Ad esempio, quando tutti i risultati dei test sono stati soddisfatti, promuovi il codice nell'ambiente di produzione in cui vengono avviati i test di regressione

automatizzati, sfruttando gli stessi casi di test. Se i risultati dei test di regressione non corrispondono alle aspettative, viene avviato il rollback automatico nel flusso di lavoro della pipeline.

Anti-pattern comuni:

- I tuoi sistemi non sono progettati in modo da consentire loro di essere aggiornati con release più piccole. Di conseguenza, è difficile annullare tali modifiche di massa (bulk changes) durante un deployment conclusosi con esito negativo.
- Il processo di deployment consiste in una serie di passaggi manuali. Dopo aver distribuito le modifiche al carico di lavoro, inizi i test post-deployment. Dopo il test, ti rendi conto che il tuo carico di lavoro è inutilizzabile e i clienti sono disconnessi. Inizi quindi a eseguire il rollback alla versione precedente. Tutti questi passaggi manuali ritardano il ripristino complessivo del sistema e provocano un impatto prolungato sui clienti.
- Hai impiegato del tempo a sviluppare casi di test automatizzati per funzionalità che non vengono utilizzate frequentemente nella tua applicazione, riducendo al minimo il ritorno sull'investimento nella tua capacità di eseguire test automatizzati.
- La versione è composta da applicazioni, infrastrutture, patch e aggiornamenti di configurazione indipendenti l'uno dall'altro. Tuttavia, è disponibile un'unica pipeline CI/CD che fornisce tutte le modifiche contemporaneamente. Un guasto in un componente obbliga a ripristinare tutte le modifiche, rendendo il rollback complesso e inefficiente.
- Il tuo team completa il lavoro di codifica nello sprint uno e inizia il lavoro dello sprint due, ma il tuo piano non includeva i test fino allo sprint tre. Come conseguenza, i test automatici hanno rivelato difetti dello sprint uno che dovevano essere risolti prima di poter avviare il test dei deliverable dello sprint due e l'intera release viene ritardata, rendendo inutili i test automatizzati.
- I casi di test di regressione automatizzati per la release di produzione sono completi, ma non stai monitorando lo stato del carico di lavoro. Poiché non è possibile verificare se il servizio è stato riavviato o meno, non sei sicuro se il rollback sia necessario o se sia già avvenuto.

Vantaggi dell'adozione di questa best practice: I test automatizzati aumentano la trasparenza del processo di verifica e la capacità di coprire più funzionalità in un periodo di tempo più breve. Testando e convalidando le modifiche nella produzione, è possibile identificare immediatamente i problemi. Il miglioramento della coerenza con strumenti di test automatizzati consente una migliore rilevazione dei difetti. Effettuando automaticamente il rollback alla versione precedente, l'impatto sui clienti viene ridotto al minimo. In ultima analisi, il rollback automatizzato ispira maggiore fiducia nelle capacità di deployment riducendo l'impatto sulle attività aziendali. Nel complesso, queste funzionalità riducono i tempi di consegna garantendo al contempo la qualità.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Automatizza i test degli ambienti distribuiti per verificare che i risultati siano quelli desiderati. Automatizza il rollback a uno stato corretto noto quando non vengono raggiunti i risultati previsti, per ridurre al minimo il tempo di ripristino e gli errori causati dai processi manuali. Integra gli strumenti di test con il flusso di lavoro della pipeline per testare in modo coerente e ridurre al minimo gli input manuali. Dai priorità all'automazione dei casi di test, come quelli che mitigano i rischi maggiori e devono essere testati frequentemente a ogni modifica. Inoltre, automatizza il rollback in base a condizioni specifiche predefinite nel tuo piano di test.

### Passaggi dell'implementazione

1. Stabilisci un ciclo di vita di test per il tuo ciclo di vita di sviluppo che definisca ogni fase del processo di test, dalla pianificazione dei requisiti allo sviluppo dei test case, alla configurazione degli strumenti, ai test automatizzati e alla chiusura dei test case.
  - a. Crea un approccio di test specifico per il carico di lavoro partendo dalla tua strategia di test complessiva.
  - b. Prendi in considerazione una strategia di test continuo, laddove appropriato, durante tutto il ciclo di vita dello sviluppo.
2. Seleziona strumenti automatizzati per il test e il rollback in base ai requisiti aziendali e agli investimenti nella pipeline.
3. Decidi quali casi di test desideri automatizzare e quali devono essere eseguiti manualmente. Questi possono essere definiti in base alla priorità del valore aziendale della funzionalità testata. Allinea tutti i membri del team su questo piano e verifica la responsabilità per l'esecuzione di test manuali.
  - a. Applica le funzionalità di test automatico a casi di test specifici che è opportuno automatizzare, come i casi ripetibili o eseguiti di frequente, quelli che richiedono attività ripetitive o quelli che sono necessari per più configurazioni.
  - b. Definisci gli script di automazione dei test e i criteri di successo nello strumento di automazione in modo da poter avviare l'automazione continua del flusso di lavoro quando casi specifici falliscono.
  - c. Definisci criteri di errore specifici per il rollback automatico.

4. Dai priorità all'automazione dei test per ottenere risultati coerenti con lo sviluppo accurato e completo di casi di test in cui la complessità e l'interazione umana hanno un rischio maggiore di fallimento.
5. Integra i tuoi strumenti di test e rollback automatizzati nella tua pipeline CI/CD.
  - a. Sviluppa criteri di successo chiari per le tue modifiche.
  - b. Monitora e osserva per rilevare questi criteri e annullare automaticamente le modifiche quando vengono soddisfatti criteri di rollback specifici.
6. Esegui diversi tipi di test di produzione automatizzati, come:
  - a. Test A/B, per mostrare i risultati rispetto alla versione corrente tra due gruppi di utenti di test.
  - b. Test Canary, che consente di distribuire la modifica a un sottoinsieme di utenti prima di rilasciarla a tutti.
  - c. Test con flag delle funzionalità, che consente di attivare e disattivare una singola funzionalità della nuova versione alla volta dall'esterno dell'applicazione, in modo che ogni nuova funzionalità possa essere convalidata una alla volta.
  - d. Test di regressione, per verificare nuove funzionalità con componenti correlati esistenti.
7. Monitora gli aspetti operativi dell'applicazione, delle transazioni e delle interazioni con altre applicazioni e componenti. Sviluppa report per mostrare il successo delle modifiche in base al carico di lavoro in modo da poter identificare quali parti dell'automazione e del flusso di lavoro possono essere ulteriormente ottimizzate.
  - a. Sviluppa report sui risultati dei test che ti aiutino a prendere decisioni rapide sull'opportunità o meno di richiamare o meno le procedure di rollback.
  - b. Implementa una strategia che consenta il rollback automatico basato su condizioni di errore predefinite derivanti da uno o più metodi di test.
8. Sviluppa i tuoi casi di test automatizzati per consentire la riutilizzabilità in caso di modifiche ripetibili future.

Livello di impegno per il piano di implementazione: Medio

## Risorse

Best practice correlate:

- [OPS06-BP01 Preparazione di un piano in caso di esito negativo delle modifiche](#)
- [OPS06-BP02 Implementazioni dei test](#)

## Documenti correlati:

- [AWS Builders Library | Ensuring rollback safety during deployments](#)
- [Redeploy and rollback a deployment with AWS CodeDeploy](#)
- [8 best practices when automating your deployments with AWS CloudFormation](#)

## Esempi correlati:

- [Serverless UI testing using Selenium, AWS Lambda, AWS Fargate \(Fargate\), and AWS Developer Tools](#)

## Video correlati:

- [re:Invent 2020 | Hands-off: Automating continuous delivery pipelines at Amazon](#)
- [re:Invent 2019 | Amazon's Approach to high-availability deployment](#)

# Prontezza operativa e gestione delle modifiche

Valuta la prontezza operativa del carico di lavoro, dei processi e delle procedure, nonché del personale, per comprendere i rischi operativi correlati al carico di lavoro. Gestisci il flusso delle modifiche nei tuoi ambienti.

È consigliabile utilizzare un processo omogeneo (inclusi elenchi di controllo manuali o automatici) per sapere quando puoi rilasciare un carico di lavoro o una modifica. Questo inoltre ti aiuterà a trovare le eventuali aree che necessitano di pianificazioni. Disporrai di istruzioni che documentano le tue attività di routine e di manuali che guidano i processi per la risoluzione dei problemi. Utilizza un meccanismo di gestione delle modifiche che sostenga la realizzazione del valore aziendale e contribuisca a mitigare i rischi associati al cambiamento.

## Best practice

- [OPS07-BP01 Verifica della capacità del personale](#)
- [OPS07-BP02 Revisione costante della prontezza operativa](#)
- [OPS07-BP03 Utilizzo di runbook per eseguire le procedure](#)
- [OPS07-BP04 Utilizzo dei playbook per analizzare i problemi](#)
- [OPS07-BP05 Adozione di decisioni informate per implementare sistemi e modifiche](#)

- [OPS07-BP06 Abilitazione dei piani di supporto per i carichi di lavoro di produzione](#)

## OPS07-BP01 Verifica della capacità del personale

Predisponi un meccanismo per stabilire se possiedi il numero appropriato di risorse del personale qualificate per supportare il carico di lavoro. Le risorse devono essere state formate sulla piattaforma e sui servizi che costituiscono il tuo carico di lavoro. Fornisci loro le informazioni necessarie per eseguire il carico di lavoro. Devi avere a disposizione personale qualificato sufficiente per supportare il normale funzionamento del carico di lavoro e gestire gli eventuali incidenti. Predisponi personale sufficiente per la rotazione durante la reperibilità e le ferie per evitare motivi di frustrazione.

Risultato desiderato:

- Presenza di personale qualificato sufficiente per supportare il carico di lavoro nei momenti in cui è disponibile.
- Capacità di fornire al personale formazione sul software e sui servizi che costituiscono il carico di lavoro.

Anti-pattern comuni:

- Implementazione di un carico di lavoro senza membri del team qualificati per l'esecuzione della piattaforma e dei servizi in uso.
- Mancanza di personale sufficiente per supportare la reperibilità a rotazione o le richieste di permesso del personale.

Vantaggi dell'adozione di questa best practice:

- Membri del team qualificati costituiscono un supporto efficace al carico di lavoro.
- Con un numero sufficiente di membri del team, puoi supportare il carico di lavoro e la reperibilità a rotazione, riducendo il rischio di frustrazione.

Livello di rischio associato alla mancata adozione di questa best practice: Elevato

## Guida all'implementazione

Verifica che sia disponibile personale qualificato sufficiente per supportare il carico di lavoro. Assicurati che il numero di membri del team di cui disponi sia sufficiente a coprire le normali attività operative, inclusa la reperibilità a rotazione.

### Esempio del cliente

AnyCompany Retail si assicura che i team che supportano il carico di lavoro includano personale qualificato sufficiente. L'azienda ha al suo interno un numero sufficiente di tecnici per supportare la reperibilità a rotazione. Il personale riceve formazione sul software e sulla piattaforma su cui è basato il carico di lavoro e viene incoraggiato a conseguire certificazioni. Vi è personale sufficiente per permettere alle persone di richiedere permessi di assenza, continuando a supportare il carico di lavoro durante la reperibilità a rotazione.

### Passaggi dell'implementazione

1. Assegna un numero adeguato di risorse del personale per eseguire e supportare il carico di lavoro, tenendo conto della reperibilità.
2. Forma il personale sul software e sulle piattaforme che costituiscono il carico di lavoro.
  - a. [AWS Training and Certification](#) offre una raccolta di corsi su AWS. Sono disponibili corsi gratuiti e a pagamento, online e di persona.
  - b. [AWS organizza eventi e webinar](#) in cui puoi apprendere da esperti AWS.
3. Valuta regolarmente le dimensioni e le competenze del team in base al mutare delle condizioni operative e del carico di lavoro. Adegua le dimensioni e le competenze del team ai requisiti operativi.

Livello di impegno per il piano di implementazione: elevato L'assunzione e la formazione di un team per supportare il carico di lavoro possono richiedere un impegno significativo, ma assicurano solidi vantaggi a lungo termine.

## Risorse

### Best practice correlate:

- [OPS11-BP04 Gestione delle informazioni](#) – I membri del team devono disporre delle informazioni necessarie per eseguire e supportare il carico di lavoro. La gestione delle informazioni è il fattore chiave a questo scopo.

## Documenti correlati:

- [Eventi e webinar AWS](#)
- [AWS Training and Certification](#)

## OPS07-BP02 Revisione costante della prontezza operativa

Usa le revisioni della prontezza operativa (ORR) per verificare che puoi utilizzare il carico di lavoro. ORR è un meccanismo sviluppato da Amazon per verificare che i team possano utilizzare in sicurezza i propri carichi di lavoro. ORR è un processo di revisione e ispezione che utilizza un elenco di controllo per i requisiti. È un'esperienza self-service che i team utilizzano per certificare i propri carichi di lavoro. Le ORR includono le best practice delle lezioni apprese durante gli anni dedicati alla creazione di software.

Un elenco di controllo ORR è composto da suggerimenti sull'architettura, processo operativo, gestione degli eventi e qualità del rilascio. Il nostro processo di correzione dell'errore (CoE, Correction of Error) è uno dei principali fattori trainanti di questi elementi. L'analisi post-incidente deve guidare l'evoluzione della ORR. Una ORR non riguarda solo l'adozione delle best practice, ma anche la prevenzione del ripetersi di eventi già visti. Infine, in una ORR possono essere inclusi anche i requisiti di sicurezza, governance e conformità.

Esegui le ORR prima che un carico di lavoro venga lanciato nella disponibilità generale e quindi durante tutto il ciclo di vita dello sviluppo software. L'esecuzione della ORR prima del lancio aumenta la tua capacità di utilizzare il carico di lavoro in sicurezza. Riesegui periodicamente la ORR sul carico di lavoro per cogliere eventuali scostamenti dalle best practice. Puoi usare gli elenchi di controllo ORR per il lancio di nuovi servizi e le ORR per le revisioni periodiche. In tal modo puoi tenerti aggiornato sulle nuove best practice che emergono e incorporare le lezioni apprese dall'analisi post-incidente. Man mano che l'utilizzo del cloud cresce, puoi creare i requisiti di ORR nella tua architettura come valori predefiniti.

**Risultato desiderato:** hai un elenco di controllo ORR con le best practice per la tua organizzazione. Le ORR vengono eseguite prima dell'avvio dei carichi di lavoro. Le ORR vengono eseguite periodicamente nel corso del ciclo di vita del carico di lavoro.

### Anti-pattern comuni:

- Avvii un carico di lavoro senza sapere se puoi utilizzarlo.

- I requisiti di governance e sicurezza non sono inclusi nella certificazione di un carico di lavoro per l'avvio.
- I carichi di lavoro non vengono rivalutati periodicamente.
- I carichi di lavoro vengono avviati senza le procedure richieste.
- Si osserva la ripetizione di errori con la stessa causa principale in più carichi di lavoro.

Vantaggi dell'adozione di questa best practice:

- I tuoi carichi di lavoro includono le best practice di architettura, processo e gestione.
- Le lezioni apprese sono incorporate nel processo ORR.
- Le procedure richieste sono in atto all'avvio dei carichi di lavoro.
- Le ORR vengono eseguite durante l'intero ciclo di vita del software dei carichi di lavoro.

Livello di rischio se questa best practice non fosse adottata: alto

## Guida all'implementazione

Una ORR è composta da un processo e un elenco di controllo. Il processo ORR deve essere adottato dall'organizzazione e supportato da uno sponsor esecutivo. Come minimo, le ORR devono essere eseguite prima che il carico di lavoro venga lanciato nella disponibilità generale. Esegui la ORR durante tutto il ciclo di vita dello sviluppo software per mantenerlo aggiornato con le best practice o i nuovi requisiti. L'elenco di controllo ORR deve includere elementi di configurazione, requisiti di sicurezza e governance e best practice dell'organizzazione. Nel tempo, puoi utilizzare i servizi, come [AWS Config](#), [AWS Security Hub](#) e [AWS Control Tower Guardrails](#) per creare le best practice dalla ORR nei guardrail per il rilevamento automatico delle best practice.

## Esempio del cliente

Dopo diversi incidenti di produzione, AnyCompany Retail ha deciso di implementare un processo ORR. Ha creato un elenco di controllo composto da best practice, requisiti di governance e conformità e lezioni apprese dalle interruzioni. I nuovi carichi di lavoro conducono le ORR prima dell'avvio. Ogni carico di lavoro esegue una ORR annuale con un sottoinsieme di best practice per incorporare nuove best practice e requisiti che vengono aggiunti all'elenco di controllo ORR. Nel tempo, AnyCompany Retail ha utilizzato [AWS Config](#) per individuare le best practices, accelerando il processo ORR.

## Passaggi dell'implementazione

Per ulteriori informazioni sulle ORR, consulta il [whitepaper Operational Readiness Reviews \(ORR\) \(Revisioni della prontezza operativa \(ORR\)\)](#). Fornisce informazioni dettagliate sulla cronologia del processo ORR, su come creare la procedura ORR e su come sviluppare il proprio elenco di controllo ORR. I passaggi seguenti costituiscono una versione abbreviata di quel documento. Per una comprensione approfondita di cosa sono le ORR e di come crearne una, ti consigliamo di leggere il whitepaper.

1. Riunisci gli stakeholder importanti, inclusi i rappresentanti della sicurezza, delle operazioni e dello sviluppo.
2. Chiedi a ogni stakeholder di indicare almeno un requisito. Per la prima iterazione, prova a limitare il numero di elementi a trenta al massimo.
  - [Appendix B: Example ORR questions \(Appendice B: Domande ORR di esempio\)](#) del whitepaper Operational Readiness Reviews (ORR) (Revisioni della prontezza operativa (ORR)) contiene domande di esempio che puoi utilizzare per iniziare.
3. Raccogli i tuoi requisiti in un foglio di calcolo.
  - Puoi utilizzare [gli obiettivi personalizzati](#) nella funzione [AWS Well-Architected Tool](#) per sviluppare la ORR e condividerla tra i tuoi account e l'organizzazione AWS.
4. Identifica un carico di lavoro su cui condurre la ORR. L'ideale è un carico di lavoro pre-lancio o un carico di lavoro interno.
5. Scorri l'elenco di controllo ORR e prendi nota di tutti i rilevamenti fatti. I rilevamenti potrebbero non essere validi se è in atto una mitigazione. Aggiungi qualsiasi rilevamento privo di mitigazione al tuo backlog di elementi e implementalo prima del lancio.
6. Continua ad aggiungere le best practice e i requisiti all'elenco di controllo ORR nel corso del tempo.

I clienti di AWS Support con supporto Enterprise possono richiedere il [workshop Operational Readiness Review \(Revisione sulla prontezza operativa\)](#) al proprio Technical Account Manager (TAM). Il workshop è una sessione interattiva di lavoro a ritroso per sviluppare il tuo elenco di controllo ORR.

Livello di impegno per il piano di implementazione: alto. L'adozione di una procedura ORR nella tua organizzazione richiede la sponsorizzazione dell'esecutivo e l'adesione degli stakeholder. Crea e aggiorna l'elenco di controllo con input provenienti da tutta l'organizzazione.

## Risorse

### Best practice correlate:

- [OPS01-BP03 Valutazione dei requisiti di governance](#) - I requisiti di governance sono una scelta naturale per un elenco di controllo ORR.
- [OPS01-BP04 Valutazione dei requisiti di conformità](#) - I requisiti di conformità sono talvolta inclusi in un elenco di controllo ORR. Altre volte costituiscono un processo separato.
- [OPS03-BP07 Fornitura di risorse appropriate ai team](#) - La capacità del team è un buon requisito ORR.
- [OPS06-BP01 Preparazione di un piano in caso di esito negativo delle modifiche](#) - Prima di avviare il carico di lavoro, è necessario stabilire un piano di rollback o rollforward.
- [OPS07-BP01 Verifica della capacità del personale](#) - Per supportare un carico di lavoro è necessario disporre del personale necessario.
- [SEC01-BP03 Identificazione e convalida degli obiettivi di controllo](#) - Gli obiettivi di controllo della sicurezza costituiscono eccellenti requisiti ORR.
- [REL13-BP01 Definizione degli obiettivi di ripristino in caso di downtime e perdita di dati](#) - I piani di ripristino di emergenza sono un buon requisito ORR.
- [COST02-BP01 Sviluppo di politiche basate sui requisiti dell'organizzazione](#) - Le policy di gestione dei costi sono utili da includere nell'elenco di controllo ORR.

### Documenti correlati:

- [AWS Control Tower - Guardrails in AWS Control Tower \(Guardrail in AWS Control Tower\)](#)
- [AWS Well-Architected Tool - Custom Lenses \(Obiettivi personalizzati\)](#)
- [Operational Readiness Review Template by Adrian Hornsby \(Modello di revisione della prontezza operativa di Adrian Hornsby\)](#)
- [Whitepaper Operational Readiness Reviews \(ORR\) \(Revisioni della prontezza operativa \(ORR\)\)](#)

### Video correlati:

- [AWS Supports You | Building an Effective Operational Readiness Review \(ORR\) \(AWS ti supporta | Creazione di un'efficace revisione della prontezza operativa \(ORR\)\)](#)

### Esempi correlati:

- [Sample Operational Readiness Review \(ORR\) Lens \(Esempio di obiettivi per la revisione della prontezza operativa \(ORR\)\)](#)

Servizi correlati:

- [AWS Config](#)
- [AWS Control Tower](#)
- [AWS Security Hub](#)
- [AWS Well-Architected Tool](#)

## OPS07-BP03 Utilizzo di runbook per eseguire le procedure

Un runbook è un processo documentato finalizzato al raggiungimento di un determinato risultato. I runbook sono composti da una serie di passaggi che è necessario eseguire per conseguire un obiettivo. L'uso dei runbook può essere fatto risalire agli albori dell'aviazione. Nelle operazioni cloud, è possibile utilizzare i runbook per ridurre i rischi e ottenere i risultati desiderati. In estrema sintesi, un runbook è un elenco di controllo da seguire per completare un'attività.

