



La cartina di CI/CD tornasole: la vostra pipeline è completamente CI/CD?

AWS Guida prescrittiva



AWS Guida prescrittiva: La cartina di CI/CD tornasole: la vostra pipeline è completamente CI/CD?

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

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

Table of Contents

Introduzione	1
Obiettivi	1
Comprendere CI/CD	3
Informazioni sull'integrazione continua	4
Informazioni sulla consegna continua	4
Tests	5
Parametri	6
Differenze nei processi CI/CD	8
Approccio Gitflow	8
Approccio basato su Trunk	10
Integrità dell'ambiente	11
Rilasci	12
Sicurezza	13
Cartina tornasole per CI/CD condotte	14
Best practice	16
Domande frequenti	18
Quali sono alcuni indicatori chiave del fatto che il mio processo di implementazione non è completamente CI/CD?	18
Cosa succede se desidero utilizzare un CI/CD processo completo ma comunque pianificare il rilascio di determinate funzionalità per momenti specifici?	18
Cosa succede se alcune fasi del mio processo di distribuzione non possono essere automatizzate?	18
E se il mio staff tecnico fosse più a suo agio con i flussi di lavoro esistenti piuttosto che con un CI/CD processo completo?	19
Cosa succede se i miei ambienti sono suddivisi in più account? Posso ancora utilizzare un CI/CD processo completo?	19
Fasi successive	20
Risorse	21
AWS documentazione e riferimenti	21
Servizi e strumenti	21
Cronologia dei documenti	22
Glossario	23
#	23
A	24

B	27
C	29
D	32
E	36
F	38
G	40
H	41
I	43
L	45
M	46
O	51
P	53
Q	56
R	57
S	60
T	64
U	65
V	66
W	66
Z	68
.....	lxix

La cartina di CI/CD tornasole: la tua pipeline è completamente CI/CD?

Steven Guggenheimer e Ananya Koduri, Amazon Web Services (AWS)

Agosto 2023 ([cronologia dei documenti](#))

La tua pipeline è automatizzata? È una domanda semplice, ma molte organizzazioni affrontano la risposta in modo troppo semplice. La risposta è molto più complicata di un sì o di un no.

Le innovazioni tecnologiche avvengono costantemente e a volte può essere difficile per le organizzazioni tenere il passo. Questa novità è una moda passeggera o è la prossima grande novità? Devo rivedere le mie pratiche attuali o devo aspettare? Spesso, quando diventa chiaro che qualcosa è davvero la prossima grande novità, ci si ritrova a giocare a recuperare il ritardo. L'integrazione continua e la distribuzione continua (CI/CD) sono destinate a rimanere, ma non è sempre stato così. Molte persone hanno impiegato molto tempo per essere convinte, e alcune hanno ancora bisogno di essere convinte di più.

CI/CD is the process of automating the source, build, test, staging, and production stages of the software release process, and it is commonly described as a pipeline. Today, the cost savings and speed of CI/CD automation hanno convinto la maggior parte delle organizzazioni del suo valore. Ma passare a questo nuovo approccio non è un compito facile. È necessario assicurarsi che il personale riceva la formazione giusta, è necessario potenziare alcune risorse e poi è necessario testare, testare, testare. C'è molto da fare. Nella maggior parte dei casi, si desidera apportare queste modifiche gradualmente per aiutare l'organizzazione ad adattarsi.

Lo scopo di questo documento è definire cosa significa avere un CI/CD processo completo. Fornisce uno strumento per valutare i propri processi e presenta un percorso da seguire per processi che non sono ancora disponibili. Questo percorso verso il futuro è raramente una conversione dall'oggi al domani. Questi processi sono complessi e dipendono da molti fattori, tra cui le attuali competenze dei dipendenti e le attuali esigenze di infrastruttura. Ti consigliamo di stabilire le priorità e apportare piccole modifiche incrementali.

Obiettivi

Di seguito sono riportati i potenziali vantaggi derivanti dall'implementazione delle raccomandazioni contenute in questa guida:

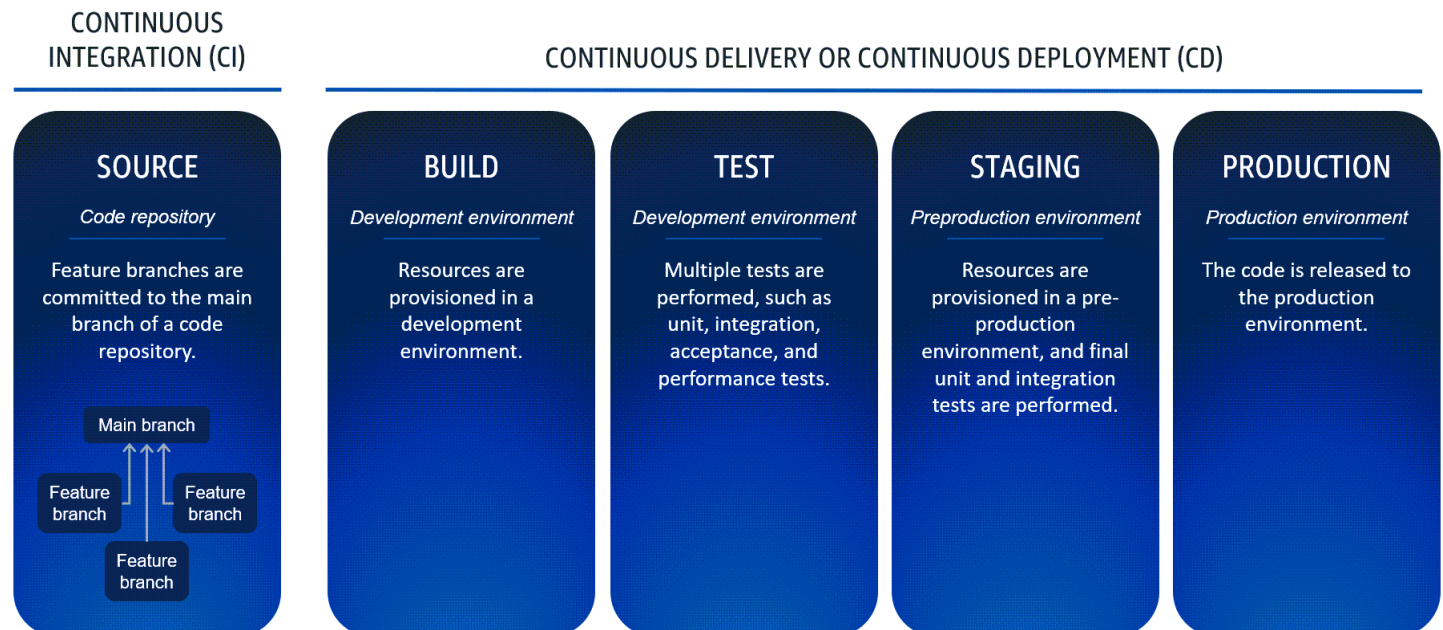
- **Efficienza:** un processo di CI/CD implementazione completo può ridurre la complessità, i carichi di lavoro e le innumerevoli ore dedicate al debug, all'esecuzione di processi manuali e alla manutenzione. Per ulteriori informazioni, consulta [Vantaggi della distribuzione continua](#). Secondo un [post TechAhead sul blog](#), l'implementazione del CI/CD processo può portare a un risparmio stimato del 20% in termini di tempo, impegno e risorse.
- **Riduzione dei costi** — Secondo un [rapporto di Forbes Insight](#), «Tre dirigenti su quattro concordano sul fatto che la quantità di tempo, denaro e risorse spesi per la manutenzione e la gestione continue, rispetto allo sviluppo di nuovi progetti o nuove iniziative, influisca sulla competitività complessiva della loro organizzazione». Più breve è il ciclo di sviluppo, maggiori sono le possibilità che l'organizzazione possa raggiungere obiettivi ambiziosi time-to-market e cogliere le opportunità giuste al momento giusto.
- **Velocità:** in genere, una CI/CD pipeline is able to release software changes to customers within a few hours. Especially in cases with quick fault isolations and small patch pushes, the CI/CD pipeline completa aiuta a migliorare il tempo medio di ripristino (MTTR). [Per ulteriori informazioni, vedere Riduzione dell'MTTR.](#)
- **Sicurezza:** CI/CD le pipeline complete proteggono anche il processo di rilascio riducendo i possibili punti di ingresso per gli attacchi e riducendo il rischio di errore umano. I vantaggi in termini di sicurezza derivanti da CI/CD pipeline completamente automatizzate aiutano a evitare le costose conseguenze di violazioni dei dati, interruzioni del servizio e altro ancora.
- **Riduzione del tasso di attrito:** gli sviluppatori sono più soddisfatti quando possono dedicare più tempo alla creazione di ottime funzionalità e meno tempo a dedicare meno tempo a un ciclo infinito di manutenzione e debug. Per le organizzazioni, ciò significa acquisire e trattenere i migliori talenti per periodi di tempo più lunghi.
- **Codice di qualità superiore:** gli sviluppatori rilasciano il codice in un repository condiviso in piccoli batch, il che consente loro di condurre [test paralleli](#) (post BrowserStack sul blog). Piuttosto che lavorare in modo isolato, condividono spesso le loro build con il team e collaborano per identificare i bug critici. Ciò fornisce supporto agli sviluppatori, il che aiuta a impedire che codice errato venga messo in produzione. Il supporto dei colleghi sviluppatori contribuisce a rilasci di alta qualità e favorisce la crescita organizzativa.
- **Manutenzione:** la manutenzione e gli aggiornamenti sono fondamentali per realizzare un ottimo prodotto. Tuttavia, non spegnete il sistema durante le ore di punta del traffico. È possibile utilizzare una CI/CD pipeline per eseguire la manutenzione durante le ore di utilizzo ridotto, in modo da ridurre al minimo i tempi di inattività e gli impatti sulle prestazioni.