I runbook costituiscono una parte essenziale del funzionamento dei carichi di lavoro. Dall'inserimento di un nuovo membro in un team all'implementazione di una versione principale, i runbook sono processi codificati che garantiscono risultati coerenti indipendentemente da chi li utilizza. I runbook devono essere pubblicati a livello centralizzato e aggiornati in base all'evoluzione del processo. L'aggiornamento dei runbook rappresenta infatti un elemento chiave dell'intero processo di gestione delle modifiche. Devono inoltre includere le linee guida relative a gestione degli errori, strumenti, autorizzazioni, eccezioni ed escalation in caso di problemi.

A mano a mano che l'organizzazione cresce, è consigliabile automatizzare i runbook. Inizia con runbook concisi e di frequente utilizzo. Utilizza un linguaggio di scripting per automatizzare le procedure o semplificarne l'esecuzione. Dopo aver automatizzato i primi runbook, potrai dedicare altro tempo all'automazione dei runbook più complessi. Gradualmente dovrai automatizzare la maggior parte dei runbook.

Risultato desiderato: il team dispone di una raccolta di linee guida dettagliate per l'esecuzione delle attività relative ai carichi di lavoro. I runbook contengono il risultato desiderato, gli strumenti e le autorizzazioni necessari e le istruzioni per la gestione degli errori. Vengono archiviati in una posizione centralizzata (sistema di controllo delle versioni) e aggiornati di frequente. Ad esempio, i runbook forniscono ai team le funzionalità per monitorare, comunicare e rispondere agli eventi AWS Health

degli account critici durante gli allarmi delle applicazioni, i problemi operativi e gli eventi del ciclo di vita pianificati.

Anti-pattern comuni:

- Ricorso alla memoria per completare i singoli passaggi di un processo.
- Implementazione manuale delle modifiche senza utilizzare un elenco di controllo.
- Vari membri dei team eseguono lo stesso processo con procedure o risultati diversi.
- Mancato aggiornamento dei runbook in base alle modifiche o ai processi di automazione del sistema.

Vantaggi dell'adozione di questa best practice:

- Riduzione della percentuale degli errori per le attività manuali.
- Le operazioni vengono eseguite in modo coerente.
- I nuovi membri dei team possono essere operativi da subito.
- I runbook possono essere automatizzati per semplificare le operazioni più impegnative.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

I runbook possono avere vari formati, a seconda del livello di "maturità" dell'organizzazione. Nella loro formulazione minima, devono essere un documento di testo in cui sono dettagliate le procedure. Il risultato desiderato deve essere indicato in modo chiaro e preciso. Devono inoltre documentare in modo chiaro le autorizzazioni e gli strumenti speciali necessari. Devono includere linee guida dettagliate relative alla gestione degli errori e ai livelli di escalation nel caso in cui si verifichino problemi o errori. I runbook devono riportare il nome del proprietario ed essere pubblicati in una posizione centralizzata. Dopo averlo compilato, un runbook deve essere convalidato. A tale scopo, devi far eseguire il runbook da un membro diverso del tuo team. A mano a mano che la procedura si evolve, aggiorna i runbook in base al processo di gestione delle modifiche.

I runbook in formato testuale devono essere automatizzati a seconda dell'evoluzione dell'organizzazione. Utilizzando servizi come le [automazioni AWS Systems Manager](#), puoi trasformare un testo non formattato in automazioni che possono essere eseguite nell'ambito del carico di lavoro. Queste automazioni possono essere eseguite in risposta a eventi, per ridurre il carico

operativo a salvaguardia del carico di lavoro. L'automazione AWS Systems Manager offre anche [un'esperienza di progettazione visiva](#) low-code per creare più facilmente i runbook di automazione.

## Esempio del cliente

AnyCompany Retail deve eseguire aggiornamenti dello schema del database durante le implementazioni del software. Il team responsabile delle operazioni cloud ha lavorato assieme al team addetto all'amministrazione del database per redigere un runbook per l'implementazione manuale di queste modifiche. Nel runbook sono incluse le procedure dettagliate sotto forma di elenco di controllo. È presente anche una sezione sulla gestione degli errori in caso di problemi. Il runbook è stato pubblicato assieme ad altri runbook sul wiki interno. Il team responsabile delle operazioni cloud pensa di pianificare l'automazione del runbook in futuro.

## Passaggi dell'implementazione

Se non è presente un repository di documenti, è consigliabile creare una libreria di runbook utilizzando un repository per il controllo delle versioni. Puoi creare i runbook utilizzando Markdown. Di seguito è riportato un modello di runbook di esempio che è possibile utilizzare come riferimento per la creazione dei runbook.

```
# Runbook Title
## Runbook Info
| Runbook ID | Description | Tools Used | Special Permissions | Runbook Author | Last
Updated | Escalation POC |
|-----|-----|-----|-----|-----|-----|-----|
| RUN001 | What is this runbook for? What is the desired outcome? | Tools | Permissions
| Your Name | 2022-09-21 | Escalation Name |
## Steps
1. Step one
2. Step two
```

1. Se non disponi di un repository o di un wiki per la documentazione, crea un repository per il controllo delle versioni nel sistema di controllo delle versioni in uso.
2. Individua un processo che non ha un runbook. Un processo ideale è un processo eseguito a cadenza più o meno regolare, con un numero limitato di passaggi e con errori a basso impatto.
3. Nel repository di documenti, crea una nuova bozza di documento Markdown utilizzando il modello. Specifica il titolo del runbook e i campi obbligatori in Informazioni runbook.
4. Partendo dal primo passaggio, compila l'area Passaggi del runbook.

5. Associa il runbook a un membro del team. Chiedi a tale membro di utilizzare il runbook per convalidare i passaggi. In caso di informazioni mancanti o poca chiarezza, aggiorna il runbook.
6. Pubblica il runbook nell'archivio della documentazione interna. Comunica l'avvenuta pubblicazione al team e alle altre parti interessate.
7. In questo modo, nel corso del tempo creerai una libreria di runbook. Man mano che la libreria cresce, comincia a pensare di automatizzare i runbook.

Livello di impegno per il piano di implementazione: basso Lo standard minimo previsto per i runbook è una guida dettagliata in formato testo. L'automazione dei runbook può aumentare l'impegno a livello di implementazione.

## Risorse

Best practice correlate:

- [OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure](#)
- [OPS07-BP04 Utilizzo dei playbook per analizzare i problemi](#)
- [OPS10-BP01 Utilizzo di un processo per la gestione di eventi, incidenti e problemi](#)
- [OPS10-BP02 Definizione di un processo per ogni avviso](#)
- [OPS11-BP04 Gestione delle conoscenze](#)

Documenti correlati:

- [AWS Well-Architected Framework: Concepts: Runbook development](#)
- [Achieving Operational Excellence using automated playbook and runbook](#)
- [AWS Systems Manager: Working with runbooks](#)
- [Migration playbook for AWS large migrations - Task 4: Improving your migration runbooks](#)
- [Use AWS Systems Manager Automation runbooks to resolve operational tasks](#)

Video correlati:

- [AWS re:Invent 2019: DIY guide to runbooks, incident reports, and incident response](#)
- [How to automate IT Operations on AWS | Amazon Web Services](#)
- [Integrate Scripts into AWS Systems Manager](#)

## Esempi correlati:

- [Well-Architected Labs: Automating operations with Playbooks and Runbooks](#)
- [AWS Blog Post: Build a Cloud Automation Practice for Operational Excellence: Best Practices from AWS Managed Services](#)
- [AWS Systems Manager: Automation walkthroughs](#)
- [AWS Systems Manager: Restore a root volume from the latest snapshot runbook](#)
- [Building an AWS incident response runbook using Jupyter notebooks and CloudTrail Lake](#)
- [Gitlab - Runbooks](#)
- [Rubix - A Python library for building runbooks in Jupyter Notebooks](#)
- [Using Document Builder to create a custom runbook](#)

## Servizi correlati:

- [AWS Systems Manager Automation](#)

## OPS07-BP04 Utilizzo dei playbook per analizzare i problemi

I playbook sono guide dettagliate che vengono utilizzate quando si verificano incidenti per analizzare, valutare l'impatto e identificare la causa principale del problema. I playbook sono utili in molti scenari diversi, dalle implementazioni non riuscite agli incidenti di sicurezza. In molti casi, i playbook identificano la causa principale che viene poi mitigata tramite un runbook. I playbook costituiscono un componente essenziale dei piani di risposta agli incidenti di ogni organizzazione.

Un buon playbook include diverse funzionalità chiave che guidano l'utente, passo dopo passo, nel processo di rilevamento. Ma quali passaggi deve eseguire l'utente per diagnosticare un incidente? Illustra chiaramente nel playbook se sono necessari strumenti speciali o autorizzazioni elevate. È essenziale predisporre un piano di comunicazione per aggiornare le parti interessate sullo stato dell'analisi. Nelle situazioni in cui non è possibile identificare la causa principale, il playbook deve prevedere un piano di escalation. Se viene identificata la causa principale, il playbook deve includere il riferimento di un runbook che descrive come risolvere il problema. I playbook devono essere archiviati centralmente e aggiornati regolarmente. Se i playbook vengono utilizzati per avvisi specifici, fornisci al team i riferimenti dei playbook all'interno degli avvisi.

Man mano che l'organizzazione acquisisce maturità, puoi automatizzare i playbook. Inizia con i playbook che trattano incidenti a basso rischio. Utilizza gli script per automatizzare i passaggi di rilevamento. Assicurati di avere i relativi runbook per mitigare le cause principali più comuni.

Risultato desiderato: l'organizzazione dispone dei playbook per gli incidenti comuni. I playbook sono archiviati in una posizione centrale e disponibili per i membri del team. I playbook vengono aggiornati frequentemente. Per qualsiasi causa principale nota, vengono creati i relativi runbook.

Anti-pattern comuni:

- Non esiste un modo standard per analizzare un incidente.
- I membri del team confidano nella "memoria muscolare" o nelle conoscenze istituzionali per risolvere i problemi di un'implementazione non riuscita.
- I nuovi membri del team apprendono come analizzare i problemi attraverso tentativi ed errori.
- Le best practice per l'analisi dei problemi non sono condivise tra i team.

Vantaggi dell'adozione di questa best practice:

- I playbook rendono più efficaci le tue attività per mitigare gli incidenti.
- Uno stesso playbook può essere utilizzato da diversi membri del team in modo da identificare la causa principale in modo coerente.
- Le cause principali note possono già disporre di runbook appositamente sviluppati, accelerando i tempi di ripristino.
- I playbook contribuiscono ad accelerare la collaborazione tra i membri del team.
- I team possono applicare i processi su vasta scala tramite i playbook ripetibili.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Il modo in cui crei e utilizzi i playbook dipende dalla maturità della tua organizzazione. Se non hai familiarità con il cloud, crea i playbook in formato testo in un repository per i documenti centrale. Man mano che l'organizzazione acquisisce maturità, i playbook possono diventare semi automatizzati tramite script scritti in linguaggi come Python. Questi script possono essere eseguiti all'interno di un notebook Jupyter per accelerare il rilevamento. Le organizzazioni avanzate dispongono di playbook completamente automatizzati per i problemi comuni che vengono risolti automaticamente con i runbook.

Inizia a creare i playbook elencando gli incidenti comuni che si verificano nel tuo carico di lavoro. Scegli i playbook per gli incidenti a basso rischio e in cui la causa principale è riconducibile a pochi problemi. Una volta creati i playbook per gli scenari più semplici, passa agli scenari a rischio più elevato o in cui la causa principale non è ancora nota.

I playbook in formato testo vengono automatizzati man mano che l'organizzazione acquisisce maturità. Utilizzando servizi come le [automazioni AWS Systems Manager](#), il testo normale può essere trasformato in automazioni che possono essere eseguite sul carico di lavoro per accelerare le analisi. Queste automazioni possono essere attivate in risposta agli eventi, riducendo il tempo medio per rilevare e risolvere gli incidenti.

I clienti possono utilizzare [AWS Systems Manager Incident Manager](#) per rispondere agli incidenti. Questo servizio fornisce un'unica interfaccia per valutare gli incidenti, informare le parti interessate circa il rilevamento e la mitigazione e collaborare per tutta la durata dell'incidente. Utilizza le automazioni AWS Systems Manager per accelerare il rilevamento e il ripristino.

### Esempio del cliente

Si è verificato un incidente che ha avuto un impatto sulla produzione della società AnyCompany Retail. L'ingegnere di turno utilizza un playbook per analizzare il problema e man mano che esegue i passaggi, mantiene aggiornati gli stakeholder indicati nel playbook. L'ingegnere identifica la causa principale come una race condition di un servizio di back-end. Utilizzando un runbook, l'ingegnere riavvia il servizio e riporta quindi AnyCompany Retail online.

### Passaggi dell'implementazione

Se non è già presente, è consigliabile creare un repository per i documenti con il controllo delle versioni per la libreria di playbook. Puoi creare i tuoi playbook utilizzando Markdown, che è compatibile con la maggior parte dei sistemi di automazione dei playbook. Se parti da zero, utilizza il seguente modello di playbook come esempio.

```
# Playbook Title
## Playbook Info
| Playbook ID | Description | Tools Used | Special Permissions | Playbook Author | Last
Updated | Escalation POC | Stakeholders | Communication Plan |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| RUN001 | What is this playbook for? What incident is it used for? | Tools |
Permissions | Your Name | 2022-09-21 | Escalation Name | Stakeholder Name | How will
updates be communicated during the investigation? |
## Steps
1. Step one
```

## 2. Step two

1. Se non disponi di un repository o di un wiki per i documenti, crea nel sistema di controllo delle versioni in uso un nuovo repository con il controllo delle versioni per i tuoi playbook.
2. Identifica un problema comune che richieda un'analisi, vale a dire uno scenario in cui la causa principale è riconducibile a pochi problemi e la risoluzione è a basso rischio.
3. Utilizzando il modello Markdown, compila la sezione Titolo del playbook e i campi in Informazioni sul playbook.
4. Includi i passaggi per la risoluzione dei problemi. Illustra nel modo più chiaro possibile le azioni da eseguire o le aree da analizzare.
5. Chiedi a un membro del team di esaminare e convalidare il tuo playbook. Se manca un'informazione o è necessario un chiarimento, aggiorna il playbook.
6. Pubblica il tuo playbook nel repository per i documenti e informa il tuo team e tutte le parti interessate.
7. Questa libreria diventerà sempre più ricca man mano che aggiungi altri playbook. Una volta che sono disponibili diversi playbook, inizia ad automatizzarli con strumenti come le automazioni AWS Systems Manager per mantenere sincronizzati l'automazione e i playbook.

Livello di impegno per il piano di implementazione: basso I playbook sono documenti di testo archiviati in una posizione centrale. Le organizzazioni che hanno acquisito maturità applicano l'automazione dei playbook.

## Risorse

Best practice correlate:

- [OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure](#)
- [OPS07-BP03 Utilizzo di runbook per eseguire le procedure](#)
- [OPS10-BP01 Utilizzo di un processo per la gestione di eventi, incidenti e problemi](#)
- [OPS10-BP02 Definizione di un processo per ogni avviso](#)
- [OPS11-BP04 Gestione delle conoscenze](#)

Documenti correlati:

- [AWS Well-Architected Framework: Concepts: Playbook development](#)

- [Achieving Operational Excellence using automated playbook and runbook](#)
- [AWS Systems Manager: Working with runbooks](#)
- [Use AWS Systems Manager Automation runbooks to resolve operational tasks](#)

#### Video correlati:

- [AWS re:Invent 2019: DIY guide to runbooks, incident reports, and incident response \(SEC318-R1\)](#)
- [AWS Systems Manager Incident Manager - AWS Virtual Workshops](#)
- [Integrate Scripts into AWS Systems Manager](#)

#### Esempi correlati:

- [AWS Customer Playbook Framework](#)
- [AWS Systems Manager: Automation walkthroughs](#)
- [Building an AWS incident response runbook using Jupyter notebooks and CloudTrail Lake](#)
- [Rubix – A Python library for building runbooks in Jupyter Notebooks](#)
- [Using Document Builder to create a custom runbook](#)
- [Well-Architected Labs: Automating operations with Playbooks and Runbooks](#)
- [Well-Architected Labs: Incident response playbook with Jupyter](#)

#### Servizi correlati:

- [AWS Systems Manager Automation](#)
- [AWS Systems Manager Incident Manager](#)

## OPS07-BP05 Adozione di decisioni informate per implementare sistemi e modifiche

Predisponi processi per la gestione delle modifiche efficaci e infruttuose al carico di lavoro. Si definisce "pre-mortem" un esercizio in cui il team simula un errore per sviluppare strategie di mitigazione. Utilizza questo esercizio per prevedere errori e creare procedure ove opportuno. Valuta i vantaggi e i rischi dell'implementazione di modifiche nel carico di lavoro. Verifica che tutte le modifiche siano conformi ai requisiti di governance.

## Risultato desiderato:

- Adozione di decisioni informate durante l'implementazione di modifiche nel carico di lavoro.
- Le modifiche sono conformi ai requisiti di governance.

## Anti-pattern comuni:

- Implementazione di una modifica nel carico di lavoro senza un processo per la gestione di un'implementazione errata.
- Applicazione di modifiche all'ambiente di produzione che non sono conformi ai requisiti di governance.
- Implementazione di una nuova versione del carico di lavoro senza stabilire valori di riferimento per l'utilizzo delle risorse.

## Vantaggi dell'adozione di questa best practice:

- L'azienda è preparata all'effetto di modifiche infruttuose al carico di lavoro.
- Le modifiche apportate al carico di lavoro sono conformi ai criteri di governance.

Livello di rischio associato alla mancata adozione di questa best practice: basso

## Guida all'implementazione

Usa esercizi pre-mortem per sviluppare processi per la gestione di modifiche infruttuose. Documenta i processi di gestione delle modifiche infruttuose. Verifica che tutte le modifiche siano conformi ai requisiti di governance. Valuta i vantaggi e i rischi dell'implementazione di modifiche nel carico di lavoro.

## Esempio del cliente

AnyCompany Retail svolge regolarmente esercizi pre-mortem per convalidare i propri processi di gestione delle modifiche infruttuose. L'azienda documenta i propri processi in un Wiki condiviso che aggiorna spesso. Tutte le modifiche sono conformi ai requisiti di governance.

## Passaggi dell'implementazione

1. Prendi decisioni informate durante l'implementazione di modifiche nel carico di lavoro. Definisci ed esamina i criteri per un'implementazione corretta. Sviluppa scenari o criteri che attiverrebbero

il ripristino dello stato precedente a una modifica. Soppesa i vantaggi dell'implementazione di modifiche rispetto ai rischi di una modifica infruttuosa.

2. Verifica che tutte le modifiche siano conformi ai requisiti di governance.
3. Usa esercizi pre-mortem per pianificare la gestione delle modifiche infruttuose e documentare le strategie di mitigazione. Esegui un esercizio di simulazione di un'emergenza per modellare una modifica infruttuosa e convalidare le procedure di ripristino dello stato precedente.

Livello di impegno per il piano di implementazione: moderato. L'implementazione di una procedura di pre-mortem richiede il coordinamento e l'impegno degli stakeholder in tutta l'organizzazione

## Risorse

Best practice correlate:

- [OPS01-BP03 Valutazione dei requisiti di governance](#) – I requisiti di governance sono un fattore chiave per determinare se implementare una modifica.
- [OPS06-BP01 Preparazione di un piano in caso di esito negativo delle modifiche](#) – Predisponi piani per mitigare un'implementazione non riuscita e usa esercizi di pre-mortem per convalidarli.
- [OPS06-BP02 Implementazioni dei test](#) – Ogni modifica software deve essere testata nel modo adeguato prima dell'implementazione per ridurre gli errori nell'ambiente di produzione.
- [OPS07-BP01 Verifica della capacità del personale](#) – La presenza di personale qualificato sufficiente per supportare il carico di lavoro è essenziale per prendere una decisione informata riguardo all'implementazione di una modifica di sistema.

Documenti correlati:

- [Amazon Web Services: rischio e conformità](#)
- [Modello di responsabilità condivisa AWS](#)
- [Governance nel Cloud AWS: il giusto equilibrio tra agilità e sicurezza](#)

## OPS07-BP06 Abilitazione dei piani di supporto per i carichi di lavoro di produzione

Abilita il supporto per qualsiasi software e servizio a cui si affida il tuo carico di lavoro di produzione. Seleziona un livello di supporto adeguato per soddisfare le esigenze di assistenza della produzione.

I piani di supporto per queste dipendenze sono necessari nel caso si verifichi un'interruzione del servizio o un problema di software. Documenta i piani di supporto e come chiedere assistenza per tutti i servizi e i fornitori di software. Implementa meccanismi di verifica per controllare che i riferimenti del supporto siano aggiornati.

Risultato desiderato:

- Implementa piani di supporto per software e servizi a cui si affidano i carichi di lavoro di produzione.
- Scegli un piano di supporto adeguato in base alle esigenze di assistenza.
- Documenta i piani e i livelli di supporto e come richiedere assistenza.

Anti-pattern comuni:

- Non hai piani di supporto per un fornitore software strategico. Il tuo carico di lavoro è coinvolto e non puoi fare nulla per accelerare un intervento risolutivo o per ricevere aggiornamenti tempestivi dal fornitore.
- Uno sviluppatore, che era il punto di contatto primario di un fornitore di software, ha lasciato l'azienda. Non puoi contattare direttamente l'assistenza del fornitore. Devi investire il tuo tempo per cercare le informazioni e orientarti tra sistemi di contatto generici, aumentando così il livello di impegno richiesto per intervenire quando necessario.
- Si verifica un'interruzione della produzione con un fornitore di software. Non esiste una documentazione su come inserire una richiesta di assistenza.

Vantaggi dell'adozione di questa best practice:

- Con il livello di supporto adeguato, puoi ottenere una risposta nei tempi previsti per soddisfare le esigenze in termini di livelli di servizio.
- In caso di problemi in produzione, puoi inoltrare il problema se sei un cliente assistito.
- Fornitori di software e servizi possono essere di aiuto per la risoluzione dei problemi durante un incidente.

Livello di rischio associato se questa best practice non fosse adottata: basso

## Guida all'implementazione

Abilita i piani di supporto per qualsiasi fornitore di software e servizi a cui si affida il tuo carico di lavoro di produzione. Configura piani di supporto adeguati per soddisfare le esigenze di assistenza. Per i clienti AWS, questo significa abilitare il supporto Business di AWS o superiore su qualsiasi account su cui hai carichi di lavoro di produzione. Incontra con regolarità i fornitori del servizio di assistenza per ricevere aggiornamenti sulle offerte di supporto, sui processi e sui contatti. Documenta come richiedere assistenza ai fornitori di software e servizi, incluso come inoltrare il problema in caso si verificasse un'interruzione. Implementa meccanismi di aggiornamento dei contatti del supporto.

### Esempio del cliente

In AnyCompany Retail, tutte le dipendenze di servizi e software commerciali hanno piani di supporto. Ad esempio, hanno il supporto Enterprise di AWS abilitato su tutti gli account con carichi di lavoro di produzione. In caso di problemi, qualsiasi sviluppatore può inserire una richiesta di assistenza. Esiste una pagina wiki con informazioni su come richiedere assistenza, chi contattare e quali best practice seguire per accelerare il processo di risoluzione.

### Passaggi dell'implementazione

1. Lavora con le parti interessate all'interno della tua organizzazione per identificare i fornitori di software e servizi su cui si basa il tuo carico di lavoro. Documenta queste dipendenze.
2. Stabilisci le esigenze in termini di assistenza del tuo carico di lavoro. Seleziona un piano di supporto in linea con tali esigenze.
3. Per software e servizi commerciali definisci un piano di supporto con i fornitori.
  - a. Sottoscrivere il supporto Business di AWS o un livello superiore per tutti gli account di produzione garantisce tempi di risposta più rapidi da AWS Support ed è una scelta fortemente consigliata. Se non hai il supporto premium, devi avere un piano di azione per gestire i problemi, che richiede l'aiuto di AWS Support. AWS Support offre un mix di strumenti e tecnologie, persone e programmi progettati per aiutarti in modo proattivo a ottimizzare le performance, ridurre i costi e innovare più rapidamente. Il supporto Business di AWS offre vantaggi aggiuntivi, tra cui l'accesso a AWS Trusted Advisor e ad AWS Personal Health Dashboard, nonché tempi di risposta più rapidi.
4. Documenta il tuo piano di supporto nello strumento di gestione delle conoscenze. Includi come richiedere assistenza, chi avvertire se viene inviata una richiesta di assistenza e come inoltrare il problema durante un incidente. Un wiki è un buon meccanismo che consente a tutti di apportare gli aggiornamenti necessari alla documentazione, nel momento in cui vengono a conoscenza di modifiche a processi o contatti del supporto.

Livello di impegno per il piano di implementazione: Basso. La maggior parte di fornitori di servizi e software offre piani di supporto da attivare. Documentando e condividendo le best practice di supporto sul tuo sistema di gestione delle conoscenze puoi verificare che il tuo team sappia cosa fare quando si verifica un problema in produzione.

## Risorse

Best practice correlate:

- [OPS02-BP02 Assegnazione di proprietari identificati a processi e procedure](#)

Documenti correlati:

- [Piani AWS Support](#)

Servizi correlati:

- [Supporto del Business AWS](#)
- [Supporto Enterprise AWS](#)

# Operatività

Il successo consiste nel raggiungimento dei risultati aziendali misurati in base ai parametri che definisci. Comprendendo lo stato di integrità del carico di lavoro e delle operazioni, puoi stabilire quando i risultati organizzativi e aziendali possono risultare o diventare a rischio, e rispondere in modo appropriato.

Per avere successo, devi essere in grado di:

## Argomenti

- [Utilizzare l'osservabilità del carico di lavoro](#)
- [Comprensione dello stato operativo](#)
- [Risposta agli eventi](#)

## Utilizzare l'osservabilità del carico di lavoro

Garantire l'integrità del carico di lavoro sfruttando l'osservabilità. Utilizzare metriche, log e tracce pertinenti per ottenere una visione completa delle prestazioni del carico di lavoro e risolvere i problemi in modo efficiente.

L'osservabilità ti consente di concentrarti su dati significativi e di comprendere le interazioni e l'output del tuo carico di lavoro. Concentrandoti sugli approfondimenti essenziali ed eliminando i dati non necessari, mantieni un approccio diretto alla comprensione delle prestazioni del carico di lavoro.

È essenziale non solo raccogliere dati, ma anche interpretarli correttamente. Definisci linee guida chiare, imposta soglie di avviso appropriate e monitora attivamente eventuali deviazioni. Un cambiamento in una metrica chiave, specialmente se correlata ad altri dati, permette di individuare aree problematiche specifiche.

Grazie all'osservabilità hai strumenti per prevedere e affrontare potenziali sfide, assicurando che il tuo carico di lavoro funzioni senza intoppi e soddisfi le esigenze aziendali.

AWS offre strumenti specifici come [Amazon CloudWatch](#) per il monitoraggio e la registrazione e [AWS X-Ray](#) per il tracciamento distribuito. Questi servizi si integrano facilmente con varie risorse AWS, consentendo un'efficiente raccolta dei dati, impostazione di avvisi basati su soglie predefinite e presentazione dei dati su dashboard per una facile interpretazione. Sfruttando queste informazioni dettagliate, puoi prendere decisioni ben informate e basate sui dati in linea con i tuoi obiettivi operativi.

## Best practice

- [OPS08-BP01 Analisi delle metriche del carico di lavoro](#)
- [OPS08-BP02 Analizza i log relativi ai carichi di lavoro](#)
- [OPS08-BP03 Analisi delle tracce del carico di lavoro](#)
- [OPS08-BP04 Creare avvisi fruibili](#)
- [OPS08-BP05 Creare dashboard](#)

## OPS08-BP01 Analisi delle metriche del carico di lavoro

Dopo aver implementato la telemetria dell'applicazione, analizza regolarmente le metriche raccolte. Sebbene latenza, richieste, errori e capacità (o quote) forniscano informazioni dettagliate sulle prestazioni del sistema, è fondamentale dare priorità alla revisione delle metriche relative ai risultati aziendali. Ciò ti assicura di prendere decisioni basate sui dati in linea con i tuoi obiettivi aziendali.

Risultato desiderato: Informazioni dettagliate sulle prestazioni del carico di lavoro che guidano decisioni basate sui dati, garantendo l'allineamento con gli obiettivi aziendali.

### Anti-pattern comuni:

- Analisi isolata delle metriche senza considerare il loro impatto sui risultati aziendali.
- Eccessiva dipendenza dalle metriche tecniche trascurando quelle aziendali.
- Revisione poco frequente delle metriche, perdita di opportunità di prendere decisioni in tempo reale.

### Vantaggi dell'adozione di questa best practice:

- Comprensione migliorata della correlazione tra prestazioni tecniche e risultati aziendali.
- Processo decisionale migliorato basato su dati in tempo reale.
- Identificazione e mitigazione proattive dei problemi prima che influiscano sui risultati aziendali.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Sfrutta strumenti come Amazon CloudWatch per l'esecuzione di analisi delle metriche. Utilizza servizi AWS come AWS Cost Anomaly Detection e Amazon DevOps Guru per rilevare anomalie, soprattutto

quando le soglie statiche non sono conosciute o quando i modelli di comportamento evidenziano possibili anomalie.

## Passaggi dell'implementazione

1. Analizza e revisiona: revisiona e interpreta regolarmente le metriche relative al carico di lavoro.
  - a. Dai priorità alle metriche relative ai risultati aziendali rispetto a quelle puramente tecniche.
  - b. Comprendi l'importanza di picchi, cali o schemi nei dati.
2. Utilizza Amazon CloudWatch: utilizza Amazon CloudWatch per una visualizzazione centralizzata e un'analisi approfondita.
  - a. Configura dashboard CloudWatch per visualizzare le tue metriche e confrontarle nel tempo.
  - b. Utilizza [percentili in CloudWatch](#) per avere una visione chiara della distribuzione delle metriche, il che può aiutarti a definire gli SLA e a identificare valori anomali.
  - c. Configura [AWS Cost Anomaly Detection](#) per identificare modelli insoliti senza fare affidamento su soglie statiche.
  - d. Implementa [l'osservabilità CloudWatch tra account](#) per monitorare e risolvere i problemi delle applicazioni che si estendono su più account all'interno di una regione.
  - e. Utilizza [gli approfondimenti sulle metriche CloudWatch](#) per interrogare e analizzare i dati delle metriche tra account e regioni, identificando tendenze e anomalie.
  - f. Applica [Metrica matematica CloudWatch](#) per trasformare, aggregare o eseguire calcoli sulle metriche per ottenere informazioni più approfondite.
3. Impiega Amazon DevOps Guru: incorpora [Amazon DevOps Guru](#) per il rilevamento delle anomalie basato sul machine learning, che consente di identificare i primi segnali di problemi operativi che riguardano le applicazioni serverless e di correggerli prima che abbiano un impatto sui clienti.
4. Ottimizza in base agli approfondimenti: prendi decisioni informate sulla base dell'analisi delle metriche per adeguare e migliorare i carichi di lavoro.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS04-BP02 Implementazione della telemetria dell'applicazione](#)

## Documenti correlati:

- [The Wheel Blog - Emphasizing the importance of continually reviewing metrics](#)
- [Percentile are important](#)
- [Utilizzo di AWS Cost Anomaly Detection](#)
- [CloudWatch cross-account observability](#)
- [Query your metrics with CloudWatch Metrics Insights](#)

## Video correlati:

- [Enable Cross-Account Observability in Amazon CloudWatch](#)
- [Introduction to Amazon DevOps Guru](#)
- [Continuously Analyze Metrics using AWS Cost Anomaly Detection](#)

## Esempi correlati:

- [One Observability Workshop](#)
- [Gaining operation insights with AIOps using Amazon DevOps Guru](#)

## OPS08-BP02 Analizza i log relativi ai carichi di lavoro

L'analisi regolare dei log dei carichi di lavoro è essenziale per acquisire una comprensione più approfondita degli aspetti operativi dell'applicazione. Attraverso l'analisi, la consultazione e l'interpretazione efficiente dei dati di log, è possibile ottimizzare continuamente le prestazioni e la sicurezza delle applicazioni.

Risultato desiderato: approfondimenti sul comportamento dell'applicazione e sulle operazioni derivanti da un'analisi completa dei log, che garantisce il rilevamento e la mitigazione proattiva dei problemi.

### Anti-pattern comuni:

- Si trascura l'analisi dei log fino a quando non si verifica un problema critico.
- Il mancato utilizzo della suite completa degli strumenti disponibili per l'analisi dei log comporta la perdita di approfondimenti importanti.

- Si fa affidamento esclusivamente sulla revisione manuale dei log senza sfruttare le funzionalità di automazione e query.

Vantaggi dell'adozione di questa best practice:

- Identificazione proattiva dei colli di bottiglia operativi, delle minacce alla sicurezza e di altri problemi potenziali.
- Utilizzo efficiente dei dati di log per l'ottimizzazione continua dell'applicazione.
- Comprensione migliorata del comportamento dell'applicazione, facilitando il debug e la risoluzione dei problemi.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

[Amazon CloudWatch Logs](#) è un potente strumento per l'analisi dei log. Le funzionalità integrate come CloudWatch Logs Insights e Contributor Insights rendono intuitivo ed efficiente il processo di acquisizione di approfondimenti significativi dai log.

### Passaggi dell'implementazione

1. Configura CloudWatch Logs: configura applicazioni e servizi per inviare log a CloudWatch Logs.
2. Usa il rilevamento delle anomalie nei log: utilizza il [rilevamento delle anomalie Amazon CloudWatch Logs](#) per identificare e avvisare automaticamente se si verificano modelli di log insoliti. Questo strumento consente di gestire in modo proattivo le anomalie nei log e di rilevare tempestivamente i potenziali problemi.
3. Configura CloudWatch Logs Insights: usa [CloudWatch Logs Insights](#) per cercare e analizzare in modo interattivo i dati di log.
  - a. Crea query per estrarre modelli, visualizzare i dati di log e ricavare approfondimenti utili.
  - b. Usa l'[analisi dei modelli CloudWatch Logs Insights](#) per analizzare e visualizzare i modelli di log frequenti. Questa funzionalità consente di comprendere le tendenze operative più comuni e i potenziali valori anomali nei dati di log.
  - c. Usa il [confronto \(diff\) di CloudWatch Logs](#) per eseguire analisi differenziali tra diversi periodi di tempo o diversi gruppi di log. Questa funzionalità ti consente di individuare le modifiche e valutarne l'impatto sulle prestazioni o sul comportamento del sistema.

4. Monitora i log in tempo reale con Live Tail: usa [Amazon CloudWatch Logs Live Tail](#) per visualizzare i dati di log in tempo reale. Puoi monitorare attivamente le attività operative dell'applicazione man mano che si verificano, ottenendo una visibilità immediata sulle prestazioni del sistema e sui potenziali problemi.
5. Utilizza Contributor Insights: utilizza [CloudWatch Contributor Insights](#) per identificare i top talker in dimensioni ad alta cardinalità come gli indirizzi IP o gli utenti-agenti.
6. Implementa filtri di metriche CloudWatch Logs: configura i [filtri di metriche CloudWatch Logs](#) per convertire i dati di log in metriche fruibili. Puoi così impostare allarmi o analizzare ulteriormente i modelli.
7. Implementa [l'osservabilità tra account CloudWatch](#): monitora e risolvi i problemi delle applicazioni che si estendono su più account all'interno di una regione.
8. Rivedi regolarmente e perfeziona: rivedi periodicamente le tue strategie di analisi dei log per acquisire tutte le informazioni pertinenti e ottimizzare continuamente le prestazioni delle applicazioni.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS04-BP02 Implementazione della telemetria dell'applicazione](#)
- [OPS08-BP01 Analisi delle metriche del carico di lavoro](#)

Documenti correlati:

- [Analyzing Log Data with CloudWatch Logs Insights](#)
- [Using CloudWatch Contributor Insights](#)
- [Creating and Managing CloudWatch Log Metric Filters](#)

Video correlati:

- [Analyze Log Data with CloudWatch Logs Insights](#)
- [Use CloudWatch Contributor Insights to Analyze High-Cardinality Data](#)

Esempi correlati:

- [CloudWatch Logs Sample Queries](#)
- [One Observability Workshop](#)

## OPS08-BP03 Analisi delle tracce del carico di lavoro

L'analisi dei dati di tracciamento è fondamentale per ottenere una visione completa del percorso operativo di un'applicazione. Visualizzando e comprendendo le interazioni tra i vari componenti, è possibile ottimizzare le prestazioni, identificare i colli di bottiglia e migliorare l'esperienza utente.

Risultato desiderato: ottieni una chiara visibilità sulle operazioni distribuite della tua applicazione, che si traduce in una risoluzione più rapida dei problemi e in un'esperienza utente migliorata.

Anti-pattern comuni:

- I dati di tracciamento vengono trascurati e ci si affida esclusivamente a log e metriche.
- I dati di tracciamento non sono correlati ai log associati.
- Vengono ignorate le metriche derivate dalle tracce, come la latenza e i tassi di errore.

Vantaggi dell'adozione di questa best practice:

- Miglioramento della risoluzione dei problemi e riduzione del tempo medio di risoluzione (MTTR).
- Informazioni dettagliate sulle dipendenze e sul loro impatto.
- Identificazione e correzione rapide dei problemi di prestazione.
- Vengono sfruttate le metriche derivate dalle tracce per un processo decisionale informato.
- Esperienze utente migliorate attraverso interazioni con i componenti ottimizzate.

Livello di rischio associato se questa best practice non fosse adottata: medio

### Guida all'implementazione

[AWS X-Ray](#) offre una suite completa per l'analisi dei dati di tracciamento, fornendo una visione olistica delle interazioni con i servizi, monitorando le attività degli utenti e rilevando i problemi di prestazioni. Funzionalità come ServiceLens, X-Ray Insights, X-Ray Analytics e Amazon DevOps Guru permettono di ottenere informazioni fruibili più approfondite derivate dai dati di tracciamento.

## Passaggi dell'implementazione

I seguenti passaggi offrono un approccio strutturato per implementare efficacemente l'analisi dei dati di tracciamento utilizzando i servizi AWS:

1. **Integra AWS X-Ray:** assicurati che X-Ray sia integrato con le tue applicazioni per acquisire dati di tracciamento.
2. **Analizza le metriche X-Ray:** analizza le metriche derivate dalle tracce di X-Ray, come latenza, tassi di richiesta, percentuale di errore e distribuzione dei tempi di risposta utilizzando la [mappa dei servizi](#) per monitorare lo stato delle applicazioni.
3. **Usa ServiceLens:** utilizza la [mappa di ServiceLens](#) per ottenere una maggiore osservabilità dei servizi e delle applicazioni. Fornisce la visualizzazione integrata di tracce, metriche, log, allarmi e altre informazioni correlate all'integrità.
4. **Abilita X-Ray Insights:**
  - a. attiva [X-Ray Insights](#) per il rilevamento automatico delle anomalie nelle tracce.
  - b. Esamina gli approfondimenti per individuare i modelli e determinare le cause ultime, come l'aumento dei tassi di errore o delle latenze.
  - c. Consulta la cronologia degli approfondimenti per un'analisi cronologica dei problemi rilevati.
5. **Usa X-Ray Analytics:** [X-Ray Analytics](#) consente di esplorare a fondo i dati di tracciamento, individuare modelli ed estrarre approfondimenti.
6. **Usa i gruppi in X-Ray:** crea i gruppi in X-Ray per filtrare le tracce in base a criteri come l'elevata latenza, per eseguire un'analisi più mirata.
7. **Incorpora Amazon DevOps Guru:** integra [Amazon DevOps Guru](#) per trarre vantaggio dai modelli di machine learning che individuano le anomalie operative nelle tracce.
8. **Usa CloudWatch Synthetics:** usa [CloudWatch Synthetics](#) per creare canary per il monitoraggio continuo di endpoint e flussi di lavoro. Questi canary possono integrarsi con X-Ray per fornire dati di tracciamento per un'analisi approfondita delle applicazioni testate.
9. **Usa Real User Monitoring (RUM):** con [AWS X-Ray e CloudWatch RUM](#) puoi analizzare ed eseguire il debug del percorso della richiesta a partire dagli utenti finali dell'applicazione tramite servizi AWS gestiti a valle. Ciò ti aiuta a identificare le tendenze e gli errori di latenza che hanno un impatto sugli utenti finali.
10. **Esegui correlazioni con i log:** esegui correlazioni [tra i dati di tracciamento e i relativi log](#) all'interno della vista di tracce di X-Ray per una prospettiva granulare del comportamento delle applicazioni. Ciò consente di visualizzare gli eventi di log associati direttamente alle transazioni tracciate.

11 Implementa [l'osservabilità tra account CloudWatch](#): monitora e risolvi i problemi delle applicazioni che si estendono su più account all'interno di una regione.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS08-BP01 Analisi delle metriche del carico di lavoro](#)
- [OPS08-BP02 Analizza i log relativi ai carichi di lavoro](#)

Documenti correlati:

- [Using ServiceLens to Monitor Application Health](#)
- [Exploring Trace Data with X-Ray Analytics](#)
- [Detecting Anomalies in Traces with X-Ray Insights](#)
- [Continuous Monitoring with CloudWatch Synthetics](#)

Video correlati:

- [Analyze and Debug Applications Using Amazon CloudWatch Synthetics & AWS X-Ray](#)
- [Use AWS X-Ray Insights](#)

Esempi correlati:

- [One Observability Workshop](#)
- [Implementing X-Ray with AWS Lambda](#)
- [CloudWatch Synthetics Canary Templates](#)

## OPS08-BP04 Creare avvisi fruibili

Rilevare e rispondere tempestivamente alle deviazioni di comportamento dell'applicazione è fondamentale. È importante riconoscere quando i risultati basati sugli indicatori chiave di prestazione (KPI) sono a rischio o quando si verificano anomalie impreviste. Basare gli avvisi sui KPI garantisce che i segnali ricevuti siano direttamente correlati all'impatto aziendale od operativo. Questo approccio

verso avvisi fruibili promuove risposte proattive e aiuta a mantenere le prestazioni e l'affidabilità del sistema.

Risultato desiderato: si ricevono avvisi tempestivi, pertinenti e fruibili per l'identificazione e la mitigazione rapida di potenziali problemi, soprattutto quando i risultati dei KPI sono a rischio.

Anti-pattern comuni:

- Si impostano troppi avvisi non critici, con conseguente affaticamento da avvisi ("alert fatigue").
- Non viene data priorità agli avvisi in base ai KPI, il che rende difficile comprendere l'impatto dei problemi sull'azienda.
- Non affrontare le cause principali porta a ricevere avvisi ripetuti per lo stesso problema.