Comprendere CI/CD

L'integrazione e la distribuzione continue (CI/CD) sono il processo di automazione del ciclo di vita delle release del software. In alcuni casi, il D in CI/CD può anche significare implementazione. La differenza tra distribuzione continua e distribuzione continua si verifica quando si rilascia una modifica all'ambiente di produzione. Con la distribuzione continua, è necessaria un'approvazione manuale prima di promuovere modifiche alla produzione. L'implementazione continua prevede un flusso ininterrotto attraverso l'intera pipeline e non sono richieste approvazioni esplicite. Poiché questa strategia illustra CI/CD concetti generali, i consigli e le informazioni fornite sono applicabili sia agli approcci di distribuzione continua che a quelli di distribuzione continua.

CI/CD automates much or all of the manual processes traditionally required to get new code from a commit into production. A CI/CD pipeline encompasses the source, build, test, staging, and production stages. In each stage, the CI/CD pipelines provisions any infrastructure that is needed to deploy or test the code. By using a CI/CD pipeline, i team di sviluppo possono apportare modifiche al codice che vengono poi testate automaticamente e inviate alla distribuzione.

Esaminiamo il CI/CD processo di base prima di esaminare alcuni dei modi in cui è possibile, consapevolmente o inconsapevolmente, discostarsi dall'essere fasi e attività complete CI/CD. The following diagram shows the CI/CD in ogni fase.



Informazioni sull'integrazione continua

L'integrazione continua avviene in un repository di codice, ad esempio un repository Git in. GitHub Consideri un singolo ramo principale come la fonte di verità per la base di codice e crei rami di breve durata per lo sviluppo di funzionalità. Si integra un feature branch nel ramo principale quando si è pronti a implementare la funzionalità negli ambienti superiori. I feature branch non vengono mai distribuiti direttamente negli ambienti superiori. Per ulteriori informazioni sul tagging, consulta [Approccio basato su Trunk](#) in questa guida.

Processo di integrazione continuo

1. Lo sviluppatore crea un nuovo ramo dal ramo principale.
2. Lo sviluppatore apporta modifiche, compilazioni e test a livello locale.
3. Quando le modifiche sono pronte, lo sviluppatore crea una [pull request](#) (GitHub documentazione) con il ramo principale come destinazione.
4. Il codice viene esaminato.
5. Quando il codice viene approvato, viene unito al ramo principale.

Informazioni sulla consegna continua

La distribuzione continua avviene in ambienti isolati, come ambienti di sviluppo e ambienti di produzione. Le azioni che si verificano in ogni ambiente possono variare. Spesso, una delle prime fasi viene utilizzata per aggiornare la pipeline stessa prima di procedere. Il risultato finale della distribuzione è che ogni ambiente viene aggiornato con le ultime modifiche. Anche il numero di ambienti di sviluppo per la creazione e il test varia, ma è consigliabile utilizzarne almeno due. Nella pipeline, ogni ambiente viene aggiornato in base alla sua importanza, per finire con l'ambiente più importante, l'ambiente di produzione.

Processo di consegna continuo

La parte di distribuzione continua della pipeline inizia estraendo il codice dal ramo principale del repository di origine e passandolo alla fase di compilazione. Il documento Infrastructure as Code (IaC) per il repository delinea le attività eseguite in ogni fase. Sebbene l'utilizzo di un documento IaC non sia obbligatorio, si consiglia vivamente di utilizzare un servizio o uno strumento IaC, come [AWS CloudFormation](#) o [AWS Cloud Development Kit \(AWS CDK\)](#). I passaggi più comuni includono:

1. Test unitari

2. Compilazione del codice
3. Fornitura di risorse
4. Test di integrazione

Se si verificano errori o i test falliscono in qualsiasi fase della pipeline, la fase corrente torna allo stato precedente e la pipeline viene terminata. Le modifiche successive devono iniziare nell'archivio del codice e completare l'intero processo. CI/CD

Test per le condutture CI/CD

I due tipi di test automatici a cui si fa comunemente riferimento nelle pipeline di distribuzione sono i test unitari e i test di integrazione. Tuttavia, esistono molti tipi di test che è possibile eseguire su una base di codice e nell'ambiente di sviluppo. L'[architettura di riferimento di AWS Deployment Pipeline](#) definisce i seguenti tipi di test:

- Test unitario: questi test creano ed eseguono il codice dell'applicazione per verificare che funzioni in base alle aspettative. Simulano tutte le dipendenze esterne utilizzate nella codebase. [Esempi di strumenti di unit test includono JUnitJest e pytest.](#)
- Test di integrazione: questi test verificano che l'applicazione soddisfi i requisiti tecnici eseguendo test in un ambiente di test predisposto. [Esempi di strumenti di test di integrazione includono Cucumber, VRest NG e integ-tests \(for\).](#) AWS CDK
- Test di accettazione: questi test verificano che l'applicazione soddisfi i requisiti degli utenti eseguendo test in un ambiente di test predisposto. [Esempi di strumenti di test di accettazione includono Cypress e Selenium.](#)
- Test sintetici: questi test vengono eseguiti continuamente in background per generare traffico e verificare che il sistema sia integro. Esempi di strumenti di test sintetici includono [Amazon CloudWatch Synthetics e Dynatrace Synthetic Monitoring.](#)
- Test delle prestazioni: questi test simulano la capacità di produzione. Determinano se l'applicazione soddisfa i requisiti prestazionali e confrontano le metriche con le prestazioni precedenti. [Esempi di strumenti di test delle prestazioni includono Apache JMeter, Locust e Gatling.](#)
- Test di resilienza: noti anche come test del caos, questi test iniettano i guasti negli ambienti per identificare le aree a rischio. I periodi in cui i guasti vengono iniettati vengono quindi confrontati con i periodi senza guasti. [Esempi di strumenti di test di resilienza includono e Gremlin. AWS Fault Injection Service](#)

- Test statico di sicurezza delle applicazioni (SAST): questi test analizzano il codice per rilevare eventuali violazioni della sicurezza, come [SQL injection](#) o [cross-site scripting \(XSS\)](#). Esempi di strumenti SAST includono [Amazon CodeGuru](#) e [SonarQubeCheckmarx](#).
- Test dinamico di sicurezza delle applicazioni (DAST): questi test sono noti anche come test di penetrazione o pen test. Identificano vulnerabilità, come SQL injection o XSS in un ambiente di test predisposto. [Esempi di strumenti DAST includono Zed Attack Proxy \(ZAP\) e HCL. AppScan Per ulteriori informazioni, vedere Penetration Testing.](#)

Non tutte le CI/CD pipeline eseguono completamente tutti questi test. Tuttavia, come minimo, una pipeline dovrebbe eseguire test unitari e test SAST sulla base di codice, nonché test di integrazione e accettazione in un ambiente di test.

Metriche per le pipeline CI/CD

Secondo l'[architettura di riferimento AWS di Deployment Pipeline](#), è necessario tenere traccia almeno delle seguenti quattro metriche per le pipeline: CI/CD

- Lead time: il tempo medio impiegato da un singolo commit per arrivare fino alla produzione. Ti consigliamo di impostare un lead time compreso tra 1 ora e 1 giorno, a seconda del caso d'uso.
- Frequenza di implementazione: il numero di implementazioni di produzione in un determinato periodo di tempo. Consigliamo di impostare come target le frequenze di implementazione da più volte al giorno a due volte alla settimana, a seconda del caso d'uso.
- Tempo medio tra i guasti (MTBF): il periodo di tempo medio tra l'avvio di una pipeline riuscita e l'inizio di una pipeline fallita. Consigliamo di puntare a un MTBF il più alto possibile. [Per ulteriori informazioni, consulta Incremento dell'MTBF.](#)
- Tempo medio di ripristino (MTTR): il periodo di tempo medio tra l'avvio di una pipeline fallita e l'inizio della successiva pipeline riuscita. Ti consigliamo di scegliere come target un MTTR il più basso possibile. [Per ulteriori informazioni, consulta Ridurre l'MTTR.](#)

Queste metriche aiutano i team a tenere traccia dei progressi verso la completa trasformazione in CI/CD. I team dovrebbero avere discussioni aperte con le parti interessate dell'organizzazione su quali dovrebbero essere gli obiettivi ottimali. Le situazioni e le esigenze variano notevolmente da un'organizzazione all'altra e persino da un team all'altro.

È molto importante ricordare che cambiamenti rapidi e drastici di solito aumentano il rischio di insorgenza di problemi. Stabilisci degli obiettivi per puntare a piccoli miglioramenti incrementali. Un

lead time ottimale comune per CI/CD pipeline complete è inferiore a 3 ore. Un team che inizia con un lead time di 5,2 giorni dovrebbe puntare a una riduzione di un giorno ogni poche settimane. Dopo che questo team ha raggiunto un lead time di un giorno o meno, può rimanere lì per diversi mesi e passare a un lead time più aggressivo solo se le parti interessate del team e dell'organizzazione lo ritengono necessario.

In che modo CI/CD i processi sono completamente diversi

Le pipeline CI/CD utilizzano un moderno flusso di lavoro basato su trunk, in cui gli sviluppatori uniscono aggiornamenti piccoli e frequenti in un ramo principale (o trunk) che viene creato e testato tramite la parte CD della pipeline. CI/CD Questo flusso di lavoro ha sostituito il flusso di lavoro Gitflow, in cui i rami di sviluppo e rilascio sono separati da una pianificazione dei rilasci. In molte organizzazioni, Gitflow è ancora un metodo popolare per il controllo e la distribuzione delle versioni. Tuttavia, ora è considerato obsoleto e può essere difficile integrarlo in una CI/CD pipeline.

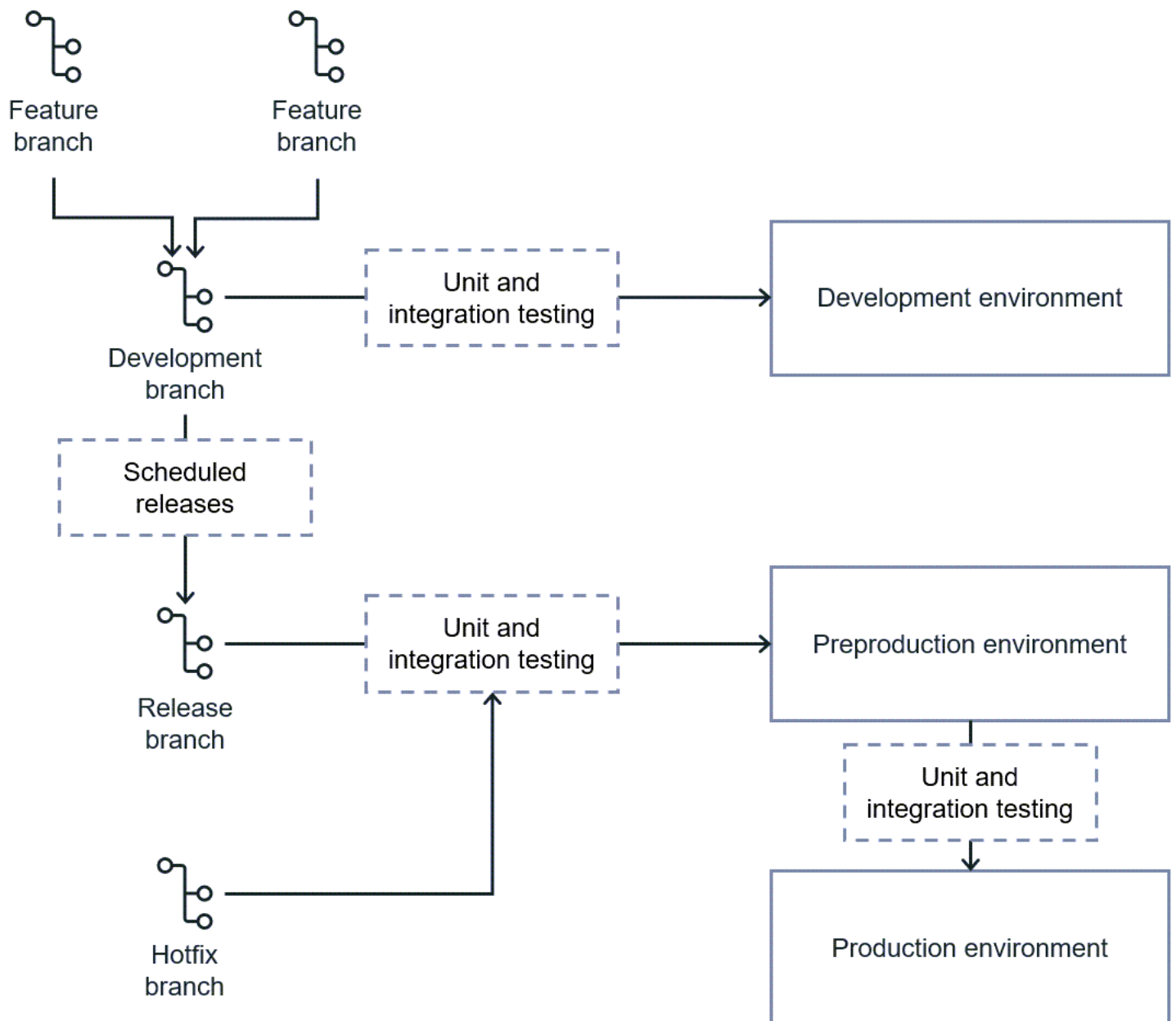
Per molte organizzazioni, la transizione da un flusso di lavoro Gitflow a un flusso di lavoro basato su trunk è stata incompleta e il risultato è che rimangono bloccate da qualche parte lungo il percorso e non migrano mai completamente a CI/CD. In qualche modo, le loro pipeline finiscono per rimanere aggrappate a certi resti del flusso di lavoro tradizionale, bloccate in uno stato di transizione tra passato e presente. Esamina le differenze nei flussi di lavoro Git, quindi scopri come l'utilizzo di un flusso di lavoro legacy può influire su quanto segue:

- [Integrità dell'ambiente](#)
- [Rilasci](#)
- [Sicurezza](#)

[Per semplificare l'identificazione dei resti di un flusso di lavoro Git legacy in una configurazione moderna, confrontiamo Gitflow con il moderno approccio basato su trunk.](#)

Approccio Gitflow

L'immagine seguente mostra un flusso di lavoro Gitflow. L'approccio Gitflow utilizza più rami per tenere traccia di diverse versioni del codice contemporaneamente. Pianifichi i rilasci degli aggiornamenti di un'applicazione per un certo periodo futuro mentre gli sviluppatori lavorano ancora sulla versione corrente del codice. I repository basati su Trunk possono utilizzare i flag di funzionalità per eseguire questa operazione, ma per impostazione predefinita sono integrati in Gitflow.