Vantaggi dell'adozione di questa best practice:

- Riduzione dell'affaticamento da avvisi ("alert fatigue") concentrandosi su avvisi pertinenti e fruibili.
- Maggiore operatività e affidabilità del sistema grazie al rilevamento e alla mitigazione proattiva dei problemi.
- Migliore collaborazione tra team e risoluzione più rapida dei problemi grazie all'integrazione con i più diffusi strumenti di avviso e comunicazione.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

Per creare un meccanismo di avviso efficace, è fondamentale utilizzare metriche, log e dati di tracciamento che segnalino quando i risultati basati sui KPI sono a rischio o vengono rilevate anomalie.

Passaggi dell'implementazione

1. Determina gli indicatori chiave di prestazione (KPI): identifica i KPI dell'applicazione. Gli avvisi devono essere correlati a questi KPI per riflettere accuratamente l'impatto aziendale.
2. Implementa il rilevamento delle anomalie:
  - Usa il rilevamento delle anomalie Amazon CloudWatch: configura il [rilevamento delle anomalie Amazon CloudWatch](#) per rilevare automaticamente modelli insoliti e generare avvisi solo per anomalie reali.
  - Utilizza AWS X-Ray Insights:

- a. Configura [X-Ray Insights](#) per rilevare anomalie nei dati di tracciamento.
- b. Configura [le notifiche per X-Ray Insights](#) per ricevere avvisi quando si rilevano problemi.
- Esegui l'integrazione con Amazon DevOps Guru:
  - a. Utilizza [Amazon DevOps Guru](#) e le sue capacità di machine learning per rilevare anomalie operative nei dati esistenti.
  - b. Accedi alle [impostazioni di notifica](#) in DevOps Guru per configurare gli avvisi per le anomalie.
3. Implementa avvisi fruibili: progetta avvisi che forniscano informazioni adeguate per intraprendere un'azione immediata.
  1. Monitora [gli eventi AWS Health con le regole Amazon EventBridge](#) o integra a livello di programmazione l'API AWS Health per automatizzare le azioni quando ricevi eventi AWS Health. Possono essere azioni generali, come l'invio di tutti i messaggi pianificati sugli eventi del ciclo di vita a un'interfaccia di chat, oppure azioni specifiche, come l'avvio di un flusso di lavoro in uno strumento di gestione dei servizi IT.
4. Riduci l'affaticamento da avvisi: riduci al minimo gli avvisi non critici. Quando i team sono sovraccaricati da numerosi avvisi insignificanti, possono trascurare i problemi critici, riducendo l'efficacia complessiva del meccanismo di avviso.
5. Configura allarmi compositi: utilizza [gli allarmi compositi Amazon CloudWatch](#) per raggruppare più allarmi.
6. Integra strumenti di avviso: incorpora strumenti come [Ops Genie](#) e [PagerDuty](#).
7. Integra AWS Chatbot: integra [AWS Chatbot](#) per inoltrare avvisi a Amazon Chime, Microsoft Teams e Slack.
8. Usa l'avviso basato sui log: utilizza i [filtri delle metriche dei log](#) in CloudWatch per creare allarmi basati su eventi di log specifici.
9. Rivedi e itera: riesamina e perfeziona regolarmente le configurazioni degli avvisi.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS04-BP02 Implementazione della telemetria dell'applicazione](#)
- [OPS04-BP03 Implementazione della telemetria dell'esperienza utente](#)

- [OPS04-BP04 Implementazione della telemetria delle dipendenze](#)
- [OPS04-BP05 Implementazione del tracciamento distribuito](#)
- [OPS08-BP01 Analisi delle metriche del carico di lavoro](#)
- [OPS08-BP02 Analizza i log relativi ai carichi di lavoro](#)
- [OPS08-BP03 Analisi delle tracce del carico di lavoro](#)

#### Documenti correlati:

- [Using Amazon CloudWatch alarms](#)
- [Create a composite alarm](#)
- [Create a CloudWatch alarm based on anomaly detection](#)
- [DevOps Guru Notifications](#)
- [X-ray insights notifications](#)
- [Monitor, operate, and troubleshoot your AWS resources with interactive ChatOps](#)
- [Amazon CloudWatch Integration Guide | PagerDuty](#)
- [Integrate Opsgenie with Amazon CloudWatch](#)

#### Video correlati:

- [Create Composite Alarms in Amazon CloudWatch](#)
- [AWS Chatbot Overview](#)
- [AWS On Air ft. Mutative Commands in AWS Chatbot](#)

#### Esempi correlati:

- [Alarms, incident management, and remediation in the cloud with Amazon CloudWatch](#)
- [Tutorial: Creating an Amazon EventBridge rule that sends notifications to AWS Chatbot](#)
- [One Observability Workshop](#)

## OPS08-BP05 Creare dashboard

Le dashboard rappresentano la visualizzazione incentrata sull'utente dei dati di telemetria dei carichi di lavoro. Sebbene forniscano un'interfaccia visiva fondamentale, non dovrebbero sostituire i

meccanismi di allarme, ma integrarli. Se realizzate con cura, sono in grado di fornire approfondimenti rapidi sullo stato e sulle prestazioni del sistema e possono informare le parti interessate in tempo reale riguardo ai risultati aziendali e all'impatto dei problemi.

Risultato desiderato:

Approfondimenti chiari e fruibili sullo stato del sistema e dell'azienda attraverso rappresentazioni visive.

Anti-pattern comuni:

- Dashboard eccessivamente complicate con troppe metriche.
- Affidarsi a dashboard senza avvisi per il rilevamento delle anomalie.
- Non aggiornare le dashboard man mano che i carichi di lavoro si evolvono.

Vantaggi dell'adozione di questa best practice:

- Visibilità immediata delle metriche e dei KPI critici di sistema.
- Miglioramento della comunicazione e della comprensione con gli stakeholder.
- Approfondimenti rapidi sull'impatto dei problemi operativi.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

### Dashboard incentrate sul business

Le dashboard personalizzate in base ai KPI aziendali coinvolgono una gamma più ampia di parti interessate. Anche se queste persone potrebbero non essere interessate alle metriche di sistema, desiderano comprendere le implicazioni aziendali di questi numeri. Una dashboard incentrata sul business garantisce che tutte le metriche tecniche e operative monitorate e analizzate siano allineate con gli obiettivi aziendali generali. Questo allineamento fornisce chiarezza, garantendo che tutti siano sulla stessa lunghezza d'onda per quanto riguarda ciò che è essenziale e ciò che non lo è. Inoltre, le dashboard che mettono in evidenza i KPI aziendali tendono ad essere più fruibili. Gli stakeholder possono comprendere rapidamente lo stato delle operazioni, le aree che richiedono attenzione e il potenziale impatto sui risultati aziendali.

Con questo in mente, al momento di creare una dashboard, assicurati che ci sia un equilibrio tra metriche tecniche e KPI aziendali. Entrambi sono fondamentali, ma si rivolgono a un pubblico

diverso. Idealmente, dovresti disporre di dashboard che forniscano una visione olistica dello stato e delle prestazioni del sistema, mettendo in evidenza al contempo i principali risultati aziendali e le loro implicazioni.

Le dashboard di Amazon CloudWatch sono home page personalizzabili nella console CloudWatch che puoi utilizzare per monitorare le tue risorse in un'unica visualizzazione, anche quelle distribuite tra Regioni AWS e account diversi.

### Passaggi dell'implementazione

1. Crea una dashboard di base: [crea una nuova dashboard in CloudWatch](#), assegnandole un nome descrittivo.
2. Usa i widget Markdown: prima di utilizzare le metriche, [usa i widget Markdown](#) per aggiungere un contesto testuale nella parte superiore della dashboard. Questo contesto specifica cosa include la dashboard, qual è l'importanza delle metriche rappresentate e può contenere anche link ad altri dashboard e strumenti di risoluzione dei problemi.
3. Crea le variabili della dashboard: [incorpora le variabili della dashboard](#) laddove appropriato per consentire una visualizzazione dinamica e flessibile della dashboard.
4. Crea i widget per le metriche: [aggiungi i widget](#) per visualizzare le varie metriche generate dall'applicazione e personalizza questi widget in modo che rappresentino efficacemente lo stato del sistema e i risultati aziendali.
5. Esegui query con Log Insights: utilizza [CloudWatch Log Insights](#) per ottenere metriche fruibili dai log e visualizzare questi approfondimenti sulla dashboard.
6. Configura gli allarmi: integra gli [allarmi CloudWatch](#) nella tua dashboard per una rapida visualizzazione di tutte le metriche che superano le soglie prestabilite.
7. Usa Contributor Insights: incorpora [CloudWatch Contributor Insights](#) per analizzare i campi ad alta cardinalità e comprendere meglio i principali fattori di contribuzione della risorsa.
8. Progetta widget personalizzati: per esigenze specifiche non soddisfatte dai widget standard, valuta la possibilità di creare [widget personalizzati](#). Questi possono attingere da varie origini dati o rappresentare i dati in modi unici.
9. Usa AWS Health Dashboard: utilizza [AWS Health Dashboard](#) per ottenere approfondimenti sullo stato del tuo account, sugli eventi e sulle prossime modifiche che potrebbero influire sui servizi e sulle risorse. Puoi anche ottenere una visualizzazione centralizzata degli eventi di integrità in AWS Organizations o creare dashboard personalizzate (per maggiori dettagli, consulta Esempi correlati).

10 Itera e perfeziona: man mano che la tua applicazione si evolve, riesamina regolarmente la dashboard per assicurarne la pertinenza.

## Risorse

Best practice correlate:

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS08-BP01 Analisi delle metriche del carico di lavoro](#)
- [OPS08-BP02 Analizza i log relativi ai carichi di lavoro](#)
- [OPS08-BP03 Analisi delle tracce del carico di lavoro](#)
- [OPS08-BP04 Creare avvisi fruibili](#)

Documenti correlati:

- [Building Dashboards for Operational Visibility](#)
- [Using Amazon CloudWatch Dashboards](#)

Video correlati:

- [Create Cross Account & Cross Region CloudWatch Dashboards](#)
- [AWS re:Invent 2021 - Gain enterprise visibility with Cloud AWS operation dashboards\)](#)

Esempi correlati:

- [One Observability Workshop](#)
- [Application Monitoring with Amazon CloudWatch](#)
- [AWS Health Events Intelligence Dashboards and Insights](#)
- [Visualize AWS Health events using Amazon Managed Grafana](#)

## Comprensione dello stato operativo

Definisci, acquisisci e analizza le metriche delle operazioni per ottenere visibilità alle attività dei team operativi, in modo da intraprendere le azioni appropriate.

La tua organizzazione dovrebbe essere in grado di comprendere con facilità lo stato delle tue operazioni. Dovrai definire gli obiettivi aziendali dei team operativi, identificare gli indicatori chiave di performance che li riflettano, utilizzare e sviluppare metriche basate sui risultati delle operazioni per ottenere approfondimenti utili. Sarebbe opportuno utilizzare tali metriche per implementare dashboard e report con punti di vista aziendali e tecnici che aiuteranno i leader e gli stakeholder a prendere decisioni informate.

AWS semplifica la raccolta e l'analisi dei tuoi log relativi alle operazioni, in modo che tu possa generare parametri, conoscere lo stato operativo e ottenere informazioni approfondite dalle operazioni nel tempo.

### Best practice

- [OPS09-BP01 Misura gli obiettivi operativi e i KPI con le metriche](#)
- [OPS09-BP02 Comunicare lo stato e le tendenze per garantire la visibilità delle operazioni](#)
- [OPS09-BP03 Revisione delle metriche operative e assegnazione delle priorità per favorire il miglioramento](#)

## OPS09-BP01 Misura gli obiettivi operativi e i KPI con le metriche

Ottieni obiettivi e KPI dalla tua organizzazione che definiscano il successo delle operazioni e stabilisci metriche che li riflettano. Definisci previsioni da utilizzare come riferimento e rivalutale regolarmente. Sviluppa meccanismi per raccogliere queste metriche dai team per la valutazione.

### Risultato desiderato:

- Gli obiettivi e i KPI per i team operativi dell'organizzazione sono stati pubblicati e condivisi.
- Vengono stabilite metriche che riflettono questi KPI. Gli esempi possono includere:
  - Lunghezza della coda dei ticket o età media del ticket
  - Numero di ticket raggruppati per tipo di problema
  - Tempo impiegato per lavorare ai problemi con o senza una procedura operativa standardizzata (SOP)
  - Tempo impiegato per il ripristino dopo un push di codice non riuscito
  - Volume delle chiamate

### Anti-pattern comuni:

- Le scadenze di implementazione non vengono rispettate perché gli sviluppatori sono costretti a dedicarsi alle attività di risoluzione dei problemi. I team di sviluppo chiedono più personale, ma non possono quantificarne il numero perché il tempo impiegato non può essere misurato.
- È stato installato un desk di livello 1 per gestire le chiamate degli utenti. Nel corso del tempo, sono aumentati i carichi di lavoro ma non il personale assegnato al desk di livello 1. La soddisfazione dei clienti ne risente a causa dell'aumento dei tempi di chiamata e di quelli per arrivare a una soluzione, ma il team manageriale non vede indicatori di questo problema e non intraprende azioni.
- Un carico di lavoro problematico è stato affidato a un team operativo separato per la gestione. A differenza di altri carichi di lavoro, questo non è accompagnato dalla documentazione e dai runbook adeguati. Pertanto, i team dedicano più tempo alla risoluzione dei problemi e alla gestione degli errori. Tuttavia, non esistono metriche che lo documentino, il che rende difficile comprendere le responsabilità.

Vantaggi dell'adozione di questa best practice: Quando il monitoraggio del carico di lavoro mostra lo stato delle nostre applicazioni e servizi, i team operativi dedicati al monitoraggio forniscono ai proprietari informazioni dettagliate sui cambiamenti avvenuti tra i consumatori di tali carichi di lavoro, come le mutate esigenze aziendali. Misura l'efficacia di questi team e valutali rispetto agli obiettivi aziendali creando metriche in grado di riflettere lo stato delle operazioni. Le metriche possono evidenziare problemi relativi al supporto o identificare quando si verificano deviazioni rispetto a un obiettivo di livello di servizio.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Fissa un appuntamento per parlare con i leader aziendali e gli stakeholder per stabilire gli obiettivi generali del servizio. Stabilisci quali devono essere i compiti dei vari team operativi e quali sfide potrebbero affrontare. Utilizza queste informazioni per un'attività di brainstorming sugli indicatori chiave di prestazione (KPI) che potrebbero riflettere questi obiettivi operativi. Questi potrebbero essere la soddisfazione del cliente, il tempo trascorso dall'ideazione della funzionalità alla sua implementazione, il tempo medio di risoluzione dei problemi e altro.

Partendo dai KPI, identifica le metriche e le origini di dati che potrebbero rispecchiare al meglio questi obiettivi. La soddisfazione del cliente può essere una combinazione di diverse metriche, come i tempi di attesa o di risposta durante le chiamate, i punteggi di soddisfazione e i tipi di problemi sollevati. I tempi di implementazione possono essere la somma del tempo necessario per il test e l'implementazione, con l'aggiunta di eventuali correzioni post-implementazione. Le statistiche che

mostrano il tempo dedicato a diversi tipi di problemi (o il numero di tali problemi) possono fornire indicazioni su dove è necessario un impegno mirato.

## Risorse

Documenti correlati:

- [Amazon QuickSight - Using KPIs](#)
- [Amazon CloudWatch - Using Metrics](#)
- [Creazione di pannelli di controllo](#)
- [How to track your cost optimization KPIs with KPI Dashboard](#)

## OPS09-BP02 Comunicare lo stato e le tendenze per garantire la visibilità delle operazioni

Conoscere lo stato delle operazioni e la direzione verso la quale tendono a muoversi è necessario per identificare quando i risultati possono essere a rischio, se è possibile supportare o meno carichi di lavoro aggiuntivi o per verificare gli effetti che le modifiche hanno avuto sui team. Durante gli eventi operativi, disporre di pagine di stato a cui gli utenti e i team operativi possono fare riferimento per ottenere informazioni può ridurre la pressione sui canali di comunicazione e diffondere informazioni in modo proattivo.

Risultato desiderato:

- I responsabili delle operazioni hanno a disposizione informazioni dettagliate per conoscere il volume di chiamate che i loro team stanno gestendo e quali operazioni sono in corso, ad esempio le implementazioni.
- Quando si verificano eventi che possono compromettere le normali operazioni, vengono inviati avvisi agli stakeholder e alle comunità di utenti.
- Quando ricevono un avviso o si verifica un problema, la leadership dell'organizzazione e gli stakeholder possono controllare una pagina di stato e ottenere informazioni relative a un evento operativo, come punti di contatto, informazioni sui ticket e tempi di ripristino stimati.
- I report messi a disposizione della leadership e degli stakeholder contengono statistiche operative come il volume delle chiamate in un periodo di tempo, i punteggi di soddisfazione degli utenti, il numero e l'età di ticket in sospenso.

## Anti-pattern comuni:

- Se un carico di lavoro si interrompe, il servizio diventa non disponibile. Il volume delle chiamate aumenta quando gli utenti chiedono di sapere cosa sta succedendo. Le richieste dei manager di sapere chi sta risolvendo un problema comportano un ulteriore aumento del volume. Vari team operativi duplicano gli sforzi mentre effettuano indagini.
- La volontà di acquisire una nuova capacità porta a riassegnare gli sforzi di alcuni membri del personale verso compiti di tipo tecnico. Non viene fornito alcun backfill e i tempi di risoluzione dei problemi aumentano. Queste informazioni non vengono acquisite e i manager vengono a conoscenza del problema solo dopo diverse settimane o quando viene ricevuto il feedback negativo degli utenti.

Vantaggi dell'adozione di questa best practice: A volte, durante eventi operativi che hanno un impatto sull'azienda, si spreca molto tempo ed energia in query per ottenere informazioni da vari team nel tentativo di comprendere la situazione. Grazie alla creazione di pagine di stato e dashboard ampiamente diffuse, gli stakeholder possono ottenere rapidamente informazioni, ad esempio, se è stato rilevato o meno un problema, chi è a capo delle attività di risoluzione o quando è previsto un ritorno alle normali operazioni. Ciò permette ai membri del team di avere più tempo per affrontare i problemi, perché non devono dilungarsi a comunicare lo stato agli altri.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Crea dashboard che mostrino le metriche fondamentali attuali per i tuoi team operativi e rendile facilmente accessibili ai responsabili operativi e ai manager.

Crea pagine di stato aggiornabili rapidamente per diffondere informazioni relative a un incidente o un evento, come chi ne è responsabile e chi coordina la risposta. Condividi in questa pagina eventuali passaggi o soluzioni alternative che gli utenti dovrebbero prendere in considerazione e divulga ampiamente la posizione della pagina. Incoraggia gli utenti a controllare prima questa pagina quando si trovano di fronte a un problema sconosciuto.

Raccogli e fornisci report che mostrino lo stato di salute delle operazioni nel tempo e distribuiscili a leader e responsabili decisionali per illustrare il lavoro dei team operativi e le loro sfide ed esigenze.

Condividi con i team le metriche e i report che meglio riflettono gli obiettivi e i KPI e come hanno influito nel guidare il cambiamento. Dedica del tempo a queste attività per aumentare l'importanza delle operazioni nei e tra i team.

## Risorse

Documenti correlati:

- [Measure Progress](#)
- [Creazione di pannelli di controllo per visibilità operativa](#)

Soluzioni correlate:

- [Data Operations](#)

## OPS09-BP03 Revisione delle metriche operative e assegnazione delle priorità per favorire il miglioramento

L'assegnazione di tempo e risorse per la revisione dello stato delle operazioni garantisce che servire il settore d'attività rimanga una priorità quotidiana. Effettua regolarmente riunioni con i responsabili operativi e gli stakeholder per rivedere le metriche, riconfermare o modificare traguardi e obiettivi e dare priorità ai miglioramenti.

Risultato desiderato:

- I responsabili operativi e il personale si incontrano regolarmente per esaminare le metriche in un determinato periodo di riferimento. Si comunicano le sfide, si celebrano le vittorie e si condividono le lezioni apprese.
- Gli stakeholder e i leader aziendali vengono regolarmente informati sullo stato delle operazioni e sollecitati a fornire input su obiettivi, KPI e iniziative future. Vengono discusse e contestualizzate le scelte tra erogazione dei servizi, operazioni e manutenzione.

Anti-pattern comuni:

- Viene lanciato un nuovo prodotto, ma i team operativi di livello 1 e 2 non sono adeguatamente formati per fornire supporto oppure non dispongono di personale aggiuntivo. I leader non vedono le metriche che mostrano la diminuzione dei tempi di risoluzione dei ticket e l'aumento del volume degli incidenti. Si agisce settimane dopo, quando i numeri delle sottoscrizioni iniziano a diminuire a causa di utenti scontenti che abbandonano la piattaforma.
- Da molto tempo esiste un processo manuale per eseguire la manutenzione su un carico di lavoro. La volontà di automatizzare, seppur presente, costituiva una priorità bassa data la scarsa

importanza del sistema. Nel corso del tempo, tuttavia, l'importanza del sistema è cresciuta e ora i team operativi sono impegnati per la maggior parte del tempo in questi processi manuali. Non sono previste risorse per fornire una maggiore strumentazione ai team operativi oberati dall'aumento dei carichi di lavoro, con rischi di burnout per il personale. La leadership viene a conoscenza del problema una volta che viene segnalato da un membro del personale che lascia l'azienda per un concorrente.