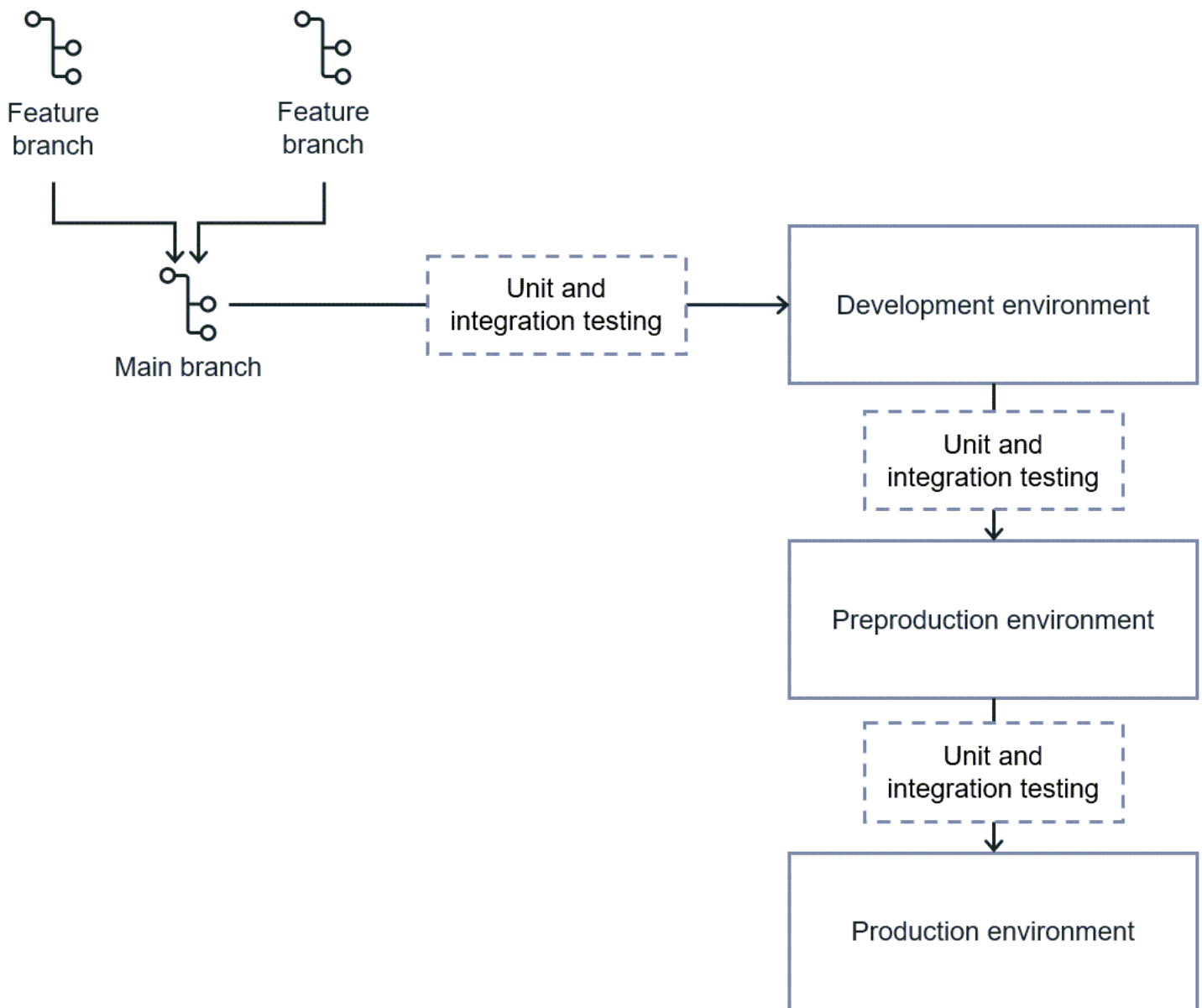
Un risultato dell'approccio Gitflow è che gli ambienti applicativi di solito non sono sincronizzati. In un'implementazione standard di Gitflow, gli ambienti di sviluppo riflettono lo stato attuale del codice, mentre gli ambienti di preproduzione e produzione rimangono bloccati sullo stato della base di codice della versione più recente.

Ciò complica le cose quando appare un difetto nell'ambiente di produzione, perché la base di codice su cui lavorano gli sviluppatori non può essere unita alla produzione senza esporre funzionalità inedite. Il modo in cui Gitflow gestisce questa situazione consiste nell'utilizzare un hotfix. Un ramo

hotfix viene creato dal ramo di rilascio e quindi distribuito direttamente negli ambienti superiori. Il ramo hotfix viene quindi unito al ramo di sviluppo per mantenere il codice aggiornato.

Approccio basato su Trunk

L'immagine seguente mostra un flusso di lavoro basato su trunk. In un flusso di lavoro basato su trunk, gli sviluppatori creano e testano le funzionalità localmente in un ramo di funzionalità, quindi uniscono tali modifiche nel ramo principale. Il ramo principale viene quindi integrato negli ambienti di sviluppo, preproduzione e produzione, in sequenza. I test unitari e di integrazione vengono eseguiti tra ogni ambiente.



Utilizzando questo flusso di lavoro, tutti gli ambienti utilizzano la stessa base di codice. Non è necessario un ramo hotfix per gli ambienti superiori perché è possibile implementare modifiche nel ramo principale senza esporre funzionalità non rilasciate. Si presume sempre che il ramo principale sia stabile, privo di difetti e pronto per il rilascio. Questo ti aiuta a integrarlo come fonte per una CI/CD pipeline, che può testare e distribuire automaticamente la tua base di codice in tutti gli ambienti della pipeline.

Vantaggi dell'integrità ambientale di un approccio basato su trunk

Come molti sviluppatori fanno, una modifica al codice a volte può creare un [effetto farfalla](#) (articolo di American Scientist), in cui una piccola deviazione apparentemente non correlata innesca una reazione a catena che causa risultati inaspettati. Gli sviluppatori devono quindi indagare a fondo per scoprire la causa principale.

Quando gli scienziati conducono un esperimento, separano i soggetti del test in due gruppi: il gruppo sperimentale e il gruppo di controllo. L'intenzione è di rendere il gruppo sperimentale e il gruppo di controllo completamente identici ad eccezione dell'oggetto testato nell'esperimento. Quando succede qualcosa nel gruppo sperimentale che non accade nel gruppo di controllo, l'unica causa può essere la cosa testata.

Pensate ai cambiamenti in una distribuzione come a un gruppo sperimentale e pensate a ciascun ambiente come a gruppi di controllo separati. I risultati dei test in un ambiente inferiore sono affidabili solo quando i controlli sono gli stessi di un ambiente superiore. Più gli ambienti si discostano, maggiore è la possibilità di scoprire difetti negli ambienti superiori. In altre parole, se le modifiche al codice dovessero fallire in fase di produzione, preferiremmo di gran lunga che fallissero prima nella versione beta in modo che non arrivino mai in produzione. Questo è il motivo per cui è necessario fare ogni sforzo per mantenere sincronizzato ogni ambiente, dall'ambiente di test più basso alla produzione stessa. Questa si chiama integrità dell'ambiente.

L'obiettivo di ogni CI/CD processo completo è quello di scoprire i problemi il prima possibile. Preservare l'integrità dell'ambiente utilizzando un approccio basato su trunk può praticamente eliminare la necessità di aggiornamenti rapidi. In un flusso di lavoro basato su trunk, è raro che un problema compaia per la prima volta nell'ambiente di produzione.

In un approccio Gitflow, dopo che un hotfix è stato distribuito direttamente negli ambienti superiori, viene quindi aggiunto al ramo di sviluppo. In questo modo viene preservata la correzione per le versioni future. Tuttavia, l'hotfix è stato sviluppato e testato direttamente sullo stato corrente dell'applicazione. Anche se l'hotfix funziona perfettamente in produzione, esiste la possibilità che

si verifichino problemi quando interagisce con le nuove funzionalità del ramo di sviluppo. Poiché in genere non è consigliabile implementare un hotfix per un hotfix, gli sviluppatori dedicano più tempo a cercare di adattare l'hotfix all'ambiente di sviluppo. In molti casi, ciò può comportare un notevole indebitamento tecnico e ridurre la stabilità complessiva dell'ambiente di sviluppo.

Quando si verifica un errore in un ambiente, tutte le modifiche vengono ripristinate in modo che l'ambiente ritorni allo stato precedente. Qualsiasi modifica a una base di codice dovrebbe riavviare la pipeline sin dalla prima fase. Quando si verifica un problema nell'ambiente di produzione, la correzione dovrebbe riguardare anche l'intera pipeline. Il tempo aggiuntivo necessario per gestire gli ambienti inferiori è in genere trascurabile rispetto ai problemi che si evitano utilizzando questo approccio. Poiché l'intero scopo degli ambienti inferiori è quello di catturare gli errori prima che raggiungano la produzione, aggirare questi ambienti attraverso un approccio Gitflow è un rischio inefficiente e non necessario.

Scopri i vantaggi di un approccio basato su trunk

Uno degli aspetti che spesso rende necessario un hotfix è che, in un workflow legacy, lo stato dell'applicazione su cui stanno lavorando gli sviluppatori potrebbe contenere diverse funzionalità inedite che non sono ancora disponibili in produzione. L'ambiente di produzione e l'ambiente di sviluppo si sincronizzano solo quando si verifica un rilascio pianificato e quindi ricominciano immediatamente a divergere fino alla successiva versione pianificata.

È possibile disporre di rilasci programmati all'interno di un CI/CD processo completo. È possibile ritardare il rilascio del codice alla produzione utilizzando i flag di funzionalità. Tuttavia, un CI/CD processo completo consente una maggiore flessibilità rendendo superflui i rilasci programmati. Dopotutto, continuo è una parola chiave in CI/CD e ciò suggerisce che le modifiche vengono rilasciate non appena sono pronte. Evitate di mantenere un ambiente di rilascio separato, quasi sempre non sincronizzato con gli ambienti di test precedenti.

Se una pipeline non è completamente CI/CD, la divergenza tra gli ambienti superiori e quelli inferiori di solito si verifica a livello di filiale. Gli sviluppatori lavorano in un ramo di sviluppo e gestiscono un ramo di rilascio separato che viene aggiornato solo quando è il momento di una versione pianificata. Man mano che il ramo di rilascio e il ramo di sviluppo divergono, possono sorgere altre complicazioni.

Oltre alla mancata sincronizzazione degli ambienti, man mano che gli sviluppatori lavorano nel ramo di sviluppo e si abituano a uno stato dell'applicazione molto più avanzato di quello in produzione, devono riadattarsi allo stato di produzione ogni volta che si verifica un problema. Lo stato del ramo di sviluppo potrebbe riguardare molte funzionalità prima della produzione. Quando gli sviluppatori

lavorano ogni giorno in quel settore, è difficile ricordare cosa viene rilasciato e cosa non viene rilasciato alla produzione. Ciò aumenta il rischio che vengano introdotti nuovi bug durante il processo di correzione di altri bug. Il risultato è un ciclo apparentemente infinito di correzioni che allungano le tempistiche e ritardano il rilascio delle funzionalità per settimane, mesi o addirittura anni.

Vantaggi in termini di sicurezza di un approccio basato su trunk

Un CI/CD processo completo fornisce un approccio all'implementazione basato su un'unica fonte di verità completamente automatizzato. La pipeline ha un unico punto di ingresso. Gli aggiornamenti software entrano nella pipeline all'inizio e vengono passati così come sono da un ambiente all'altro. Se viene rilevato un problema in qualsiasi fase della pipeline, le modifiche al codice che lo risolvono devono passare attraverso lo stesso processo e iniziare nella prima fase. La riduzione dei punti di ingresso in una pipeline riduce anche i possibili modi in cui le vulnerabilità possono essere introdotte nella pipeline.

Inoltre, poiché il punto di ingresso è il punto più lontano possibile dall'ambiente di produzione, ciò riduce drasticamente la probabilità che le vulnerabilità raggiungano la produzione. Se si implementa un processo di approvazione manuale in una pipeline CI/CD completa, è comunque possibile decidere se promuovere o meno le modifiche all'ambiente successivo. Il decisore non è necessariamente la stessa persona che implementa le modifiche. Ciò separa le responsabilità dell'autore delle modifiche al codice e dell'approvatore di tali modifiche. Inoltre, consente a un responsabile organizzativo meno tecnico di svolgere il ruolo di approvatore.

Infine, il punto di ingresso singolo consente di limitare l'accesso in scrittura alla console dell'interfaccia utente (UI) dell'ambiente di produzione a pochi o addirittura zero utenti. Riducendo il numero di utenti che possono apportare modifiche manuali alla console, riduci il rischio di eventi di sicurezza. La capacità di gestire manualmente la console nell'ambiente di produzione è molto più necessaria nei flussi di lavoro legacy che in un approccio CI/CD automatizzato. Queste modifiche manuali sono più difficili da tracciare, rivedere e testare. Di solito vengono eseguite per risparmiare tempo, ma a lungo termine aggiungono un notevole debito tecnico al progetto.

I problemi di sicurezza delle console non sono necessariamente causati da malintenzionati. Molti dei problemi che si verificano nella console sono accidentali. L'esposizione accidentale alla sicurezza è molto comune e ha portato alla nascita del modello di sicurezza zero-trust. Questo modello presuppone, in parte, che gli incidenti di sicurezza siano meno probabili quando anche il personale interno ha il minor accesso possibile, le cosiddette autorizzazioni con privilegi minimi. Preservare l'integrità dell'ambiente di produzione limitando tutti i processi a una pipeline automatizzata elimina praticamente il rischio di problemi di sicurezza relativi alla console.

Cartina tornasole per CI/CD condotte

In chimica, la cartina di tornasole è una sottile striscia di carta trattata con uno speciale colorante rosso o blu che viene utilizzato per determinare l'acidità di una sostanza. Un acido diventa blu la cartina tornasole rossa, una base diventa blu la cartina tornasole rossa e le sostanze neutre non influiscono affatto sul colore della carta.

Il modo in cui la cartina di tornasole determina l'acidità consiste nel misurare il livello di pH di una sostanza. Se un livello di pH è superiore a 8, è acido; se è inferiore a 5, è basico; e se è compreso tra 5 e 8, è neutro. Allo stesso modo, la [CI/CD cartina di tornasole](#) ti aiuta a misurare il CI/CD livello della tua pipeline.

Per verificare se la pipeline è completamente CI/CD

1. Inizia con un punteggio pari a 0.
2. Rispondi a ciascuna delle seguenti domande e aggiungi 1 al tuo punteggio per ogni volta che rispondi sì:
 - Ciascuno dei nostri repository ha esattamente un ramo principale che viene utilizzato per l'implementazione negli ambienti?
 - Immettiamo spesso codice nel ramo principale ed evitiamo di avere rami di funzionalità a lunga durata?
 - La nostra pipeline ha un unico punto di ingresso? In altre parole, la nostra pipeline estrae il codice da ogni repository esattamente una volta?
 - Disponiamo di più di un ambiente di implementazione?
 - Quando la pipeline non è in esecuzione, i nostri ambienti superiori e inferiori sono generalmente sincronizzati?
 - Eseguiamo test sul codice prima della distribuzione?
 - Eseguiamo test su un ambiente prima di passare all'ambiente successivo?
 - La nostra pipeline esegue un rollback completo ed esce dopo un errore?
 - La nostra pipeline si riavvia dalla prima fase durante il ripristino da un guasto?
 - Seguiamo lo stesso processo per correggere i bug in produzione che seguiamo per rilasciare le funzionalità in produzione?
 - Utilizziamo modelli di tipo Infrastructure as Code (IaC) per implementare il codice?

3. Rispondi a ciascuna delle seguenti domande e aggiungi 1 al tuo punteggio per ogni volta che rispondi no:
- Effettuiamo mai l'implementazione direttamente in un ambiente di distribuzione da filiali diverse dalla filiale principale?
 - Effettuiamo mai l'implementazione direttamente da una filiale a un ambiente superiore o di produzione?
 - Troviamo spesso bug negli ambienti superiori che non erano presenti negli ambienti inferiori?
 - Riusciamo mai a bypassare gli ambienti inferiori durante un'implementazione?
 - Dobbiamo aspettare l'orario di rilascio programmato per passare alla produzione?
 - Effettuiamo regolarmente aggiornamenti nella console dell'ambiente di produzione?
 - Esistono passaggi di distribuzione manuali che devono essere eseguiti nella console dell'ambiente di produzione per completare l'implementazione?
 - Più di una persona ha accesso in scrittura all'ambiente di produzione?
 - Più di cinque persone hanno accesso in scrittura all'ambiente di produzione?
4. Dividi il tuo punteggio per 2. Questo è il CI/CD punteggio della tua pipeline.
5. Confronta il CI/CD punteggio della pipeline con la tabella seguente per determinare il livello della pipeline. CI/CD

Punteggio CI/CD	Livello CI/CD
9.5 o superiore	Completamente CI/CD
8—9	Principalmente CI/CD
5-7	Neutral
Inferiore a 5	Non CI/CD

Se hai ottenuto un punteggio inferiore a 8, ti consigliamo di fissare un obiettivo per passare gradualmente al livello successivo. Una volta raggiunto tale obiettivo, gli stakeholder del prodotto dovrebbero valutare se e quando fissare un nuovo obiettivo. L'obiettivo di questo esercizio non è necessariamente quello di promuovere un cambiamento nella pipeline, ma piuttosto di sensibilizzare l'opinione pubblica su come si presenta un processo di CI/CD implementazione completa e sulla posizione attuale delle pipeline in questo spettro.

Le migliori pratiche per le CI/CD pipeline

Di seguito sono riportate le best practice per una CI/CD pipeline completa:

- Proteggi l'ambiente di produzione: poiché è possibile eseguire praticamente tutto il necessario per la manutenzione dell'account e dell'ambiente utilizzando IaC, è importante fare ogni sforzo per proteggere l'ambiente di produzione limitando l'accesso da console e programmatico. Consigliamo di limitare l'accesso a pochi o addirittura a zero utenti. Quando si distribuisce IaC tramite AWS CloudFormation, l'utente necessita di autorizzazioni limitate. La maggior parte delle autorizzazioni viene assegnata al CloudFormation servizio tramite un ruolo di servizio. Per ulteriori informazioni, consulta [Ruolo di servizio](#) nella CloudFormation documentazione e [Implementazione delle politiche per le autorizzazioni con privilegi minimi](#) per. AWS CloudFormation
- Crea account separati per ogni ambiente: dedicando un account separato a ciascun ambiente, puoi semplificare il processo di implementazione e creare controlli di accesso dettagliati a livello di account. Quando più ambienti condividono risorse, si riduce l'integrità dell'ambiente come unità isolata. È meglio mantenere gli ambienti sincronizzati e distinti. Questo è ancora più importante per l'ambiente di produzione, perché tutto ciò che si trova in quell'account deve essere trattato come una risorsa di produzione.
- Limita le informazioni di identificazione personale (PII) all'ambiente di produzione: sia per la sicurezza che per la protezione dai rischi di responsabilità, proteggi le informazioni personali il più possibile. Se possibile, in ambienti inferiori, utilizza dati anonimi o campionari invece di copiare dati potenzialmente sensibili dall'ambiente di produzione.
- Revisione del codice negli archivi: un processo completamente CI/CD riduce i punti di ingresso di una pipeline a un unico punto, che deve essere protetto. Per questo motivo, si consiglia di richiedere più revisioni del codice prima di unire i rami delle funzionalità nel ramo principale. Queste revisioni del codice possono essere condotte da qualsiasi membro del team qualificato, ma almeno un membro senior deve esaminarle. Il codice deve essere testato rigorosamente dal revisore. Dopotutto, il modo migliore per risolvere i problemi in una pipeline è evitare di introdurli. Inoltre, è importante risolvere tutti i commenti fatti da qualsiasi revisore prima di procedere all'unione. Questa risoluzione potrebbe semplicemente essere una spiegazione del motivo per cui non sono necessarie modifiche, ma rispondere a tutti i commenti è un importante controllo aggiuntivo per evitare l'introduzione di problemi nella pipeline.
- Effettua fusioni piccole e frequenti: per sfruttare appieno l'integrazione continua, è una buona idea inserire continuamente anche le modifiche locali nella pipeline. Dopotutto, è molto più vantaggioso

che gli ambienti di sviluppo rimangano sincronizzati se anche gli ambienti locali li mantengono aggiornati.