Vantaggi dell'adozione di questa best practice: In alcune organizzazioni, può diventare difficile dedicare lo stesso tempo e la stessa attenzione alla fornitura di servizi e a nuovi prodotti od offerte. Quando ciò si verifica, il settore d'attività può risentirne a causa del lento deterioramento del livello di servizio atteso. Questo perché le operazioni non cambiano e non si evolvono di pari passo con la crescita del business e possono diventare presto obsolete. Senza una revisione regolare delle informazioni raccolte dai team operativi, il rischio che l'azienda corre potrebbe diventare visibile solo quando è troppo tardi. Dedicare tempo alla revisione delle metriche e delle procedure insieme al personale operativo e alla leadership, permette di mettere in luce il ruolo cruciale svolto dai team operativi nell'identificare i rischi molto prima che raggiungano livelli critici. I team operativi ottengono una visione migliore dei cambiamenti e delle iniziative aziendali imminenti, il che permette di intraprendere azioni proattive. Grazie alla visibilità delle metriche operative, la leadership è consapevole del ruolo che i team operativi svolgono nel garantire la soddisfazione dei clienti, sia interni che esterni, ed è in grado di valutare meglio le scelte in base alle priorità o di garantire che ci sia sufficiente tempo per modificare e fare evolvere operazioni e risorse attraverso nuove iniziative aziendali e di carico di lavoro.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Dedica del tempo alla revisione delle metriche operative con gli stakeholder e i team operativi e alla revisione dei dati dei report. Inserisci questi report nel contesto degli scopi e degli obiettivi dell'organizzazione per stabilire se vengono raggiunti. Identifica le cause di ambiguità quando gli obiettivi non sono chiari o possono esserci conflitti tra ciò che viene chiesto e ciò che viene fornito.

Identifica come il tempo, le persone e gli strumenti possono contribuire agli esiti delle operazioni. Stabilisci quali KPI ne verrebbero influenzati e quali devono essere gli obiettivi di successo. Effettua regolarmente una revisione per assicurarti che i team operativi dispongano di risorse sufficienti per supportare il settore d'attività.

## Risorse

Documenti correlati:

- [Amazon Athena](#)
- [Documentazione di riferimento su parametri e dimensioni di Amazon CloudWatch](#)
- [Amazon QuickSight](#)
- [AWS Glue](#)
- [AWS Glue Data Catalog](#)
- [Collect metrics and logs from Amazon EC2 instances and on-premises servers with the Amazon CloudWatch Agent](#)
- [Utilizzare i parametri Amazon CloudWatch](#)

## Risposta agli eventi

È consigliabile poter prevedere eventi operativi, sia previsti (per esempio vendite promozionali, distribuzioni e test di vulnerabilità) che imprevisi (per esempio aumenti nell'utilizzo della soluzione ed errori nei suoi componenti). Si dovrebbero utilizzare istruzioni e manuali propri per fornire risultati coerenti nella risposta agli avvisi. Gli avvisi definiti dovrebbero essere controllati da un ruolo o da un team responsabile per la risposta e le escalation. Dovrai conoscere l'impatto aziendale dei componenti del tuo sistema e utilizzarlo per concentrare gli sforzi quando necessario. Dopo ogni evento, dovrai effettuare un'analisi della causa principale (RCA) al fine di evitare il ripetersi di errori o l'utilizzo di soluzioni temporanee.

AWS semplifica la tua risposta agli eventi fornendo strumenti che supportano tutti gli aspetti del tuo carico di lavoro e le operazioni in forma di codice. Tali strumenti ti danno modo di scrivere script di risposta agli eventi delle operazioni e avviare la loro esecuzione in risposta ai dati di monitoraggio.

In AWS, è possibile migliorare il tempo di recupero sostituendo i componenti difettosi con versioni sicuramente funzionanti piuttosto che tentarne la riparazione. Puoi eseguire l'analisi sulla risorsa guasta in un secondo momento fuori banda.

Best practice

- [OPS10-BP01 Utilizzo di un processo per la gestione di eventi, incidenti e problemi](#)
- [OPS10-BP02 Definizione di un processo per ogni avviso](#)

- [OPS10-BP03 Definizione della priorità degli eventi operativi in base agli effetti sul business](#)
- [OPS10-BP04 Definizione dei percorsi di escalation](#)
- [OPS10-BP05 Definizione di un piano di comunicazione con i clienti per eventi che incidono sul servizio](#)
- [OPS10-BP06 Comunicazione dello stato tramite pannelli di controllo](#)
- [OPS10-BP07 Automazione delle risposte agli eventi](#)

## OPS10-BP01 Utilizzo di un processo per la gestione di eventi, incidenti e problemi

La capacità di gestire in modo efficiente eventi, incidenti e problemi è fondamentale per mantenere l'integrità e le prestazioni del carico di lavoro. È essenziale riconoscere e comprendere le differenze tra questi elementi per sviluppare una strategia di risposta e risoluzione efficace. Stabilire e seguire un processo ben definito per ogni aspetto facilita la gestione rapida ed efficace da parte del tuo team di qualsiasi sfida operativa che si presenti.

**Risultato desiderato:** La tua organizzazione gestisce efficacemente eventi operativi, incidenti e problemi attraverso processi ben documentati e archiviati a livello centrale. Questi processi vengono costantemente aggiornati per riflettere le modifiche, semplificando la gestione e mantenendo l'affidabilità del servizio e delle prestazioni dei carichi di lavoro elevata.

**Anti-pattern comuni:**

- Rispondi in modo reattivo, anziché proattivo, agli eventi.
- Vengono adottati approcci incoerenti a diversi tipi di eventi o incidenti.
- La tua organizzazione non effettua analisi e non impara dagli incidenti per prevenire eventi futuri.

**Vantaggi dell'adozione di questa best practice:**

- Processi di risposta semplificati e standardizzati.
- Riduzione dell'impatto degli incidenti su servizi e clienti.
- Risoluzione rapida dei problemi.
- Miglioramento continuo dei processi operativi.

**Livello di rischio associato se questa best practice non fosse adottata:** alto

## Guida all'implementazione

L'implementazione di questa best practice prevede la registrazione degli eventi dei carichi di lavoro. Per la gestione di incidenti e problemi, è necessario ricorrere ai processi. I processi sono documentati, condivisi e aggiornati con frequenza. I problemi vengono identificati, classificati in base alla priorità e corretti.

### Comprensione di eventi, incidenti e problemi

- **Eventi:** Un evento può essere l'adempimento di un'azione, un'occorrenza o un cambiamento di stato. Gli eventi possono essere pianificati o non pianificati e possono avere origine all'interno o all'esterno del carico di lavoro.
- **Incidenti:** Gli incidenti sono eventi che richiedono una risposta, come interruzioni non pianificate o il peggioramento della qualità del servizio. Rappresentano interruzioni che richiedono un'attenzione immediata al fine di ripristinare il normale funzionamento del carico di lavoro.
- **Problemi:** I problemi sono le cause alla base di uno o più incidenti. Identificare e risolvere i problemi implica approfondire gli incidenti per prevenire eventi futuri.

### Passaggi dell'implementazione

#### Eventi

##### 1. Monitora gli eventi:

- [Implementa l'osservabilità](#) e [utilizza l'osservabilità del carico di lavoro](#).
- Le azioni di monitoraggio intraprese da un utente, ruolo o servizio AWS vengono registrate come eventi in [AWS CloudTrail](#).
- Rispondi alle modifiche operative delle tue applicazioni in tempo reale con [Amazon EventBridge](#).
- Valuta, monitora e registra continuamente le modifiche alla configurazione delle risorse con [AWS Config](#).

##### 2. Crea processi:

- Sviluppa un processo per valutare quali eventi sono significativi e richiedono di essere monitorati. Ciò comporta l'impostazione di soglie e parametri per le attività normali e anomale.
- Determina i criteri in base ai quali un evento viene segnalato come un incidente. Ad esempio, la gravità dell'evento, l'impatto sugli utenti o la deviazione dal comportamento previsto.
- Rivedi regolarmente i processi di monitoraggio e risposta agli eventi. Ciò include l'analisi degli incidenti passati, l'adeguamento delle soglie e il perfezionamento dei meccanismi di avviso.

## Incidenti

### 1. Rispondi agli incidenti:

- Usa gli approfondimenti degli strumenti di osservabilità per identificare e rispondere rapidamente agli incidenti.
- Implementa [OpsCenter di AWS Systems Manager](#) per aggregare, organizzare e dare priorità agli elementi operativi e agli incidenti.
- Utilizza servizi come [Amazon CloudWatch](#) e [AWS X-Ray](#) per un'analisi e una risoluzione dei problemi più approfondite.
- Considera il servizio [AWS Managed Services \(AMS\)](#) per una migliore gestione degli incidenti, grazie alle sue capacità proattive, preventive e investigative. AMS estende il supporto operativo con servizi come monitoraggio, rilevamento e risposta agli incidenti e gestione della sicurezza.
- Per i clienti del supporto Enterprise è disponibile [Rilevamento e risposta agli incidenti di AWS](#), che fornisce il monitoraggio proattivo continuo e la gestione degli incidenti per i carichi di lavoro di produzione.

### 2. Crea un processo di gestione degli incidenti:

- Definisci un processo strutturato di gestione degli incidenti, che includa ruoli, protocolli di comunicazione e passaggi per la risoluzione chiari.
- Integra la gestione degli incidenti con strumenti come [AWS Chatbot](#) per una risposta e un coordinamento efficienti.
- Classifica gli incidenti in base alla gravità, con [piani di risposta agli incidenti](#) predefiniti per ogni categoria.

### 3. Apprendi e migliora:

- Conduci [analisi post-incidente](#) per comprendere le cause principali e trovare una risoluzione efficace.
- Aggiorna e migliora continuamente i piani di risposta in base alle revisioni e alle pratiche in evoluzione.
- Documenta e condividi le lezioni apprese tra i team per migliorare la resilienza operativa.
- I clienti del supporto Enterprise possono richiedere di seguire il [workshop relativo alla gestione degli incidenti](#) al proprio Technical Account Manager (TAM). Questo workshop guidato consente di verificare il piano di risposta agli incidenti esistente e ti aiuta a individuare eventuali aree da migliorare.

## Problemi

## 1. Identifica i problemi:

- Utilizza i dati degli incidenti passati per identificare modelli ricorrenti che potrebbero indicare la presenza di problemi sistemici più profondi.
- Usa strumenti come [AWS CloudTrail](#) e [Amazon CloudWatch](#) per analizzare le tendenze e scoprire i problemi sottostanti.
- Coinvolgi team interfunzionali, ad esempio i team dediti alle operazioni, allo sviluppo e i reparti aziendali, per ottenere prospettive diverse sulle cause principali.

## 2. Crea un processo di gestione dei problemi:

- Sviluppa un processo strutturato per la gestione dei problemi, concentrandoti su soluzioni a lungo termine piuttosto che su correzioni rapide.
- Incorpora tecniche di analisi delle cause principali (RCA) per indagare e comprendere le cause alla base degli incidenti.
- Aggiorna le policy e le procedure operative e l'infrastruttura in base ai risultati per prevenire il ripetersi degli incidenti.

## 3. Continua a migliorare:

- Promuovi una cultura di apprendimento e miglioramento continui, incoraggiando i team a identificare e affrontare in modo proattivo i problemi potenziali.
- Analizza e rivedi regolarmente i processi e gli strumenti di gestione dei problemi per allinearli agli scenari aziendali e tecnologici in evoluzione.
- Condividi approfondimenti e best practice in tutta l'organizzazione per creare un ambiente operativo più resiliente ed efficiente.

## 4. Integra AWS Support:

- Utilizza risorse di supporto AWS, come [AWS Trusted Advisor](#), per una guida proattiva e suggerimenti in merito all'ottimizzazione.
- I clienti del supporto Enterprise possono accedere a programmi specializzati come [Countdown AWS](#) per ricevere supporto durante gli eventi critici.
- 

Livello di impegno per il piano di implementazione: Medio

## Risorse

### Best practice correlate:

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS04-BP02 Implementazione della telemetria dell'applicazione](#)
- [OPS07-BP03 Utilizzo di runbook per eseguire le procedure](#)
- [OPS07-BP04 Utilizzo dei playbook per analizzare i problemi](#)
- [OPS08-BP01 Analisi delle metriche del carico di lavoro](#)
- [OPS11-BP02 Esecuzione di analisi post-incidente](#)

#### Documenti correlati:

- [AWS Security Incident Response Guide](#)
- [Rilevamento e risposta agli incidenti di AWS](#)
- [AWS Cloud Adoption Framework: Operations Perspective - Incident and problem management](#)
- [Incident Management in the Age of DevOps and SRE \(Gestione degli incidenti nell'era di DevOps e SRE\)](#)
- [PagerDuty - What is Incident Management? \(PagerDuty - Che cos'è la gestione degli incidenti?\)](#)

#### Video correlati:

- [Top incident response tips from AWS](#)
- [AWS re:Invent 2022 - The Amazon Builders' Library: 25 yrs of Amazon operational excellence](#)
- [AWS re:Invent 2022 - AWS Incident Detection and Response \(SUP201\)](#)
- [Introducing Incident Manager from AWS Systems Manager](#)

#### Esempi correlati:

- [AWS Proactive Services – Incident Management Workshop](#)
- [How to Automate Incident Response with PagerDuty and AWS Systems Manager Incident Manager](#)
- [Engage Incident Responders with the On-Call Schedules in AWS Systems Manager Incident Manager](#)
- [Improve the Visibility and Collaboration during Incident Handling in AWS Systems Manager Incident Manager](#)

- [Incident reports and service requests in AMS](#)

Servizi correlati:

- [Amazon EventBridge](#)

## OPS10-BP02 Definizione di un processo per ogni avviso

Stabilire un processo chiaro e definito per ogni avviso nel sistema è essenziale per una gestione efficace ed efficiente degli incidenti. Questa pratica garantisce che ogni avviso porti a una risposta specifica e attuabile, migliorando l'affidabilità e la reattività delle operazioni.

Risultato desiderato: Ogni avviso avvia un piano di risposta specifico e ben definito. Ove possibile, le risposte sono automatizzate e dotate di una chiara titolarità e di un percorso di escalation definito. Gli avvisi sono collegati a una base di conoscenze aggiornata, in modo che qualsiasi operatore sia in grado di rispondere in modo coerente ed efficace. Le risposte sono rapide e uniformi su tutta la linea, migliorando l'efficienza e l'affidabilità operativa.

Anti-pattern comuni:

- Gli avvisi non hanno un processo di risposta predefinito, il che porta a risoluzioni improvvisate e tardive.
- Il sovraccarico di avvisi comporta che gli avvisi importanti vengano trascurati.
- Gli avvisi vengono gestiti in modo incoerente a causa della mancanza di titolarità e responsabilità chiare.

Vantaggi dell'adozione di questa best practice:

- Vengono generati solo avvisi utilizzabili, il che riduce l'affaticamento da avvisi.
- Riduzione del tempo medio di risoluzione (MTTR) per problemi operativi.
- Riduzione del tempo medio di indagine (MTTI), il che aiuta a ridurre l'MTTR.
- Migliore capacità di scalare le risposte operative.
- Maggiore coerenza e affidabilità nella gestione degli eventi operativi.

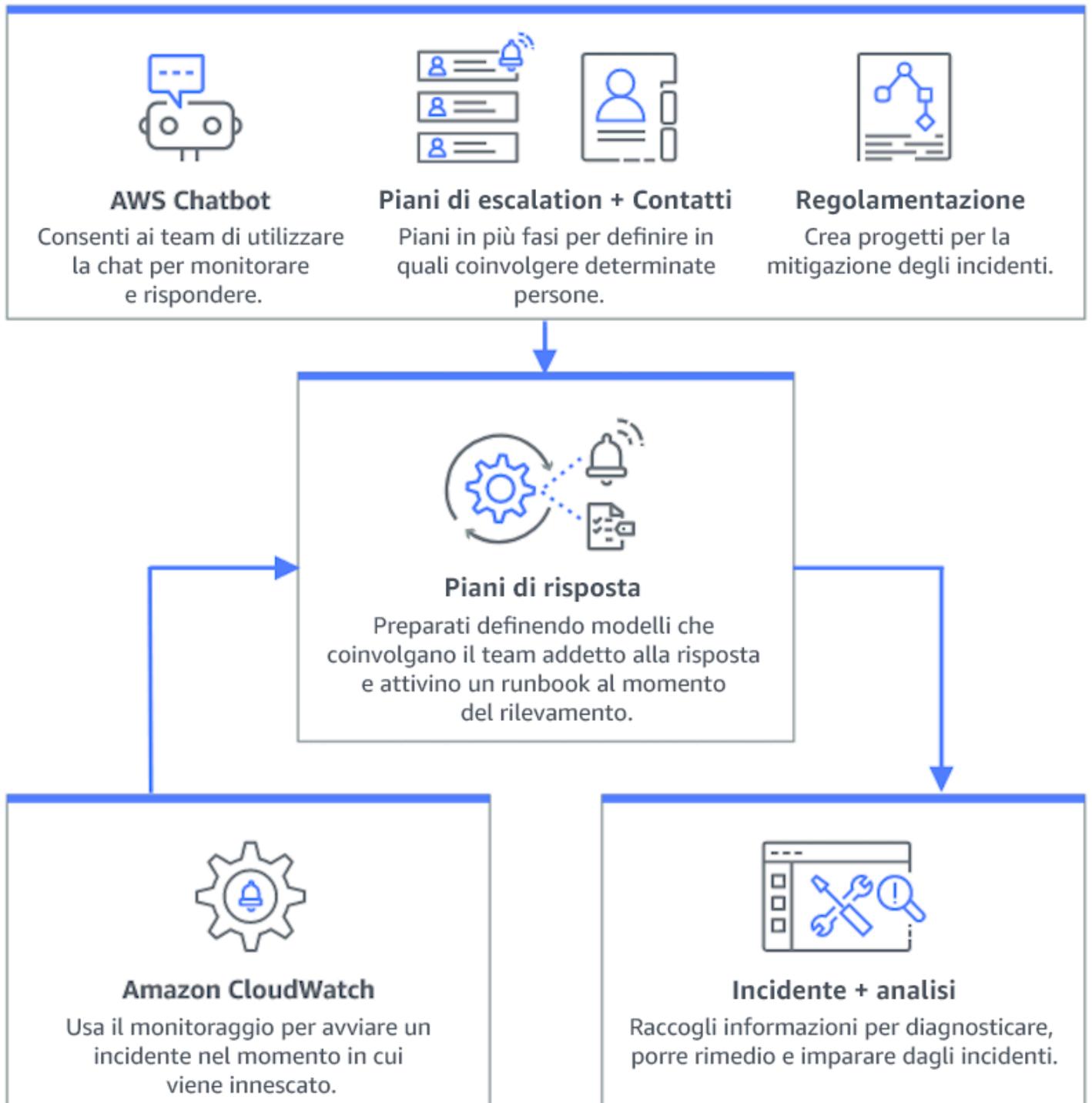
Livello di rischio associato se questa best practice non fosse adottata: alto

## Guida all'implementazione

Avere un processo per ogni avviso implica stabilire un piano di risposta chiaro per ciascun avviso, automatizzare le risposte ove possibile e perfezionare continuamente questi processi in base al feedback operativo e all'evoluzione dei requisiti.

### Passaggi dell'implementazione

Il diagramma seguente illustra il flusso di lavoro di gestione degli incidenti all'interno di [AWS Systems Manager Incident Manager](#). È progettato per rispondere rapidamente ai problemi operativi creando automaticamente incidenti in risposta a eventi specifici che si verificano in [Amazon CloudWatch](#) oppure [Amazon EventBridge](#). Quando viene creato automaticamente o manualmente un incidente, Incident Manager centralizza la gestione dell'incidente, organizza le informazioni pertinenti sulle risorse AWS e avvia piani di risposta predefiniti. Ciò include l'esecuzione dei runbook di Automazione Systems Manager per un'azione immediata e la creazione di un elemento di lavoro operativo principale in OpsCenter per tenere traccia delle attività e delle analisi correlate. Questo processo semplificato accelera e coordina la risposta agli incidenti in tutto l'ambiente AWS.



1. Utilizzo di allarmi compositi: Crea [allarmi compositi](#) in CloudWatch per raggruppare gli allarmi correlati, riducendo il rumore e consentendo risposte più significative.
2. Integra gli allarmi Amazon CloudWatch con Incident Manager Configura gli allarmi CloudWatch per creare automaticamente incidenti in [AWS Systems Manager Incident Manager](#).

3. Integra Amazon EventBridge con Incident Manager: Crea [regole EventBridge](#) per reagire agli eventi e creare incidenti utilizzando piani di risposta definiti.
4. Preparazione per gli incidenti in Incident Manager:
  - Definisci [piani di risposta dettagliati](#) in Incident Manager per ogni tipo di avviso.
  - Stabilisci canali di chat tramite [AWS Chatbot](#) collegato ai piani di risposta in Incident Manager, facilitando la comunicazione in tempo reale durante gli incidenti su piattaforme come Slack, Microsoft Teams e Amazon Chime.
  - Incorpora [i runbook di Automazione Systems Manager](#) all'interno di Incident Manager per guidare le risposte automatiche agli incidenti.

## Risorse

### Best practice correlate:

- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS08-BP04 Creare avvisi fruibili](#)

### Documenti correlati:

- [AWS Cloud Adoption Framework: Operations Perspective - Incident and problem management](#)
- [Utilizzo degli allarmi di Amazon CloudWatch](#)
- [Configurazione di AWS Systems Manager Incident Manager](#)
- [Preparazione per gli incidenti in Incident Manager](#)

### Video correlati:

- [Top incident response tips from AWS](#)

### Esempi correlati:

- [AWS Workshop - AWS Systems Manager Incident Manager - Automate incident response to security events](#)

## OPS10-BP03 Definizione della priorità degli eventi operativi in base agli effetti sul business

Rispondere tempestivamente agli eventi operativi è fondamentale, ma non tutti gli eventi sono uguali. Quando si assegnano le priorità in base all'impatto aziendale, si dà la priorità anche alla risoluzione di eventi che possono avere conseguenze significative, come la compromissione della sicurezza, perdite finanziarie, violazioni normative o danni alla reputazione.