Per ulteriori best practice per le CI/CD pipeline, consulta il [Riepilogo delle migliori pratiche](#) per la pratica dell'integrazione continua e della distribuzione continua su. AWS

Domande frequenti

Quali sono alcuni indicatori chiave del fatto che il mio processo di implementazione non è completamente CI/CD?

L'indicatore più comune è quando ci sono più rami di repository che rappresentano ambienti separati in una pipeline. I repository in un CI/CD processo completo utilizzano un flusso di lavoro basato su trunk, in cui una filiale funge da unica fonte di verità per le implementazioni di quel repository. Per ulteriori informazioni, consulta [Approccio basato su Trunk](#). Altri indicatori includono le fasi di implementazione manuali diverse dalle semplici decisioni «go go» o «no go go», l'uso di hotfix e i rilasci pianificati.

Cosa succede se desidero utilizzare un CI/CD processo completo ma comunque pianificare il rilascio di determinate funzionalità per momenti specifici?

Questo di solito viene fatto con i flag di funzionalità. In questo processo, le distribuzioni vengono comunque eseguite continuamente, ma alcune funzionalità vengono nascoste utilizzando chiusure condizionali nel codice fino al momento del rilascio.

Cosa succede se alcune fasi del mio processo di distribuzione non possono essere automatizzate?

Uno degli obiettivi di una CI/CD pipeline completa è ridurre al minimo la necessità di processi manuali, ma ci sono sicuramente potenziali casi d'uso in cui potrebbero essere necessari processi manuali. In effetti, i processi di sola lettura, come la consultazione dei registri delle applicazioni, possono spesso essere eseguiti in ambienti di produzione con rischi minimi. Tuttavia, si consiglia vivamente di considerare le azioni di scrittura manuale in produzione come l'ultima risorsa.

E se il mio staff tecnico fosse più a suo agio con i flussi di lavoro esistenti piuttosto che con un CI/CD processo completo?

È normale che il personale tecnico sia restio ai cambiamenti più importanti, soprattutto quando qualcosa che prima era una best practice viene sostituito da qualcosa di più nuovo. La tecnologia si muove rapidamente e i miglioramenti vengono costantemente scoperti. Sebbene un certo grado di scetticismo sia una buona qualità per uno staff tecnico, è altrettanto importante che sia aperto al cambiamento. Non agite troppo velocemente con personale scettico, perché deve gestire le modifiche al sistema prima che vengano implementate. La chiave è impedire agli scettici di rimanere statici per sempre.

Cosa succede se i miei ambienti sono suddivisi in più account? Posso ancora utilizzare un CI/CD processo completo?

Sì, in effetti, si consiglia di utilizzare un account separato per ogni ambiente. Per ulteriori informazioni su una pipeline che attiva fasi in account diversi, consulta [Creare una pipeline in CodePipeline cui vengono utilizzate le risorse](#) di un altro Account AWS

Fasi successive

Utilizza la [Cartina tornasole per CI/CD condotte](#) sezione per valutare i processi all'interno della tua organizzazione DevOps. Determina se i tuoi processi sono CI/CD. If they aren't, decide whether they need improvement to take full advantage of the benefits of CI/CD implementati completamente.

Come fai a sapere quando hai finito? Bene, la risposta è che molte organizzazioni non finiscono mai. Si fermano da qualche parte lungo la strada, in un punto adatto al loro caso d'uso. Sebbene una CI/CD pipeline completa sia lo scenario migliore, dipende in larga misura dalla situazione organizzativa e dalle parti interessate alla base della decisione. Le parti interessate devono decidere quale fase dell' CI/CD implementazione è più adatta al loro caso d'uso e il modo migliore per mappare la progressione verso le fasi successive.

Per ulteriori informazioni sulla progettazione e la costruzione di CI/CD condotte, vedere. [Risorse](#)

Risorse

AWS documentazione e riferimenti

- [AWS Deployment Pipeline Reference Architecture](#)
- [Che cos'è la distribuzione continua?](#)
- [Praticare l'integrazione continua e la distribuzione continua su AWS](#) (AWS white paper)
- [Configura una CI/CD pipeline](#) su (tutorial pratico) AWSAWS
- [Crea una pipeline Regioni AWS che non supporti AWS CodePipeline\(Prescriptive Guidance\)AWS](#)
- [Architettura di riferimento delle pipeline di distribuzione e implementazioni di riferimento](#) (post sul blog)AWS

Servizi e strumenti

- [La cartina di CI/CD tornasole](#)
- [AWS Cloud Development Kit \(AWS CDK\)](#)
- [AWS CloudFormation](#)
- [AWS CodePipeline](#)

Cronologia dei documenti

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

Modifica	Descrizione	Data
Pubblicazione iniziale	—	25 agosto 2023

AWS Glossario delle linee guida prescrittive

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

Numeri

7 R

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

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

A

ABAC

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

servizi astratti

Vedi [servizi gestiti](#).

ACIDO

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

migrazione attiva-attiva

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

migrazione attiva-passiva

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

funzione di aggregazione

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

Intelligenza artificiale

Vedi [intelligenza artificiale](#).

AIOps

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

anonimizzazione

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

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

anti-modello

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

controllo delle applicazioni

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

portfolio di applicazioni

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

intelligenza artificiale (IA)

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

operazioni di intelligenza artificiale (AIOps)

Il processo di utilizzo delle tecniche di machine learning per risolvere problemi operativi, ridurre gli incidenti operativi e l'intervento umano e aumentare la qualità del servizio. Per ulteriori informazioni su come AIOps viene utilizzata nella strategia di AWS migrazione, consulta la [guida all'integrazione delle operazioni](#).

crittografia asimmetrica

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

atomicità, consistenza, isolamento, durabilità (ACID)

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

Controllo degli accessi basato su attributi (ABAC)

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

fonte di dati autorevole

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

Zona di disponibilità

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

AWS Cloud Adoption Framework (CAF)AWS

Un framework di linee guida e best practice AWS per aiutare le organizzazioni a sviluppare un piano efficiente ed efficace per passare con successo al cloud. AWS CAF organizza le linee guida in sei aree di interesse chiamate prospettive: business, persone, governance, piattaforma, sicurezza e operazioni. Le prospettive relative ad azienda, persone e governance si concentrano sulle competenze e sui processi aziendali; le prospettive relative alla piattaforma, alla sicurezza e alle operazioni si concentrano sulle competenze e sui processi tecnici. Ad esempio, la prospettiva relativa alle persone si rivolge alle parti interessate che gestiscono le risorse umane (HR), le funzioni del personale e la gestione del personale. In questa prospettiva, AWS CAF fornisce linee guida per lo sviluppo delle persone, la formazione e le comunicazioni per aiutare a preparare l'organizzazione all'adozione del cloud di successo. Per ulteriori informazioni, consulta il [sito web di AWS CAF](#) e il [white paper AWS CAF](#).

AWS Workload Qualification Framework (WQF)AWS

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

B

bot difettoso

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

BCP

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

grafico comportamentale

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

sistema big-endian

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

Classificazione binaria

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

filtro Bloom

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

implementazione blu/verde

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

bot

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

botnet

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

ramo

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

accesso break-glass

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

strategia brownfield

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

cache del buffer

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

capacità di business

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

pianificazione della continuità operativa (BCP)

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

C

CAF

Vedi [Cloud Adoption AWS Framework](#).

implementazione canaria

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

CCoE

Vedi [Cloud Center of Excellence](#).

CDC

Vedi [Change Data Capture](#).

Change Data Capture (CDC)

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

ingegneria del caos

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

CI/CD

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

classificazione

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

crittografia lato client

Crittografia dei dati a livello locale, prima che il destinatario li Servizio AWS riceva.

Centro di eccellenza cloud (CCoE)

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

cloud computing

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

modello operativo cloud

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

fasi di adozione del cloud

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

- Progetto: esecuzione di alcuni progetti relativi al cloud per scopi di dimostrazione e apprendimento
- Fondamento: effettuare investimenti fondamentali per scalare l'adozione del cloud (ad esempio, creazione di una landing zone, definizione di una CCo E, definizione di un modello operativo)
- Migrazione: migrazione di singole applicazioni
- Reinvenzione: ottimizzazione di prodotti e servizi e innovazione nel cloud

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

CMDB

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

repository di codice

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

cache fredda

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

dati freddi

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

visione artificiale (CV)

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

deriva della configurazione

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

database di gestione della configurazione (CMDB)

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

Pacchetto di conformità

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

integrazione e distribuzione continua (continuous integration and continuous delivery, CI/CD)

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

[della distribuzione continua](#). CD può anche significare continuous deployment (implementazione continua). Per ulteriori informazioni, consulta [Distribuzione continua e implementazione continua a confronto](#).

CV

Vedi [visione artificiale](#).

D

dati a riposo

Dati stazionari nella rete, ad esempio i dati archiviati.

classificazione dei dati

Un processo per identificare e classificare i dati nella rete in base alla loro criticità e sensibilità. È un componente fondamentale di qualsiasi strategia di gestione dei rischi di sicurezza informatica perché consente di determinare i controlli di protezione e conservazione appropriati per i dati. La classificazione dei dati è un componente del pilastro della sicurezza nel AWS Well-Architected Framework. Per ulteriori informazioni, consulta [Classificazione dei dati](#).

deriva dei dati

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

dati in transito

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

rete di dati

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

riduzione al minimo dei dati

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

perimetro dei dati

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

pre-elaborazione dei dati

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

provenienza dei dati

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

soggetto dei dati

Un individuo i cui dati vengono raccolti ed elaborati.

data warehouse

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

linguaggio di definizione del database (DDL)

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

linguaggio di manipolazione del database (DML)

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

DDL

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

deep ensemble

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

deep learning

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

defense-in-depth

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

amministratore delegato

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

implementazione

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

Ambiente di sviluppo

[Vedi ambiente.](#)

controllo di rilevamento

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

mappatura del flusso di valore dello sviluppo (DVSM)

Un processo utilizzato per identificare e dare priorità ai vincoli che influiscono negativamente sulla velocità e sulla qualità nel ciclo di vita dello sviluppo del software. DVSM estende il processo di

mappatura del flusso di valore originariamente progettato per pratiche di produzione snella. Si concentra sulle fasi e sui team necessari per creare e trasferire valore attraverso il processo di sviluppo del software.

gemello digitale

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

tabella delle dimensioni

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

disastro

Un evento che impedisce a un carico di lavoro o a un sistema di raggiungere gli obiettivi aziendali nella sua sede principale di implementazione. Questi eventi possono essere disastri naturali, guasti tecnici o il risultato di azioni umane, come errori di configurazione involontari o attacchi di malware.

disaster recovery (DR)

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

DML

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

progettazione basata sul dominio

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

DOTT.

Vedi [disaster recovery](#).

rilevamento della deriva

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

DVSM

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

E

EDA

Vedi [analisi esplorativa dei dati](#).

MODIFICA

Vedi [scambio elettronico di dati](#).

edge computing

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

scambio elettronico di dati (EDI)

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

crittografia

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

chiave crittografica

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

endianità

L'ordine in cui i byte vengono archiviati nella memoria del computer. I sistemi big-endian memorizzano per primo il byte più importante. I sistemi little-endian memorizzano per primo il byte meno importante.

endpoint

[Vedi](#) service endpoint.

servizio endpoint

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

pianificazione delle risorse aziendali (ERP)

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

crittografia envelope

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

ambiente

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

- ambiente di sviluppo: un'istanza di un'applicazione in esecuzione disponibile solo per il team principale responsabile della manutenzione dell'applicazione. Gli ambienti di sviluppo vengono utilizzati per testare le modifiche prima di promuoverle negli ambienti superiori. Questo tipo di ambiente viene talvolta definito ambiente di test.
- ambienti inferiori: tutti gli ambienti di sviluppo di un'applicazione, ad esempio quelli utilizzati per le build e i test iniziali.

- ambiente di produzione: un'istanza di un'applicazione in esecuzione a cui gli utenti finali possono accedere. In una CI/CD pipeline, l'ambiente di produzione è l'ultimo ambiente di distribuzione.
- ambienti superiori: tutti gli ambienti a cui possono accedere utenti diversi dal team di sviluppo principale. Si può trattare di un ambiente di produzione, ambienti di preproduzione e ambienti per i test di accettazione da parte degli utenti.

epica

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

ERP

Vedi [pianificazione delle risorse aziendali](#).

analisi esplorativa dei dati (EDA)

Il processo di analisi di un set di dati per comprenderne le caratteristiche principali. Si raccolgono o si aggregano dati e quindi si eseguono indagini iniziali per trovare modelli, rilevare anomalie e verificare ipotesi. L'EDA viene eseguita calcolando statistiche di riepilogo e creando visualizzazioni di dati.

F

tabella dei fatti

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

fallire velocemente

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

limite di isolamento dei guasti

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

ramo di funzionalità

Vedi [filiale](#).

caratteristiche

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

importanza delle caratteristiche

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

trasformazione delle funzionalità

Per ottimizzare i dati per il processo di machine learning, incluso l'arricchimento dei dati con fonti aggiuntive, il dimensionamento dei valori o l'estrazione di più set di informazioni da un singolo campo di dati. Ciò consente al modello di ML di trarre vantaggio dai dati. Ad esempio, se suddividi la data "2021-05-27 00:15:37" in "2021", "maggio", "giovedì" e "15", puoi aiutare l'algoritmo di apprendimento ad apprendere modelli sfumati associati a diversi componenti dei dati.

prompt con pochi scatti

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

FGAC

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

controllo granulare degli accessi (FGAC)

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

migrazione flash-cut

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

FM

[Vedi modello di base.](#)

modello di fondazione (FM)

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

G

IA generativa

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

blocco geografico

Vedi [restrizioni geografiche](#).

limitazioni geografiche (blocco geografico)

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

Flusso di lavoro di GitFlow

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

immagine dorata

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

strategia greenfield

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

guardrail

Una regola di alto livello che aiuta a governare le risorse, le politiche e la conformità tra le unità organizzative (). OUs I guardrail preventivi applicano le policy per garantire l'allineamento agli standard di conformità. Vengono implementati utilizzando le policy di controllo dei servizi e i limiti delle autorizzazioni IAM. I guardrail di rilevamento rilevano le violazioni delle policy e i problemi di conformità e generano avvisi per porvi rimedio. Sono implementati utilizzando Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector e controlli personalizzati AWS Lambda .

H

AH

Vedi [disponibilità elevata](#).

migrazione di database eterogenea

Migrazione del database di origine in un database di destinazione che utilizza un motore di database diverso (ad esempio, da Oracle ad Amazon Aurora). La migrazione eterogenea fa in

genere parte di uno sforzo di riprogettazione e la conversione dello schema può essere un'attività complessa. [AWS offre AWS SCT](#) che aiuta con le conversioni dello schema.

alta disponibilità (HA)

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

modernizzazione storica

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

dati di blocco

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

migrazione di database omogenea

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

dati caldi

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

hotfix

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

periodo di hypercare

Subito dopo la conversione, il periodo di tempo in cui un team di migrazione gestisce e monitora le applicazioni migrate nel cloud per risolvere eventuali problemi. In genere, questo periodo dura

da 1 a 4 giorni. Al termine del periodo di hypercare, il team addetto alla migrazione in genere trasferisce la responsabilità delle applicazioni al team addetto alle operazioni cloud.

I

IaC

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

Policy basata su identità

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

applicazione inattiva

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

IIoT

Vedi [Industrial Internet of Things](#).

infrastruttura immutabile

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

VPC in ingresso (ingress)

In un'architettura AWS multi-account, un VPC che accetta, ispeziona e indirizza le connessioni di rete dall'esterno di un'applicazione. La [AWS Security Reference Architecture](#) consiglia di configurare l'account di rete con funzionalità in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

migrazione incrementale

Una strategia di conversione in cui si esegue la migrazione dell'applicazione in piccole parti anziché eseguire una conversione singola e completa. Ad esempio, inizialmente potresti spostare

I

solo alcuni microservizi o utenti nel nuovo sistema. Dopo aver verificato che tutto funzioni correttamente, puoi spostare in modo incrementale microservizi o utenti aggiuntivi fino alla disattivazione del sistema legacy. Questa strategia riduce i rischi associati alle migrazioni di grandi dimensioni.

Industria 4.0

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

infrastruttura

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

infrastruttura come codice (IaC)

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

IIoInternet delle cose industriale (T)

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

VPC di ispezione

In un'architettura AWS multi-account, un VPC centralizzato che gestisce le ispezioni del traffico di rete tra VPCs (nello stesso o in modo diverso Regioni AWS), Internet e le reti locali. La [AWS Security Reference Architecture](#) consiglia di configurare l'account di rete con informazioni in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

Internet of Things (IoT)

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

interpretabilità

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

IoT

Vedi [Internet of Things](#).

libreria di informazioni IT (ITIL)

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

gestione dei servizi IT (ITSM)

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

ITIL

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

ITSM

Vedi [Gestione dei servizi IT](#).