Risultato desiderato: La priorità delle risposte agli eventi operativi si basa sul potenziale impatto dell'evento sulle operazioni e sugli obiettivi aziendali. Ciò rende le risposte efficienti ed efficaci.

Anti-pattern comuni:

- Ogni evento viene trattato con lo stesso livello di urgenza, generando confusione e ritardi nell'affrontare le criticità.
- Non è possibile distinguere tra eventi ad alto e basso impatto, con conseguente errata allocazione delle risorse.
- L'organizzazione non dispone di un chiaro framework di assegnazione delle priorità, il che genera risposte incoerenti agli eventi operativi.
- Agli eventi viene assegnata la priorità in base all'ordine in cui vengono segnalati piuttosto che al loro impatto sui risultati aziendali.

Vantaggi dell'adozione di questa best practice:

- Assicura che la risposta si concentri in primo luogo sulle funzioni aziendali critiche , riducendo al minimo i danni potenziali.
- Migliora l'allocazione delle risorse durante più eventi simultanei.
- Migliora la capacità dell'organizzazione di mantenere la fiducia e soddisfare i requisiti normativi.

Livello di rischio associato se questa best practice non fosse adottata: medio

### Guida all'implementazione

Di fronte a molteplici eventi operativi, è essenziale un approccio strutturato alla definizione delle priorità basato sull'impatto e sull'urgenza. Questo approccio consente di prendere decisioni informate, indirizzare gli sforzi laddove sono più necessari e mitigare il rischio per la continuità aziendale.

## Passaggi dell'implementazione

1. Valuta l'impatto: Sviluppa un sistema di classificazione per valutare la gravità degli eventi in termini di potenziale impatto sulle operazioni e sugli obiettivi aziendali. L'esempio seguente mostra le categorie di impatto:

Livello di impatto	Descrizione
alto	Coinvolge molti dipendenti o clienti, ha un elevato impatto finanziario, genera un elevato danno alla reputazione o lesioni.
medio	Coinvolge un gruppo di dipendenti o clienti, ha un impatto finanziario moderato o genera un danno alla reputazione moderato.
basso	Coinvolge singoli dipendenti o clienti, ha un basso impatto finanziario o genera un danno alla reputazione di lieve entità.

2. Valuta l'urgenza: Definisci i livelli di urgenza in base alla rapidità con cui un evento deve ricevere una risposta, considerando fattori come la sicurezza, le implicazioni finanziarie e gli accordi sui livelli di servizio (SLA). L'esempio seguente illustra le categorie di urgenza:

Livello di urgenza	Descrizione
alto	Produce danni che aumentano in maniera esponenziale, incide su un lavoro sensibile al fattore tempo, escalation imminente, interessa utenti o gruppi VIP.
medio	Produce danni che aumentano nel tempo oppure interessa un singolo utente o gruppo VIP.
basso	Produce danni marginali che aumentano nel tempo o incide su lavori non sensibili al fattore tempo.

### 3. Crea una matrice di prioritizzazione:

- Usa una matrice per incrociare impatto e urgenza, assegnando livelli di priorità a diverse combinazioni.
- Rendi la matrice accessibile e comprensibile da tutti i membri del team responsabili delle risposte agli eventi operativi.
- La seguente matrice di esempio mostra la gravità dell'incidente in base all'urgenza e all'impatto:

Urgenza e impatto	alto	Media	basso
alto	Critica	Urgente	alto
medio	Urgente	alto	Normale
basso	alto	Normale	basso

4. Forma e comunica: Forma i team di risposta sulla matrice di prioritizzazione e sull'importanza di attenersi ad essa durante un evento. Comunica il processo di definizione delle priorità a tutte le parti interessate per stabilire aspettative chiare.
5. Integrazione con la risposta agli incidenti:
  - Incorpora la matrice di prioritizzazione nei tuoi piani e strumenti di risposta agli incidenti.
  - Automatizza la classificazione e la prioritizzazione degli eventi, ove possibile, per accelerare i tempi di risposta.
  - I clienti del supporto Enterprise hanno a disposizione [Rilevamento e risposta agli incidenti di AWS](#), che fornisce il monitoraggio proattivo 24 ore su 24, 7 giorni su 7 e la gestione degli incidenti per i carichi di lavoro di produzione.
6. Rivedi e adatta: Rivedi regolarmente l'efficacia del processo di definizione delle priorità e apporta modifiche in base al feedback e ai cambiamenti nell'ambiente aziendale.

## Risorse

### Best practice correlate:

- [OPS03-BP03 Incoraggiamento all'escalation](#)
- [OPS08-BP04 Creare avvisi fruibili](#)
- [OPS09-BP01 Misura gli obiettivi operativi e i KPI con le metriche](#)

## Documenti correlati:

- [Atlassian - Understanding incident severity levels](#)
- [IT Process Map - Checklist Incident Priority](#)

## OPS10-BP04 Definizione dei percorsi di escalation

Stabilisci percorsi di escalation chiari all'interno dei tuoi protocolli di risposta agli incidenti per facilitare un'azione tempestiva ed efficace. Ciò include la specificazione delle richieste relative all'escalation, la descrizione dettagliata del processo di escalation e la preapprovazione delle azioni per accelerare il processo decisionale e ridurre il tempo medio di risoluzione (MTTR).

Risultato desiderato: Un processo strutturato ed efficiente che inoltra gli incidenti al personale appropriato, riducendo al minimo i tempi di risposta e l'impatto.

### Anti-pattern comuni:

- La mancanza di chiarezza in merito alle procedure di ripristino genera risposte improvvisate in caso di incidenti critici.
- L'assenza di autorizzazioni e titolarità definite comporta ritardi quando è necessaria un'azione urgente.
- Le parti interessate e i clienti non sono informati nei tempi attesi.
- Le decisioni importanti subiscono ritardi.

### Vantaggi dell'adozione di questa best practice:

- Risposta semplificata agli incidenti tramite procedure di escalation predefinite.
- Tempi di inattività ridotti con azioni preapprovate e titolarità chiara.
- Migliore allocazione delle risorse e adeguamenti del livello di supporto in base alla gravità degli incidenti.
- Migliore comunicazione con le parti interessate e i clienti.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

I percorsi di escalation correttamente definiti sono fondamentali per una risposta rapida agli incidenti. AWS Systems Manager Incident Manager supporta l'impostazione di piani di escalation strutturati e di pianificazioni della reperibilità, che avvisano il personale pertinente preparandolo ad agire in caso di incidenti.

### Passaggi dell'implementazione

1. Configura le richieste di escalation: Configura [avvisi CloudWatch](#) per creare un incidente in [AWS Systems Manager Incident Manager](#).
2. Imposta la pianificazione della reperibilità: Crea [la pianificazione della reperibilità](#) in Incident Manager affinché sia in linea con i tuoi percorsi di escalation. Fornisci al personale di turno le autorizzazioni e gli strumenti necessari per agire rapidamente.
3. Procedure di escalation dettagliate:
  - Determina le condizioni specifiche in base alle quali un incidente deve essere inoltrato.
  - Crea [piani di escalation](#) in Incident Manager.
  - I canali di escalation devono consistere in un contatto o in una pianificazione della reperibilità.
  - Definisci i ruoli e le responsabilità del team a ogni livello di escalation.
4. Approva preventivamente le azioni di mitigazione: Collabora con i decision maker per approvare preventivamente le azioni per gli scenari previsti. Utilizza [i runbook di Automazione Systems Manager](#) integrati con Incident Manager per accelerare la risoluzione degli incidenti.
5. Specifica la proprietà: Identifica chiaramente i proprietari interni per ogni fase del percorso di escalation.
6. Fornisci dettagli in merito alle escalation a terze parti:
  - Documenta gli accordi sui livelli di servizio (SLA) di terze parti e adeguati agli obiettivi interni.
  - Stabilisci protocolli chiari per la comunicazione con i fornitori durante gli incidenti.
  - Integra i contatti dei fornitori negli strumenti di gestione degli incidenti per l'accesso diretto.
  - Conduci regolarmente esercitazioni che includano scenari di risposta di terze parti.
  - Mantieni le informazioni sulle escalation dei fornitori ben documentate e facilmente accessibili.
7. Esegui formazione e test per i piani di escalation: Forma il tuo team sul processo di escalation e conduci regolarmente esercitazioni di risposta agli incidenti o game day. I clienti del supporto Enterprise possono richiedere di seguire un [workshop relativo alla gestione degli incidenti](#).

8. Continua a migliorare: Verifica regolarmente l'efficacia dei tuoi percorsi di escalation. Aggiorna i tuoi processi in base alle lezioni apprese dalle analisi degli incidenti e dal feedback continuo.

Livello di impegno per il piano di implementazione: Moderato

## Risorse

Best practice correlate:

- [OPS08-BP04 Creare avvisi fruibili](#)
- [OPS10-BP02 Definizione di un processo per ogni avviso](#)
- [OPS11-BP02 Esecuzione di analisi post-incidente](#)

Documenti correlati:

- [Piani di escalation di AWS Systems Manager Incident Manager](#)
- [Working with on-call schedules in Incident Manager](#)
- [Creating and Managing Runbooks](#)
- [Temporary elevated access management with AWS IAM Identity Center](#)
- [Atlassian - Escalation policies for effective incident management](#)

## OPS10-BP05 Definizione di un piano di comunicazione con i clienti per eventi che incidono sul servizio

Una comunicazione efficace durante gli eventi che incidono sul servizio è fondamentale per mantenere la fiducia e la trasparenza con i clienti. Un piano di comunicazione ben definito sostiene la comunicazione rapida e chiara di informazioni all'interno e all'esterno dell'organizzazione durante gli incidenti.

Risultato desiderato:

- Un solido piano di comunicazione che informa efficacemente i clienti e le parti interessate durante gli eventi che influiscono sul servizio.
- Trasparenza nella comunicazione per creare fiducia e ridurre la preoccupazione dei clienti.
- Riduzione al minimo dell'impatto che gli eventi che incidono sul servizio hanno sull'esperienza del cliente e sulle operazioni aziendali.

## Anti-pattern comuni:

- Una comunicazione inadeguata o in ritardo genera confusione e insoddisfazione nei clienti.
- Una messaggistica eccessivamente tecnica o vaga impedisce la comunicazione dell'impatto effettivo sugli utenti.
- È assente una strategia di comunicazione predefinita, con conseguente messaggistica incoerente e reattiva.

## Vantaggi dell'adozione di questa best practice:

- Maggiore fiducia e soddisfazione dei clienti attraverso una comunicazione chiara e proattiva.
- Riduzione del carico operativo per i team di supporto grazie alla risoluzione preventiva delle preoccupazioni dei clienti.
- Maggiore efficienza di gestione e risoluzione degli incidenti.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

La creazione di un piano di comunicazione completo per gli eventi che incidono sul servizio implica prendere in considerazione molteplici aspetti, dalla scelta dei canali giusti alla creazione del messaggio e del tono. Il piano deve essere adattabile, scalabile e soddisfare diversi scenari di interruzione del servizio.

### Passaggi dell'implementazione

#### 1. Definisci ruoli e responsabilità:

- Assegna a un responsabile degli incidenti gravi la supervisione delle attività di risposta agli incidenti.
- Designa un responsabile delle comunicazioni dedicato al coordinamento di tutte le comunicazioni esterne e interne.
- Includi il responsabile dell'assistenza per fornire una comunicazione coerente attraverso ticket di supporto.

2. Identifica i canali di comunicazione: Seleziona canali come chat aziendale, e-mail, SMS, social media, notifiche in-app e pagine di stato. Questi canali devono essere resilienti e in grado di operare in maniera indipendente durante gli eventi che incidono sul servizio.

### 3. Comunica in modo rapido, chiaro e regolare con i clienti:

- Sviluppa modelli per vari scenari di compromissione del servizio, focalizzandoti sulla semplicità e sui dettagli essenziali. Includi informazioni sul problema relativo al servizio, sui tempi di risoluzione previsti e sull'impatto.
- Usa Amazon Pinpoint per avvisare i clienti tramite notifiche push, notifiche in-app, e-mail, SMS, messaggi vocali e messaggi su canali personalizzati.
- Usa Amazon Simple Notification Service (Amazon SNS) per avvisare gli abbonati in modo programmatico o tramite e-mail, notifiche push su dispositivi mobili e SMS.
- Comunica lo stato tramite dashboard condividendo pubblicamente una dashboard Amazon CloudWatch.
- Incoraggia il coinvolgimento sui social media:
  - Monitora attivamente i social media per comprendere il sentimento dei clienti.
  - Pubblica post su piattaforme di social media per aggiornare il pubblico e coinvolgere la comunità.
  - Prepara modelli per una comunicazione coerente e chiara sui social media.

### 4. Coordina la comunicazione interna: Implementa protocolli interni utilizzando strumenti come AWS Chatbot per migliorare il coordinamento e la comunicazione tra i team. Usa le dashboard CloudWatch per comunicare lo stato.

### 5. Orchestra la comunicazione con strumenti e servizi dedicati:

- Usa AWS Systems Manager Incident Manager con AWS Chatbot per configurare canali di chat dedicati per la comunicazione interna e il coordinamento in tempo reale durante gli incidenti.
- Usa i runbook AWS Systems Manager Incident Manager per automatizzare le notifiche ai clienti durante gli incidenti tramite Amazon Pinpoint, Amazon SNS o strumenti di terze parti come le piattaforme di social media.
- Incorpora i flussi di lavoro di approvazione all'interno dei runbook per rivedere e autorizzare tutte le comunicazioni esterne prima dell'invio.

### 6. Fai pratica e migliora:

- Tieni corsi di formazione sull'uso di strumenti e strategie di comunicazione. Responsabilizza i team affinché siano in grado di prendere decisioni tempestive durante gli incidenti.
- Testa il piano di comunicazione con esercitazioni regolari o game day. Usa questi test per perfezionare la messaggistica e valutare l'efficacia dei canali.

- Implementa meccanismi di feedback per valutare l'efficacia della comunicazione durante gli incidenti. Sviluppa continuamente il piano di comunicazione in base al feedback e alle esigenze mutevoli.

Livello di impegno per il piano di implementazione: alto

## Risorse

Best practice correlate:

- [OPS07-BP03 Utilizzo di runbook per eseguire le procedure](#)
- [OPS10-BP06 Comunicazione dello stato tramite pannelli di controllo](#)
- [OPS11-BP02 Esecuzione di analisi post-incidente](#)

Documenti correlati:

- [Atlassian - Incident communication best practices](#)
- [Atlassian - How to write a good status update](#)
- [PagerDuty - A Guide to Incident Communications](#)

Video correlati:

- [Atlassian - Create your own incident communication plan: Incident templates](#)

Esempi correlati:

- [Dashboard AWS Health](#)
- [Esempi di aggiornamenti di stato AWS](#)

## OPS10-BP06 Comunicazione dello stato tramite pannelli di controllo

Usa le dashboard come strumento strategico per trasmettere lo stato operativo e le metriche fondamentali in tempo reale a diversi tipi di pubblico, inclusi team tecnici interni, leader e clienti. Queste dashboard offrono una rappresentazione visiva centralizzata dello stato del sistema e delle prestazioni aziendali, il che migliora la trasparenza e l'efficienza decisionale.

### Risultato desiderato:

- Le dashboard forniscono una visione completa del sistema e delle metriche aziendali rilevanti per i diversi stakeholder.
- Le parti interessate possono accedere in modo proattivo alle informazioni operative, il che riduce la necessità di richieste di stato frequenti.
- Migliore processo decisionale in tempo reale durante le normali operazioni e gli incidenti.

### Anti-pattern comuni:

- I tecnici che partecipano a una chiamata di gestione degli incidenti hanno bisogno di ricevere aggiornamenti di stato per poter agire rapidamente.
- Affidarsi alla reportistica manuale per la gestione comporta ritardi e potenziali imprecisioni.
- I team operativi vengono spesso interrotti per aggiornamenti sullo stato durante gli incidenti.

### Vantaggi dell'adozione di questa best practice:

- Consente alle parti interessate di accedere immediatamente alle informazioni critiche, promuovendo un processo decisionale informato.
- Riduce le inefficienze operative riducendo al minimo i report manuali e le richieste di stato frequenti.
- Aumenta la trasparenza e la fiducia attraverso la visibilità in tempo reale delle prestazioni del sistema e delle metriche aziendali.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Le dashboard comunicano efficacemente lo stato dei sistemi e le metriche aziendali e possono essere personalizzate in base alle esigenze di diversi gruppi di destinatari. Strumenti come le dashboard Amazon CloudWatch e Amazon QuickSight aiutano a creare dashboard interattive e in tempo reale per il monitoraggio del sistema e la business intelligence.

### Passaggi dell'implementazione

1. Identifica le esigenze degli stakeholder: Determina le esigenze informative specifiche dei diversi gruppi di destinatari, come team tecnici, leader e clienti.

2. Scegli gli strumenti corretti: Seleziona strumenti appropriati come [dashboard Amazon CloudWatch](#) per il monitoraggio del sistema e [Amazon QuickSight](#) per una business intelligence interattiva.
3. Progetta dashboard efficaci:
  - Progetta dashboard per presentare in modo chiaro metriche e KPI pertinenti, assicurandoti che siano comprensibili e utilizzabili.
  - Incorpora visualizzazioni a livello di sistema e a livello aziendale, se necessario.
  - Includi dashboard di alto livello (per ampie panoramiche) e di basso livello (per analisi dettagliate).
  - Integra allarmi automatici all'interno di dashboard per evidenziare i problemi critici.
  - Annota le dashboard con soglie e obiettivi delle metriche importanti per una visibilità immediata.
4. Integra le fonti di dati:
  - Utilizza [Amazon CloudWatch](#) per aggregare e visualizzare le metriche di vari servizi AWS e [metriche di query da altre fonti di dati](#), creando una visione unificata delle metriche aziendali e dello stato del sistema.
  - Usa funzionalità come [CloudWatch Logs Insights](#) per effettuare query e visualizzare i dati di log provenienti da diverse applicazioni e servizi.
5. Fornisci l'accesso self-service:
  - Condividi le dashboard CloudWatch con le parti interessate per l'accesso self-service alle informazioni utilizzando [funzionalità di condivisione della dashboard](#).
  - Assicurati che le dashboard siano facilmente accessibili e contengano informazioni aggiornate in tempo reale.
6. Aggiorna e perfeziona regolarmente:
  - Aggiorna e perfeziona continuamente le dashboard per allinearle alle esigenze aziendali in evoluzione e al feedback degli stakeholder.
  - Rivedi regolarmente le dashboard per assicurarti che siano sempre pertinenti ed efficaci nella trasmissione delle informazioni necessarie.

## Risorse

### Best practice correlate:

- [OPS08-BP05 Creare dashboard](#)

## Documenti correlati:

- [Creazione di dashboard per visibilità operativa](#)
- [Utilizzo dei pannelli di controllo Amazon CloudWatch](#)
- [Crea dashboard flessibili con variabili della dashboard](#)
- [Condivisione di dashboard CloudWatch](#)
- [Metriche di query da altre fonti di dati](#)
- [Aggiungi un widget personalizzato a una dashboard CloudWatch](#)

## Esempi correlati:

- [One Observability Workshop - Dashboards](#)

## OPS10-BP07 Automazione delle risposte agli eventi

L'automazione delle risposte agli eventi è fondamentale per una gestione operativa rapida, coerente e priva di errori. Crea processi semplificati e utilizza strumenti per gestire e rispondere automaticamente agli eventi, riducendo al minimo gli interventi manuali e migliorando l'efficacia operativa.

### Risultato desiderato:

- Riduzione degli errori umani e tempi di risoluzione più rapidi grazie all'automazione.
- Gestione degli eventi operativi coerente e affidabile.
- Maggiore efficienza operativa e affidabilità del sistema.

### Anti-pattern comuni:

- La gestione manuale degli eventi comporta ritardi ed errori.
- L'automazione viene trascurata nelle attività ripetitive e critiche.
- Le attività manuali ripetitive causano affaticamento da avvisi e la mancata identificazione di problemi critici.

### Vantaggi dell'adozione di questa best practice:

- Risposte agli eventi accelerate, riduzione dei tempi di inattività del sistema.

- Operazioni affidabili con gestione automatizzata e coerente degli eventi.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

Incorpora l'automazione per creare flussi di lavoro operativi efficienti e ridurre al minimo gli interventi manuali.

### Passaggi dell'implementazione

1. Identifica le opportunità di automazione: Definisci le attività ripetitive da automatizzare, come la risoluzione dei problemi, l'arricchimento dei ticket, la gestione della capacità, la scalabilità, le implementazioni e i test.
2. Identifica i prompt di automazione:
  - Valuta e definisci condizioni o metriche specifiche al verificarsi delle quali inviare risposte automatiche utilizzando [le azioni di allarme di Amazon CloudWatch](#).
  - Utilizza [Amazon EventBridge](#) per rispondere agli eventi nei servizi AWS, nei carichi di lavoro personalizzati e nelle applicazioni SaaS.
  - Prendi in considerazione eventi scatenanti come [voci di log specifiche](#), [soglie delle metriche delle prestazioni](#) o [cambiamenti di stato](#) nelle risorse AWS.
3. Implementa l'automazione basata sugli eventi:
  - Usa i runbook di Automazione AWS Systems Manager per semplificare le attività di manutenzione, implementazione e bonifica.
  - [Quando vengono creati incidenti in Incident Manager](#), i dettagli relativi alle risorse AWS coinvolte nell'incidente vengono raccolti e aggiunti automaticamente.
  - Monitora in modo proattivo le quote utilizzando [Monitoraggio delle quote per AWS](#).
  - Regola automaticamente la capacità con [AWS Auto Scaling](#) per mantenere la disponibilità e le prestazioni.
  - Automatizza le pipeline di sviluppo con [Amazon CodeCatalyst](#).
  - Smoke test o monitoraggio continuo di endpoint e API [utilizzando il monitoraggio sintetico](#).
4. Esegui la mitigazione del rischio attraverso l'automazione:
  - Implementa [risposte di sicurezza automatizzate](#) per affrontare rapidamente i rischi.
  - Utilizza [AWS Systems Manager State Manager](#) per ridurre la deviazione delle configurazioni.
  - [Correggi le risorse non conformi con Regole di AWS Config](#).