L

controllo degli accessi basato su etichette (LBAC)

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

zona di destinazione

Una landing zone è un AWS ambiente multi-account ben progettato, scalabile e sicuro. Questo è un punto di partenza dal quale le organizzazioni possono avviare e distribuire rapidamente carichi di lavoro e applicazioni con fiducia nel loro ambiente di sicurezza e infrastruttura. Per ulteriori

informazioni sulle zone di destinazione, consulta la sezione [Configurazione di un ambiente AWS multi-account sicuro e scalabile](#).

modello linguistico di grandi dimensioni (LLM)

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

migrazione su larga scala

Una migrazione di 300 o più server.

BIANCO

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

Privilegio minimo

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

eseguire il rehosting (lift and shift)

Vedi [7 R](#).

sistema little-endian

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

LLM

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

ambienti inferiori

Vedi [ambiente](#).

M

machine learning (ML)

Un tipo di intelligenza artificiale che utilizza algoritmi e tecniche per il riconoscimento e l'apprendimento di schemi. Il machine learning analizza e apprende dai dati registrati, come i dati

dell'Internet delle cose (IoT), per generare un modello statistico basato su modelli. Per ulteriori informazioni, consulta la sezione [Machine learning](#).

ramo principale

Vedi [filiale](#).

malware

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

servizi gestiti

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

sistema di esecuzione della produzione (MES)

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

MAP

Vedi [Migration Acceleration Program](#).

meccanismo

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

account membro

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

MEH

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

Message Queuing Telemetry Transport (MQTT)

[Un protocollo di comunicazione machine-to-machine \(M2M\) leggero, basato sul modello di pubblicazione/sottoscrizione, per dispositivi IoT con risorse limitate.](#)

microservizio

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

architettura di microservizi

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

Programma di accelerazione della migrazione (MAP)

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

migrazione su larga scala

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

fabbrica di migrazione

Team interfunzionali che semplificano la migrazione dei carichi di lavoro attraverso approcci automatizzati e agili. I team di Migration Factory in genere includono addetti alle operazioni,

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

metadati di migrazione

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

modello di migrazione

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

Valutazione del portfolio di migrazione (MPA)

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

valutazione della preparazione alla migrazione (MRA)

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

strategia di migrazione

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

ML

[Vedi machine learning.](#)

modernizzazione

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

valutazione della preparazione alla modernizzazione

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

applicazioni monolitiche (monoliti)

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

MAPPA

Vedi [Migration Portfolio Assessment](#).

MQTT

Vedi [Message Queuing Telemetry Transport](#).

classificazione multiclasse

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

infrastruttura mutabile

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

O

OAC

Vedi [Origin Access Control](#).

QUERCIA

Vedi [Origin Access Identity](#).

OCM

Vedi [gestione delle modifiche organizzative](#).

migrazione offline

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

OI

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

OLA

Vedi accordo a [livello operativo](#).

migrazione online

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

OPC-UA

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

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

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

accordo a livello operativo (OLA)

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

revisione della prontezza operativa (ORR)

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

tecnologia operativa (OT)

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

integrazione delle operazioni (OI)

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

trail organizzativo

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

gestione del cambiamento organizzativo (OCM)

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

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

controllo dell'accesso all'origine (OAC)

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

PUT DELETE

identità di accesso origine (OAI)

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

ORR

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

NON

Vedi la [tecnologia operativa](#).

VPC in uscita (egress)

In un'architettura AWS multi-account, un VPC che gestisce le connessioni di rete avviate dall'interno di un'applicazione. La [AWS Security Reference Architecture](#) consiglia di configurare l'account di rete con funzionalità in entrata, in uscita e di ispezione VPCs per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

P

limite delle autorizzazioni

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

informazioni di identificazione personale (PII)

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

Informazioni che consentono l'identificazione personale degli utenti

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

playbook

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

PLC

Vedi [controllore logico programmabile](#).

PLM

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

policy

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

persistenza poliglotta

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

valutazione del portfolio

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

predicate

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

predicato pushdown

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

controllo preventivo

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

principale

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

privacy fin dalla progettazione

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

zone ospitate private

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

controllo proattivo

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

gestione del ciclo di vita del prodotto (PLM)

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

Ambiente di produzione

[Vedi ambiente.](#)

controllore logico programmabile (PLC)

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

concatenamento rapido

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

pseudonimizzazione

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

publish/subscribe (pub/sub)

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

Q

Piano di query

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

regressione del piano di query

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

R

Matrice RACI

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

RAG

Vedi [Retrieval](#) Augmented Generation.

ransomware

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

Matrice RASCI

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

RCAC

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

replica di lettura

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

riprogettare

Vedi [7 Rs](#).

obiettivo del punto di ripristino (RPO)

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

obiettivo del tempo di ripristino (RTO)

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

rifattorizzare

Vedi [7 R.](#)

Region

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

regressione

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

riospitare

Vedi [7 R.](#)

rilascio

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

trasferisco

Vedi [7 Rs.](#)

ripiattaforma

Vedi [7 Rs.](#)

riacquisto

Vedi [7 Rs.](#)

resilienza

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

policy basata su risorse

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

matrice di assegnazione di responsabilità (RACI)

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

controllo reattivo

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

retain

Vedi [7 R](#).

andare in pensione

Vedi [7 Rs](#).

Retrieval Augmented Generation (RAG)

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

rotazione

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

controllo dell'accesso a righe e colonne (RCAC)

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

RPO

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

VERSO

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

runbook

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

S

SAML 2.0

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

SCADA

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

SCP

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

Secret

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

sicurezza fin dalla progettazione

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

controllo di sicurezza

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

rafforzamento della sicurezza

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

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

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

automazione della risposta alla sicurezza

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

Crittografia lato server

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

Policy di controllo dei servizi (SCP)

Una politica che fornisce il controllo centralizzato sulle autorizzazioni per tutti gli account di un'organizzazione in. AWS Organizations SCPs definire barriere o fissare limiti alle azioni che un amministratore può delegare a utenti o ruoli. È possibile utilizzarli SCPs come elenchi consentiti o elenchi di rifiuto, per specificare quali servizi o azioni sono consentiti o proibiti. Per ulteriori informazioni, consulta [le politiche di controllo del servizio](#) nella AWS Organizations documentazione.

endpoint del servizio

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

accordo sul livello di servizio (SLA)

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

indicatore del livello di servizio (SLI)

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

obiettivo a livello di servizio (SLO)

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

Modello di responsabilità condivisa

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

SIEM

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

punto di errore singolo (SPOF)

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

SLAM

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

SLI

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

LENTA

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

split-and-seed modello

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

SPOF

Vedi [punto di errore singolo](#).

schema a stella

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

modello del fico strangolatore

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

sottorete

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

controllo di supervisione e acquisizione dati (SCADA)

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

crittografia simmetrica

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

test sintetici

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

prompt di sistema

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

T

tag

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

variabile di destinazione

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

elenco di attività

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

ambiente di test

[Vedi ambiente.](#)

training

Fornire dati da cui trarre ispirazione dal modello di machine learning. I dati di training devono contenere la risposta corretta. L'algoritmo di apprendimento trova nei dati di addestramento i pattern che mappano gli attributi dei dati di input al target (la risposta che si desidera prevedere). Produce un modello di ML che acquisisce questi modelli. Puoi quindi utilizzare il modello di ML per creare previsioni su nuovi dati di cui non si conosce il target.

Transit Gateway

Un hub di transito di rete che puoi utilizzare per interconnettere le tue reti VPCs e quelle locali. Per ulteriori informazioni, consulta [Cos'è un gateway di transito](#) nella AWS Transit Gateway documentazione.

flusso di lavoro basato su trunk

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

Accesso attendibile

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

regolazione

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

team da due pizze

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

U

incertezza

Un concetto che si riferisce a informazioni imprecise, incomplete o sconosciute che possono minare l'affidabilità dei modelli di machine learning predittivi. Esistono due tipi di incertezza: l'incertezza epistemica, che è causata da dati limitati e incompleti, mentre l'incertezza aleatoria è causata dal rumore e dalla casualità insiti nei dati. Per ulteriori informazioni, consulta la guida [Quantificazione dell'incertezza nei sistemi di deep learning](#).

compiti indifferenziati

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

ambienti superiori

[Vedi ambiente.](#)

V

vacuum

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

controllo delle versioni

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

Peering VPC

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

vulnerabilità

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

W

cache calda

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

dati caldi

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

funzione finestra

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

Carico di lavoro

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

flusso di lavoro

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

VERME

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

WQF

Vedi [AWS Workload Qualification Framework](#).

scrivi una volta, leggi molte (WORM)

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

Z

exploit zero-day

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

vulnerabilità zero-day

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

prompt zero-shot

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

applicazione zombie

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

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.