Livello di impegno per il piano di implementazione: alto

## Risorse

Best practice correlate:

- [OPS08-BP04 Creare avvisi fruibili](#)
- [OPS10-BP02 Definizione di un processo per ogni avviso](#)

Documenti correlati:

- [Using Systems Manager Automation runbooks with Incident Manager](#)
- [Creating incidents in Incident Manager](#)
- [AWS Service Quotas](#)
- [Monitor resource usage and send notifications when approaching quotas](#)
- [AWS Auto Scaling](#)
- [What is Amazon CodeCatalyst?](#)
- [Utilizzo degli allarmi di Amazon CloudWatch](#)
- [Utilizzo delle azioni di allarme di Amazon CloudWatch](#)
- [Remediating Noncompliant Resources with Regole di AWS Config](#)
- [Creazione di parametri da registro eventi mediante filtri](#)
- [AWS Systems Manager State Manager](#)

Video correlati:

- [Create Automation Runbooks with AWS Systems Manager](#)
- [How to automate IT Operations on AWS](#)
- [AWS Security Hub automation rules](#)
- [Start your software project fast with Amazon CodeCatalyst blueprints](#)

Esempi correlati:

- [Amazon CodeCatalyst Tutorial: Creating a project with the Modern three-tier web application blueprint](#)

- [One Observability Workshop](#)
- [Respond to incidents using Incident Manager](#)

# Evoluzione

L'evoluzione è il ciclo continuo di miglioramento nel corso del tempo. Implementa frequenti modifiche minime e incrementalmente in base alle lezioni apprese dalle attività operative e valuta il loro successo nell'apportare miglioramenti.

Per garantire l'evoluzione delle operazioni nel corso del tempo, devi essere in grado di:

## Argomenti

- [Imparare, condividere e migliorare](#)

## Imparare, condividere e migliorare

Dedicare del tempo regolarmente all'analisi delle attività delle operazioni, all'analisi degli errori, alla sperimentazione e ad apportare miglioramenti è essenziale. Quando si verificano degli errori, dovrai assicurarti che il tuo team, nonché tutta la tua community di ingegneri, imparino da tali errori. È consigliabile analizzare gli errori per identificare le lezioni apprese e pianificare i miglioramenti. Aggiungerai regolarmente le conoscenze apprese con altri team per confermare i tuoi criteri.

## Best practice

- [OPS11-BP01 Definizione di un processo per il miglioramento continuo](#)
- [OPS11-BP02 Esecuzione di analisi post-incidente](#)
- [OPS11-BP03 Implementazione di cicli di feedback](#)
- [OPS11-BP04 Gestione delle informazioni](#)
- [OPS11-BP05 Definizione dei fattori che promuovono il miglioramento](#)
- [OPS11-BP06 Convalida delle informazioni](#)
- [OPS11-BP07 Revisione dei parametri delle operazioni](#)
- [OPS11-BP08 Documentazione e condivisione delle conoscenze acquisite](#)
- [OPS11-BP09 Allocazione di tempo per i miglioramenti](#)

## OPS11-BP01 Definizione di un processo per il miglioramento continuo

Valuta il carico di lavoro rispetto alle best practice dell'architettura interna ed esterna. Effettua revisioni frequenti e deliberate del carico di lavoro. Dai priorità alle opportunità di miglioramento nella cadenza di sviluppo del software.

Risultato desiderato:

- Analizza di frequente il carico di lavoro rispetto alle best practice dell'architettura.
- Stabilisci per le opportunità di miglioramento la stessa priorità che assegni alle funzionalità del processo di sviluppo software.

Anti-pattern comuni:

- Non hai condotto una revisione dell'architettura del carico di lavoro da quando è stato implementato diversi anni fa.
- Stabilisci una priorità inferiore per le opportunità di miglioramento. Rispetto alle nuove funzionalità, queste opportunità rimangono nel backlog.
- Non esiste uno standard per l'implementazione delle modifiche alle best practice per l'organizzazione.

Vantaggi dell'adozione di questa best practice:

- Il carico di lavoro è aggiornato sulla base delle best practice di architettura.
- Fai evolvere il carico di lavoro in modo intenzionale.
- Puoi utilizzare le best practice dell'organizzazione per migliorare tutti i carichi di lavoro.
- Ottieni guadagni marginali che hanno un impatto cumulativo, con un incremento dell'efficienza.

Livello di rischio associato se questa best practice non fosse adottata: elevato

### Guida all'implementazione

Effettui di frequente la revisione dell'architettura del carico di lavoro. Utilizzi le best practice interne ed esterne per valutare il carico di lavoro e identificare le opportunità di miglioramento. Dai priorità alle opportunità di miglioramento nella cadenza di sviluppo del software.

## Passaggi dell'implementazione

1. Esegui la revisione periodica dell'architettura del carico di lavoro di produzione secondo una frequenza concordata. Utilizza uno standard architettonico documentato che includa best practice specifiche di AWS.
  - a. Usa gli standard definiti internamente per queste revisioni. Se non hai standard interni, usa Framework AWS Well-Architected.
  - b. Utilizza AWS Well-Architected Tool per creare un obiettivo personalizzato delle best practice interne e condurre la revisione dell'architettura.
  - c. Contatta un AWS Solution Architect o Technical Account Manager per condurre una revisione guidata di Framework Well-Architected del carico di lavoro.
2. Dai priorità alle opportunità di miglioramento identificate durante la revisione nel processo di sviluppo del software.

Livello di impegno per il piano di implementazione: Basso. Si può usare AWS Well-Architected Framework per eseguire la revisione annuale dell'architettura.

## Risorse

Best practice correlate:

- [OPS11-BP02 Esecuzione di analisi post-incidente](#)
- [OPS11-BP08 Documentazione e condivisione delle conoscenze acquisite](#)
- [OPS04 Implementazione dell'osservabilità](#)

Documenti correlati:

- [AWS Well-Architected Tool - Custom lenses](#)
- [AWS Well-Architected Whitepaper - The review process](#)
- [Customize Well-Architected Reviews using Custom Lenses and the AWS Well-Architected Tool](#)
- [Implementing the AWS Well-Architected Custom Lens lifecycle in your organization](#)

Video correlati:

- [Well-Architected Labs - Level 100: Custom Lenses on AWS Well-Architected Tool](#)

- [AWS re:Invent 2023 - Scaling AWS Well-Architected best practices across your organization](#)

Esempi correlati:

- [AWS Well-Architected Tool](#)

## OPS11-BP02 Esecuzione di analisi post-incidente

Esamina gli eventi che influiscono sui clienti e identifica i fattori che contribuiscono e le azioni preventive. Utilizza queste informazioni per sviluppare modi per limitare o prevenire il ripetersi degli incidenti. Sviluppa procedure per attivare risposte rapide ed efficaci. Comunica i fattori che hanno contribuito al presentarsi dell'imprevisto e le azioni correttive secondo necessità, specificamente mirate per il pubblico di destinazione.

Risultato desiderato:

- Stabilisci processi di gestione degli incidenti che includono l'analisi post-incidente.
- Hai a disposizione piani di osservabilità per raccogliere dati sugli eventi.
- Con questi dati comprendi e raccogli metriche che supportano il tuo processo di analisi post-incidente.
- Impari dagli incidenti per migliorare i risultati futuri.

Anti-pattern comuni:

- Sei amministratore di un server di applicazioni. Circa ogni 23 ore e 55 minuti tutte le sessioni attive vengono terminate. Hai tentato di identificare ciò che non va a buon fine sul server di applicazioni. Sospetti che potrebbe trattarsi di un problema di rete, ma non riesci a ottenere la collaborazione dal team di rete perché i suoi membri sono troppo occupati per supportarti. Ti manca un processo predefinito da seguire per ottenere supporto e raccogliere le informazioni necessarie per stabilire che cosa sta accadendo.
- Si è verificata una perdita di dati all'interno del carico di lavoro. Questa è la prima volta che si è verificata e la causa non è immediatamente identificabile. Decidi che non è importante perché puoi ricreare i dati. La perdita di dati inizia a verificarsi con maggiore frequenza e influisce sui clienti. Questo comporta inoltre un ulteriore onere operativo quando ripristini i dati mancanti.

Vantaggi dell'adozione di questa best practice:

- Disponendo di un processo predefinito per determinare i componenti, le condizioni, le azioni e gli eventi che hanno contribuito a un incidente, sei in grado di identificare le opportunità di miglioramento.
- Utilizzi i dati dell'analisi post-incidente per apportare miglioramenti.

Livello di rischio associato se questa best practice non fosse adottata: elevato

## Guida all'implementazione

Utilizza un processo per determinare i fattori che concorrono. Esamina tutti gli incidenti che influiscono sul cliente. Predisponi un processo per identificare e documentare i fattori che contribuiscono a un incidente, in modo da sviluppare azioni di mitigazione in grado di limitare o impedire il suo ripetersi e per sviluppare procedure che consentano risposte rapide ed efficaci. Comunica le cause principali degli incidenti in modo appropriato e personalizza la comunicazione in base al pubblico di destinazione. Condividi quanto appreso in maniera aperta all'interno della tua organizzazione.

### Passaggi dell'implementazione

1. Raccogli metriche come le modifiche all'implementazione e alla configurazione, l'ora di inizio dell'incidente, l'ora dell'allarme, dell'intervento, dell'inizio della mitigazione e il tempo di risoluzione dell'incidente.
2. Descrivi i momenti fondamentali sulla linea temporale per comprendere gli eventi dell'incidente.
3. Poni le seguenti domande:
  - a. Potresti migliorare il tempo di rilevamento?
  - b. Sono presenti aggiornamenti alle metriche e agli allarmi che permettono di rilevare l'incidente prima?
  - c. Puoi migliorare i tempi di diagnosi?
  - d. Sono presenti aggiornamenti ai tuoi piani di risposta o di escalation che potrebbero coinvolgere prima i team di risposta corretti?
  - e. Puoi migliorare il tempo necessario per la mitigazione?
  - f. Ci sono passaggi del runbook o del playbook che potresti aggiungere o migliorare?
  - g. È possibile prevenire che si verifichino incidenti futuri?
4. Crea liste di controllo e azioni. Monitora ed esegui tutte le azioni.

Livello di impegno per il piano di implementazione: medio

## Risorse

Best practice correlate:

- [OPS11-BP01 Definizione di un processo per il miglioramento continuo](#)
- [OPS 4 - Implementazione dell'osservabilità](#)

Documenti correlati:

- [Performing a post-incident analysis in Incident Manager](#)
- [Operational Readiness Review](#)

## OPS11-BP03 Implementazione di cicli di feedback

I cicli di feedback forniscono informazioni fruibili che guidano il processo decisionale. Vanno creati nelle procedure e nei carichi di lavoro per identificare i problemi e le aree che necessitano di miglioramenti. Inoltre, convalidano gli investimenti effettuati nei miglioramenti. Questi cicli di feedback sono la base per migliorare continuamente il carico di lavoro.

I cicli di feedback si dividono in due categorie: feedback immediato e analisi retrospettiva. Il feedback immediato viene raccolto con la revisione delle prestazioni e dei risultati delle attività operative. Questo feedback proviene dai membri del team, dai clienti o dall'output automatizzato dell'attività. Il feedback immediato viene ricevuto ad esempio dal test A/B e dall'offerta di nuove funzionalità, ed è essenziale per anticipare l'errore (fail fast).

L'analisi retrospettiva viene eseguita regolarmente per acquisire il feedback della revisione dei risultati operativi e dei parametri nel tempo. Queste retrospettive si svolgono alla fine di uno sprint, in base a una cadenza o dopo importanti rilasci o eventi. Questo tipo di ciclo di feedback convalida gli investimenti nelle operazioni o nel carico di lavoro, consente di misurare il successo e comprova la tua strategia.

Risultato desiderato: l'uso del feedback immediato e dell'analisi retrospettiva per guidare i miglioramenti. L'applicazione di un meccanismo per acquisire il feedback di utenti e membri del team. L'uso dell'analisi retrospettiva per identificare le tendenze che guidano i miglioramenti.

Anti-pattern comuni:

- Lanci una nuova funzionalità ma non hai modo di ricevere il feedback dei clienti.
- Dopo aver investito in miglioramenti delle operazioni, non conduci una retrospettiva per convalidare gli investimenti.
- Raccogli il feedback dei clienti ma non lo esamini regolarmente.
- I cicli di feedback portano alla proposta di elementi di azione non sono inclusi nel processo di sviluppo software.
- I clienti non ricevono un feedback sui miglioramenti che hanno proposto.

Vantaggi dell'adozione di questa best practice:

- Puoi lavorare a ritroso con il cliente per promuovere nuove funzionalità.
- La cultura della tua organizzazione può reagire più rapidamente ai cambiamenti.
- Le tendenze vengono utilizzate per identificare le opportunità di miglioramento.
- Le retrospettive convalidano gli investimenti effettuati per il carico di lavoro e le operazioni.

Livello di rischio associato se questa best practice non fosse adottata: alto

## Guida all'implementazione

L'implementazione di questa best practice comporta l'utilizzo del feedback immediato e dell'analisi retrospettiva. Questi cicli di feedback guidano i miglioramenti. Esistono molti meccanismi per il feedback immediato, inclusi questionari, sondaggi dei clienti o moduli di feedback. La tua organizzazione utilizza anche le retrospettive per identificare le opportunità di miglioramento e convalidare le iniziative.

### Esempio del cliente

AnyCompany Retail crea un modulo Web in cui i clienti possono fornire il feedback o segnalare problemi. Durante lo Scrum settimanale, il feedback degli utenti viene valutato dal team di sviluppo software. Il feedback viene regolarmente utilizzato per guidare l'evoluzione della piattaforma. Viene eseguita una retrospettiva alla fine di ogni sprint per identificare gli elementi che devono essere migliorati.

## Passaggi dell'implementazione

### 1. Feedback immediato

- Hai bisogno di un meccanismo per ricevere il feedback dai clienti e dai membri del team. Le attività operative possono anche essere configurate per fornire un feedback automatizzato.
- L'organizzazione ha bisogno di un processo per rivedere il feedback, determinare cosa migliorare e pianificare il miglioramento.
- Il feedback deve essere aggiunto al processo di sviluppo software.
- Quando apporti miglioramenti, contatta l'autore del feedback.
  - Puoi utilizzare [AWS Systems Manager OpsCenter](#) per creare e monitorare questi miglioramenti come [OpsItems](#).

## 2. Analisi retrospettiva

- Conduci le retrospettive alla fine di un ciclo di sviluppo, a una cadenza prestabilita o dopo un rilascio importante.
- Riunisci gli stakeholder coinvolti nel carico di lavoro per la riunione retrospettiva.
- Crea tre colonne sulla lavagna o in un foglio di lavoro: Fine, Inizio e Mantenimento.
  - Fine è per tutto ciò che vuoi che il team smetta di fare.
  - Inizio è per le idee che vuoi iniziare ad applicare.
  - Mantenimento è per ciò che vuoi continuare a fare.
- Raccogli il feedback dagli stakeholder.
- Dai priorità al feedback. Assegna le azioni e gli stakeholder a qualsiasi elemento nelle colonne Inizio e Mantenimento.
- Aggiungi le azioni al processo di sviluppo software e comunica gli aggiornamenti sullo stato agli stakeholder mentre apporti i miglioramenti.

Livello di impegno per il piano di implementazione: medio. Per implementare questa best practice è necessario un modo per ricevere il feedback immediato e analizzarlo. Inoltre, è necessario stabilire un processo di analisi retrospettiva.

## Risorse

Best practice correlate:

- [OPS01-BP01 Valutazione delle esigenze dei clienti](#): i cicli di feedback sono un meccanismo per raccogliere le esigenze dei clienti esterni.
- [OPS01-BP02 Valutazione delle esigenze dei clienti interni](#): gli stakeholder interni possono utilizzare i cicli di feedback per comunicare necessità e requisiti.

- [OPS11-BP02 Esecuzione di analisi post-incidente](#): le analisi successive agli incidenti sono una forma importante di analisi retrospettiva da condurre dopo gli incidenti.
- [OPS11-BP07 Revisione dei parametri delle operazioni](#): le revisioni dei parametri operativi identificano tendenze e aree di miglioramento.

#### Documenti correlati:

- [7 Pitfalls to Avoid When Building CCOE \(7 errori da evitare durante la creazione di un Centro di eccellenza del Cloud \(CCoE\)\)](#)
- [Atlassian Team Playbook - Retrospectives \(Playbook Atlassian Team - Retrospective\)](#)
- [Email Definitions: Feedback Loops \(Definizioni di e-mail: cicli di feedback\)](#)
- [Establishing Feedback Loops Based on the AWS Well-Architected Framework Review \(Applicazione dei cicli di feedback in base alla revisione di Framework AWS Well-Architected\)](#)
- [IBM Garage Methodology - Hold a retrospective \(Metodologia IBM Garage - Condurre una retrospettiva\)](#)
- [Investopedia - The PDCA Cycle \(Investopedia - Il ciclo PDCA\)](#)
- [Maximizing Developer Effectiveness by Tim Cochran \(Massimizzazione dell'efficacia degli sviluppatori di Tim Cochran\)](#)
- [Operations Readiness Reviews \(ORR\) Whitepaper - Iteration \(Whitepaper per le revisioni della preparazione delle operazioni - Iterazione\)](#)
- [TIL CSI - Continual Service Improvement \(TIL CSI - Miglioramento continuo del servizio\)](#)
- [When Toyota met e-commerce: Lean at Amazon \(Toyota incontra l'e-commerce: semplificazione con Amazon\)](#)

#### Video correlati:

- [Building Effective Customer Feedback Loops \(Creazione di efficaci cicli di feedback dei clienti\)](#)

#### Esempi correlati:

- [Astuto - Open source customer feedback tool \(Astuto - Strumento di feedback dei clienti open source\)](#)
- [AWS Solutions - QnABot on AWS \(Soluzioni AWS - QnABot in AWS\)](#)

- [Fider - A platform to organize customer feedback \(Fider - Una piattaforma per organizzare il feedback dei clienti\)](#)

Servizi correlati:

- [AWS Systems Manager OpsCenter](#)

## OPS11-BP04 Gestione delle informazioni

La gestione delle informazioni permette ai membri del team di trovare le informazioni necessarie per svolgere il proprio lavoro. Nelle organizzazioni che promuovono la formazione dei propri dipendenti, le informazioni vengono liberamente condivise, migliorando le competenze personali. Le informazioni possono essere vagliate o cercate. Le informazioni sono accurate e aggiornate. Esistono meccanismi per creare nuove informazioni, aggiornare quelle esistenti e archiviare quelle obsolete. L'esempio più comune di una piattaforma di gestione delle informazioni è un sistema di gestione dei contenuti come un Wiki.

Risultato desiderato:

- Accesso per i membri del team a informazioni tempestive e accurate.
- Possibilità di eseguire ricerche nelle informazioni.
- Presenza di un meccanismo per aggiungere, aggiornare e archiviare le informazioni.

Anti-pattern comuni:

- Assenza di un sistema di archiviazione centrale delle informazioni. I membri del team gestiscono i propri appunti su computer locali.
- Presenza di un Wiki self-hosted, ma senza alcun meccanismo per la gestione delle informazioni, con informazioni non aggiornate di conseguenza.
- Le informazioni mancanti vengono identificate da qualcuno, ma non esiste un processo per richiederne l'aggiunta nel Wiki del team. I dipendenti le aggiungono manualmente ma omettono un passaggio importante, causando un'interruzione.

Vantaggi dell'adozione di questa best practice:

- I membri del team acquisiscono le competenze necessarie perché le informazioni vengono condivise liberamente.
- Nuovi membri del team vengono integrati più facilmente perché la documentazione è aggiornata e può essere oggetto di ricerche.
- Le informazioni sono tempestive, accurate e di utilità pratica.

Livello di rischio associato alla mancata adozione di questa best practice: elevato

## Guida all'implementazione

La gestione delle informazioni è un aspetto importante delle aziende che promuovono la formazione dei propri dipendenti. Per iniziare, è necessario un repository centrale in cui archiviare le informazioni, un esempio comune del quale è un Wiki self-hosted. Devi sviluppare processi per l'aggiunta, l'aggiornamento e l'archiviazione delle informazioni. Sviluppa standard per gli aspetti da documentare e permetti a ciascuno di contribuire.

### Esempio del cliente

AnyCompany Retail ospita un Wiki interno in cui vengono archiviate tutte le informazioni. I membri del team sono incoraggiati ad aggiungere il proprio input nella knowledge base durante lo svolgimento delle proprie mansioni quotidiane. Ogni trimestre un team interfunzionale valuta le pagine obsolete e determina se devono essere archiviate o aggiornate.

### Passaggi dell'implementazione

1. Per iniziare, identifica il sistema di gestione dei contenuti in cui verranno archiviate le informazioni. Ottieni il consenso degli stakeholder in tutta l'organizzazione.
  - a. Se non possiedi un sistema di gestione dei contenuti, valuta se eseguire un Wiki self-hosted o usare un repository con controllo delle versioni come punto di partenza.
2. Sviluppa runbook per l'aggiunta, l'aggiornamento e l'archiviazione delle informazioni. Fornisci ai team la formazione necessaria su questi processi.
3. Identifica le informazioni che devono essere archiviate nel sistema di gestione dei contenuti. Inizia dalle attività quotidiane (runbook e playbook) svolte dai membri del team. Collabora con gli stakeholder per classificare in ordine di priorità le informazioni aggiunte.
4. Collabora periodicamente con gli stakeholder per identificare le informazioni obsolete e archivarle o aggiornarle.

Livello di impegno per il piano di implementazione: medio. Se non possiedi un sistema di gestione dei contenuti, puoi configurare un Wiki self-hosted o un repository di documenti con controllo delle versioni.

## Risorse

Best practice correlate:

- [OPS11-BP08 Documentazione e condivisione delle conoscenze acquisite](#) – La gestione delle informazioni semplifica la condivisione delle conclusioni sulle lezioni apprese.

Documenti correlati:

- [Atlassian - Knowledge Management](#)

Esempi correlati:

- [DokuWiki](#)
- [Gollum](#)
- [MediaWiki](#)
- [Wiki.js](#)

## OPS11-BP05 Definizione dei fattori che promuovono il miglioramento

Identifica i fattori che promuovono il miglioramento in modo da valutare e dare priorità alle opportunità sulla base di dati e cicli di feedback. Esplora le opportunità di miglioramento nei sistemi e nei processi e automatizza laddove appropriato.

Risultato desiderato:

- Tieni traccia dei dati provenienti da tutto l'ambiente.
- Esegui la correlazione di eventi e attività ai risultati aziendali.
- Puoi confrontare e contrapporre ambienti e sistemi.
- Mantieni una cronologia dettagliata delle attività relative alle implementazioni e ai risultati.
- Raccogli i dati a supporto del livello di sicurezza.

## Anti-pattern comuni:

- Raccogli dati da tutto l'ambiente, ma non correli eventi e attività.
- Raccogli dati dettagliati da tutta la proprietà, aumentando l'attività e i costi di Amazon CloudWatch e AWS CloudTrail, tuttavia non utilizzi questi dati in modo significativo.
- Non tieni conto dei risultati aziendali quando definisci i fattori che promuovono il miglioramento.
- Non misuri gli effetti delle nuove funzionalità.

## Vantaggi dell'adozione di questa best practice:

- Determinando i criteri di miglioramento, riduci al minimo l'impatto delle motivazioni basate sugli eventi o degli investimenti influenzati da fattori emotivi.
- Rispondi agli eventi aziendali, non solo a quelli tecnici.
- Misuri l'ambiente per identificare le aree di miglioramento.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

- Comprensione dei fattori che promuovono il miglioramento: è consigliabile apportare modifiche a un sistema solo quando un risultato desiderato è supportato.
  - Funzionalità desiderate: prendi in considerazione le funzionalità e le capacità desiderate quando valuti le opportunità di miglioramento.
    - [Novità di AWS](#)
  - Problemi inaccettabili: tieni in considerazione i problemi, i bug e le vulnerabilità inaccettabili quando valuti le opportunità di miglioramento. Tieni traccia delle giuste opzioni di dimensionamento e individua le opportunità di ottimizzazione.
    - [Ultimi bollettini sulla sicurezza AWS](#)
    - [AWS Trusted Advisor](#)
    - [Cloud Intelligence Dashboards](#)
  - Requisiti di conformità: quando esamini le opportunità di miglioramento, prendi in considerazione gli aggiornamenti e le modifiche necessarie per mantenere la conformità a normative e policy o per avere diritto al supporto di terze parti.
    - [Conformità di AWS](#)

- [Programmi per la conformità di AWS](#)
- [Ultime novità sulla conformità di AWS](#)

## Risorse

Best practice correlate:

- [OPS01 Priorità dell'organizzazione](#)
- [OPS02 Relazioni e proprietà](#)
- [OPS04-BP01 Identificazione degli indicatori chiave di prestazione](#)
- [OPS08 Utilizzare l'osservabilità del carico di lavoro](#)
- [OPS09 Comprensione dello stato operativo](#)
- [OPS11-BP03 Implementazione di cicli di feedback](#)

Documenti correlati:

- [Amazon Athena](#)
- [Amazon QuickSight](#)
- [Conformità di AWS](#)
- [Ultime novità sulla conformità di AWS](#)
- [Programmi per la conformità di AWS](#)
- [AWS Glue](#)
- [Ultimi bollettini sulla sicurezza AWS](#)
- [AWS Trusted Advisor](#)
- [Esporta i dati di log in Amazon S3](#)
- [Novità di AWS](#)
- [Gli aspetti imprescindibili dell'innovazione orientata al cliente](#)
- [Digital Transformation: Hype or a Strategic Necessity?](#)

Video correlati

- [AWS re:Invent 2023 - Improve operational efficiency and resilience with AWS Support \(SUP310\)](#)

## OPS11-BP06 Convalida delle informazioni

Rivedi i risultati dell'analisi e le risposte con i team trasversali e i proprietari dell'azienda. Utilizza queste revisioni per definire una visione comune, identificare ulteriori impatti e stabilire le linee d'azione. Adatta le risposte, se necessario.

Risultati desiderati:

- Rivedi regolarmente gli approfondimenti con i proprietari dell'azienda. I proprietari dell'azienda forniscono un contesto aggiuntivo agli approfondimenti appena acquisiti.
- Esamini gli approfondimenti e richiedi il feedback ai colleghi tecnici, quindi condividi le tue conoscenze con i team.
- Pubblichiamo i dati e gli approfondimenti affinché altri team tecnici e aziendali possano esaminarli. Tieni conto dei tuoi apprendimenti nelle nuove procedure di altri reparti.
- Riassumi ed esami i nuovi approfondimenti con i leader senior. I leader senior utilizzano i nuovi approfondimenti per definire la strategia.

Anti-pattern comuni:

- Rilasci una nuova funzionalità che modifica alcuni comportamenti dei clienti. La tua osservabilità non tiene conto di queste modifiche. Non quantifichi i vantaggi di queste modifiche.
- Effettui un nuovo aggiornamento e trascuri l'aggiornamento della rete di distribuzione di contenuti (CDN). La cache della CDN non è più compatibile con l'ultima versione. Misuri la percentuale di richieste con errori. Tutti gli utenti segnalano errori HTTP 400 durante le comunicazioni con i server di backend. Analizzi gli errori del cliente e scopri che, poiché hai misurato la dimensione sbagliata, il tuo tempo è stato improduttivo.
- L'accordo sul livello di servizio prevede un tempo di attività del 99,9% e l'obiettivo del punto di ripristino è di quattro ore. Il proprietario del servizio sostiene che il sistema non subisce tempi di inattività. Implementi una soluzione di replica costosa e complessa, che comporta uno spreco di tempo e denaro.

Vantaggi dell'adozione di questa best practice:

- Convalidando gli approfondimenti con i proprietari dell'azienda e con gli esperti in materia, è possibile stabilire una comprensione comune e gestire il miglioramento in modo più efficace.
- Individui i problemi nascosti e ne tieni conto nelle decisioni future.

- La tua attenzione si sposta dai risultati tecnici ai risultati aziendali.

Livello di rischio associato se questa best practice non fosse adottata: medio

## Guida all'implementazione

- Convalida degli approfondimenti: interagisci con i proprietari dell'azienda e gli esperti in materia per garantire la comprensione e l'accordo comuni sul significato dei dati raccolti. Individua ulteriori problemi e impatti potenziali e stabilisci le azioni da intraprendere.

## Risorse

Best practice correlate:

- [OPS01-BP06 Valutazione dei compromessi gestendo vantaggi e rischi](#)
- [OPS02-BP06 Predefinizione o negoziazione delle responsabilità tra i team](#)
- [OPS11-BP03 Implementazione di cicli di feedback](#)

Documenti correlati:

- [Designing a Cloud Center of Excellence \(CCOE\)](#)

Video correlati:

- [Building observability to increase resiliency](#)

## OPS11-BP07 Revisione dei parametri delle operazioni

Esegui regolarmente un'analisi retrospettiva dei parametri operativi con i partecipanti di vari team da diverse aree del business. Utilizza queste revisioni per identificare opportunità di miglioramento e potenziali linee d'azione e per condividere le conoscenze acquisite. Cerca opportunità di miglioramento in tutti i tuoi ambienti, ad esempio sviluppo, test e produzione.

Risultato desiderato:

- Esamini di frequente le metriche che hanno un impatto sull'azienda.
- Rilevi ed esami le anomalie con le tue capacità di osservabilità.

- Utilizzi i dati per supportare i risultati e gli obiettivi aziendali.

#### Anti-pattern comuni:

- La finestra di manutenzione interrompe un'importante promozione al dettaglio. L'azienda non è al corrente del fatto che i normali interventi di manutenzione possono essere rimandati nel caso vi siano altri eventi di particolare rilievo per l'azienda.
- Per l'uso comune di una libreria obsoleta nella tua organizzazione si è verificata una prolungata interruzione del servizio. Successivamente hai eseguito la migrazione a una libreria supportata. Gli altri team della tua organizzazione non sanno di essere a rischio.
- Non verifichi regolarmente l'aderenza agli SLA dei clienti. Le tendenze indicano un andamento negativo per quanto riguarda il rispetto degli SLA. In caso di mancato rispetto degli SLA, sono previste sanzioni economiche.

#### Vantaggi dell'adozione di questa best practice:

- Durante le riunioni che organizzi regolarmente per esaminare le metriche operative, gli eventi e gli incidenti, stabilisci una comprensione comune tra i team.
- Il tuo team si riunisce regolarmente per esaminare metriche e incidenti, il che ti consente di intervenire sui rischi e rispettare gli SLA dei clienti.
- Condividi le lezioni apprese, che forniscono dati per la definizione delle priorità e miglioramenti mirati per ottenere i risultati aziendali.

Livello di rischio associato se questa best practice non fosse adottata: medio

#### Guida all'implementazione

- Esegui regolarmente un'analisi retrospettiva dei parametri operativi con i partecipanti di vari team da diverse aree del business.
- Coinvolgi i soggetti interessati, compresi i team che si occupano di business, sviluppo e operazioni, per convalidare ciò che è emerso dal feedback immediato e dall'analisi retrospettiva e per condividere le conoscenze acquisite.
- Utilizza gli approfondimenti di cui dispongono per identificare opportunità di miglioramento e possibili linee d'azione.

## Risorse

Best practice correlate:

- [OPS08-BP05 Creare dashboard](#)
- [OPS09-BP03 Revisione delle metriche operative e assegnazione delle priorità per favorire il miglioramento](#)
- [OPS10-BP01 Utilizzo di un processo per la gestione di eventi, incidenti e problemi](#)

Documenti correlati:

- [Amazon CloudWatch](#)
- [Amazon CloudWatch metrics and dimensions reference](#)
- [Publish custom metrics](#)
- [Using Amazon CloudWatch metrics](#)
- [Dashboards and visualizations with CloudWatch](#)

## OPS11-BP08 Documentazione e condivisione delle conoscenze acquisite

Documenta e condividi le conoscenze acquisite durante le attività operative per metterle a frutto internamente e nei vari team. La condivisione di quanto appreso dai team comporta maggiori vantaggi all'interno dell'organizzazione. Condividi informazioni e risorse per impedire che si verifichino errori evitabili e semplificare le attività di sviluppo e concentrati sulla distribuzione delle funzionalità desiderate.

Utilizza AWS Identity and Access Management (IAM) per definire le autorizzazioni che consentono un accesso controllato alle risorse che desideri condividere tra i vari account.

Risultato desiderato:

- Utilizzi repository dotati di controllo versione per condividere librerie dell'applicazione, procedure di scripting, documentazione di procedure e altra documentazione di sistema.
- Condividi gli standard dell'infrastruttura come modelli AWS CloudFormation con controllo delle versioni.
- Riesamini le lezioni apprese con i team.

## Anti-pattern comuni:

- Per l'uso comune di una libreria contenente degli errori nella tua organizzazione si è verificata una prolungata interruzione del servizio. Successivamente hai eseguito la migrazione a una libreria affidabile. Gli altri team della tua organizzazione non sanno di essere a rischio. Nessuno documenta e condivide l'esperienza relativa a questa libreria e nessuno è consapevole del rischio.
- Hai identificato un caso limite in un microservizio condiviso internamente che causa l'interruzione delle sessioni. Hai aggiornato le chiamate al servizio per evitare questo caso limite. Gli altri team della tua organizzazione non sanno di essere a rischio.
- Hai trovato un modo per ridurre in modo significativo i requisiti di utilizzo della CPU per uno dei tuoi microservizi. Non sai se altri team potrebbero sfruttare questa tecnica.

Vantaggi dell'adozione di questa best practice: condividi le lezioni apprese a supporto del miglioramento e per trarre il massimo vantaggio dall'esperienza.

Livello di rischio associato se questa best practice non fosse adottata: basso

## Guida all'implementazione

- Documenta e condividi le conoscenze acquisite: predisponi le procedure per documentare le conoscenze acquisite dall'esecuzione delle attività operative e dalle analisi retrospettive affinché tali informazioni possano essere utilizzate dai altri team.
- Condividi le conoscenze acquisite: predisponi le procedure per condividere con tutti i team le conoscenze acquisite e gli artefatti associati. Ad esempio condividi le procedure, le istruzioni, la governance e le best practice aggiornate tramite un wiki accessibile. Condividi script, codice e librerie tramite un repository comune.
  - [Delegating access to your AWS environment](#)
  - [Share an AWS CodeCommit repository](#)

## Risorse

Best practice correlate:

- [OPS02-BP06 Predefinizione o negoziazione delle responsabilità tra i team](#)
- [OPS05-BP01 Utilizzo del controllo delle versioni](#)
- [OPS05-BP06 Condivisione degli standard di progettazione](#)

- [OPS11-BP03 Implementazione di cicli di feedback](#)
- [OPS11-BP07 Revisione dei parametri delle operazioni](#)

Documenti correlati:

- [Reduce project delays with a docs-as-code solution](#)

Video correlati:

- [Delegating access to your AWS environment](#)
- [AWS Supports You | Exploring the Incident Management Tabletop Exercise](#)

## OPS11-BP09 Allocazione di tempo per i miglioramenti

Dedica tempo e risorse nei processi per rendere possibile il miglioramento incrementale continuo.

Risultato desiderato:

- Crei duplicati temporanei paralleli di ambienti per ridurre il rischio, lo sforzo e il costo della sperimentazione e dell'esecuzione di test.
- Questi ambienti duplicati possono essere utilizzati per testare le conclusioni di analisi ed esperimenti, ma anche per sviluppare e testare i miglioramenti pianificati.
- Organizzi giornate di gioco e utilizzi il servizio del Simulatore di iniezione guasti (FIS) per fornire ai team i controlli e i guardrail necessari per eseguire esperimenti in un ambiente simile a quello di produzione.

Anti-pattern comuni:

- Si è verificato un problema di prestazioni noto nel server di applicazioni. Il problema viene aggiunto al backlog, dopo l'implementazione prevista delle varie funzionalità. Se la velocità con cui vengono aggiunte le funzionalità pianificate rimane costante, il problema di prestazioni non verrà mai risolto.
- Per supportare il miglioramento continuo, autorizzi amministratori e sviluppatori a utilizzare tutto il loro tempo aggiuntivo per definire e implementare miglioramenti. I miglioramenti non vengono mai completati.
- L'accettazione operativa è stata completata e non si testano più le procedure operative.

Vantaggi dell'adozione di questa best practice: dedicando tempo e risorse nei processi, è possibile apportare miglioramenti continui e incrementali.

Livello di rischio associato se questa best practice non fosse adottata: basso

## Guida all'implementazione

- Allocazione di tempo per apportare miglioramenti: dedica tempo e risorse all'interno dei processi per rendere possibili miglioramenti graduali e continui.
- Implementa modifiche per migliorare e valutare i risultati per favorire il successo.
- Se i risultati non sono in linea con gli obiettivi e il miglioramento resta prioritario, valuta procedure d'azione alternative.
- Simula i carichi di lavoro di produzione durante le giornate di gioco e utilizza le conoscenze conseguite da queste simulazioni per migliorare.

## Risorse

Best practice correlate:

- [OPS05-BP08 Utilizzo di più ambienti](#)

Video correlati:

- [AWS re:Invent 2023 - Improve application resilience with AWS Fault Injection Service](#)

# Conclusione

L'eccellenza operativa richiede un impegno costante.

Configura la tua organizzazione per il successo fissando degli obiettivi condivisi. Assicurati che tutti comprendano il proprio ruolo nel raggiungere i risultati aziendali e la propria influenza sull'efficienza degli altri. Fornisci supporto ai membri del team in modo che possano supportare i risultati aziendali.

Ciascun evento operativo ed errore dovrebbe essere considerato un'opportunità per migliorare le operazioni della tua architettura. Comprendendo le necessità dei tuoi carichi di lavoro, predefinendo runbook per le attività di routine e playbook per indirizzare la risoluzione dei problemi, utilizzando le operazioni come funzionalità del codice in AWS e mantenendo infine la consapevolezza situazionale, le operazioni saranno più preparate e in grado di rispondere in modo più efficace quando si verificano incidenti.

Concentrandoti sul miglioramento incrementale basato sulle priorità in continua evoluzione e sulle conoscenze acquisite dall'analisi retrospettiva e della risposta agli eventi, sosterrai il successo della tua azienda grazie all'incremento dell'efficienza e dell'efficacia delle tue operazioni.

AWS si impegna per aiutarti a costruire e gestire architetture che massimizzino l'efficienza mentre crei distribuzioni altamente reattive e adattive. Per migliorare l'eccellenza operativa dei carichi di lavoro, è consigliabile utilizzare le best practice discusse in questo documento.

# Collaboratori

- Rich Boyd, Operational Excellence Pillar Lead, Well-Architected, Amazon Web Services
- Jon Steele, Solutions Architect Well-Architected, Amazon Web Services
- Ryan King, Sr. Technical Program Manager, Amazon Web Services
- Chris Kunselman, Advisory Consultant, Amazon Web Services
- Peter Mullen, Advisory Consultant, Amazon Web Services
- Brian Quinn, Sr. Advisory Consultant, Amazon Web Services
- David Stanley, Cloud Operating Model Lead, Amazon Web Services
- Chris Kozlowski, Senior Specialist Technical Account Manager, Enterprise Support, Amazon Web Services
- Alex Livingstone, Principal Specialist Solutions Architect, Cloud Operations, Amazon Web Services
- Paul Moran, Principal Technologist, Enterprise Support, Amazon Web Services
- Peter Mullen, Advisory Consultant, Professional Services, Amazon Web Services
- Chris Pates, Senior Specialist Technical Account Manager, Enterprise Support, Amazon Web Services
- Arvind Raghunathan, Principal Specialist Technical Account Manager, Enterprise Support, Amazon Web Services
- Ben Mergen, Senior Cost Lead Solutions Architect, Amazon Web Services

# Approfondimenti

Per ulteriori informazioni, consulta le seguenti risorse:

- [Framework AWS Well-Architected](#)
- [Centro di progettazione AWS](#)

# Revisioni del documento

Per ricevere una notifica sugli aggiornamenti di questo whitepaper, iscriviti al feed RSS.

Modifica	Descrizione	Data
<a href="#">Whitepaper aggiornato</a>	Best practice aggiornate con nuova guida all'implementazione.	June 27, 2024
<a href="#">Aggiornamento e consolidamento principale dei contenuti</a>	<p>I contenuti sono stati aggiornati e consolidati in diverse aree di best practice. Due aree di best practice (OPS 04 e OPS 08) sono state riscritte e presentano nuovi contenuti e focus.</p> <p>Le best practice sono state aggiornate e consolidate nelle seguenti aree: <a href="#">Progettazione delle operazioni</a>, <a href="#">Riduzione dei rischi di distribuzione</a> e <a href="#">Comprensione dello stato operativo</a>. L'area di best practice OPS 04 è stata aggiornata in <a href="#">Implementa l'osservabilità</a>. L'area di best practice OPS 08 è stata aggiornata in <a href="#">Utilizzare l'osservabilità del carico di lavoro</a>.</p>	October 3, 2023
<a href="#">Aggiornamenti per il nuovo canone</a>	Best practice aggiornate con prontuario e nuove best practice aggiunte.	April 10, 2023

---

<a href="#">Whitepaper aggiornato</a>	Best practice aggiornate con nuova guida all'implementazione.	December 15, 2022
<a href="#">Whitepaper aggiornato</a>	Ampliamento delle best practice e aggiunta dei piani di miglioramento.	October 20, 2022
<a href="#">Aggiornamento di minore entità</a>	Piccolo aggiornamento editoriale.	August 8, 2022
<a href="#">Whitepaper aggiornato</a>	Aggiornamenti che rispecchiano i nuovi servizi e le nuove funzionalità di AWS e le best practice aggiornate.	February 2, 2022
<a href="#">Aggiornamento di minore entità</a>	Aggiunta del pilastro della sostenibilità all'introduzione.	December 2, 2021
<a href="#">Aggiornamenti per il nuovo canone</a>	Aggiornamenti che rispecchiano i nuovi servizi e le nuove funzionalità di AWS e le best practice aggiornate.	July 8, 2020
<a href="#">Whitepaper aggiornato</a>	Aggiornamenti che rispecchiano i nuovi servizi e le nuove funzionalità di AWS; riferimenti aggiornati.	July 1, 2018
<a href="#">Pubblicazione originale</a>	Pubblicazione del Pilastro dell'eccellenza operativa – Framework AWS Well-Architected.	November 1, 2017