



Ripiattoforma delle applicazioni mainframe utilizzando un database IBM Db2
for z/OS condiviso per una migrazione graduale

AWS Guida prescrittiva



AWS Guida prescrittiva: Ripiattaforma delle applicazioni mainframe utilizzando un database IBM Db2 for z/OS condiviso per una migrazione graduale

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
Risultati aziendali	2
Modernizzazione del mainframe AWS	4
Ridefinizione della piattaforma	4
Rifattorizzazione automatizzata	5
Vantaggi del replatforming	5
Il processo di trasformazione	7
Pianificazione	8
Scoperta delle applicazioni	8
Dipendenze tra i dati	8
Benchmark sulla capacità	9
Pianificazione delle onde	11
In fase di creazione	12
Coerenza delle applicazioni	12
Architecture	13
In esecuzione	15
Impegno in due fasi (2 PC)	16
Infrastruttura di runtime	16
Test in corso	18
Ambiente di origine	18
Ambiente di destinazione	19
Analisi	19
Test dell'applicazione in Modernizzazione del mainframe AWS	20
Conversione	22
Fornitura	22
Trasmetti in diretta	22
Rollback	23
Concludi	23
Architecture	23
Best practice	25
Latenza di rete	25
Sicurezza	26
Governance delle applicazioni	26
Elasticità	27

Fasi successive	28
Resources	29
AWS documentazione	29
Riferimenti a Rocket Software	29
Riferimenti IBM	29
Tools (Strumenti)	29
AWS Modelli e guide di orientamento prescrittivo	29
Cronologia dei documenti	31
Glossario	32
#	32
A	33
B	36
C	38
D	41
E	45
F	47
G	49
H	50
I	52
L	54
M	55
O	60
P	62
Q	65
R	66
S	69
T	73
U	74
V	75
W	75
Z	77
.....	lxxviii

Ripiattaforma delle applicazioni mainframe utilizzando un IBM Db2 for database condiviso per una migrazione graduale z/OS

Luis Gustavo Dantas e André Botura, Amazon Web Services (AWS)

Maggio 2025 ([storia del documento](#))

Nel panorama in continua evoluzione della tecnologia aziendale, la modernizzazione del mainframe è diventata un requisito fondamentale per le organizzazioni che devono rimanere competitive e agili. Questa trasformazione non riguarda semplicemente la sostituzione dei vecchi sistemi con altri nuovi; è un'evoluzione strategica che colma il divario tra le solide e affidabili fondamenta del passato e le possibilità dinamiche e innovative del futuro.

Il mainframe, che un tempo era il leader indiscusso dell'informatica aziendale, è ora a un punto di svolta. La sua impareggiabile potenza di elaborazione e le sue funzionalità di sicurezza lo hanno reso importante per decenni, ma le aziende di oggi richiedono sistemi in grado di integrarsi perfettamente con i servizi cloud, supportare le applicazioni mobili e sfruttare la potenza dell'intelligenza artificiale e dell'analisi dei big data.

La modernizzazione non richiede sempre una migrazione completa dai mainframe. Alcune organizzazioni stanno optando per approcci ibridi che sfruttano i punti di forza degli ambienti mainframe e cloud. Questa strategia consente loro di mantenere le applicazioni legacy critiche mentre passano gradualmente a piattaforme più moderne. Questa transizione tecnologica non comporta solo aggiornamenti di sistema; richiede la trasformazione della cultura e delle competenze organizzative. Man mano che le aziende si modernizzano, investono sia in nuove tecnologie che nella forza lavoro, colmando i divari generazionali e promuovendo l'apprendimento e l'innovazione continui.

Questa guida illustra una strategia di migrazione graduale che bilancia i vantaggi dei sistemi mainframe con i vantaggi delle moderne tecnologie cloud. Questo approccio di ripiattaforma graduale migra innanzitutto il livello applicativo, mantenendo al contempo la connettività al database IBM Db2 for z/OS database esistente per semplificare il processo di transizione e ridurre al minimo le interruzioni delle operazioni aziendali critiche mentre si adottano nuove funzionalità nel cloud.

Questa guida è progettata per i responsabili delle decisioni tecniche e i team di implementazione coinvolti in iniziative di modernizzazione del mainframe. I destinatari principali includono

architetti aziendali e di soluzioni, responsabili di progetto tecnici e responsabili dei programmi di modernizzazione, che devono comprendere sia gli aspetti strategici che quelli tecnici della ripiattoforma del mainframe. Il contenuto è altrettanto utile per i team di implementazione, inclusi gli sviluppatori di applicazioni mainframe AWS o gli ingegneri del cloud, gli amministratori di database e DevOps gli ingegneri, responsabili dell'esecuzione dell'implementazione della modernizzazione.

Risultati aziendali

Le aziende hanno molte valide ragioni per aggiornare le proprie applicazioni legacy. Questo processo crea un senso di urgenza in tutti i settori. Quando gli esperti più anziani vanno in pensione, lasciano un notevole divario di conoscenze, il che rende fondamentale modernizzare i sistemi prima che queste competenze vadano perdute. Inoltre, le aziende sono guidate dalla necessità di ridurre i costi, aumentare l'agilità e rispondere rapidamente alle condizioni di mercato in rapida evoluzione.

La spinta verso la trasformazione digitale è ulteriormente intensificata dalle tecnologie emergenti e dalla richiesta di esperienze migliorate per i clienti. Questi fattori, combinati con i rischi associati alla manutenzione di sistemi complessi, stanno spingendo le organizzazioni ad agire rapidamente per modernizzare la propria infrastruttura IT.

La modernizzazione del mainframe, in particolare, rappresenta un delicato equilibrio. Le aziende devono preservare la stabilità e la sicurezza per cui sono noti i mainframe, abbracciando al contempo la flessibilità e la scalabilità offerte dalle architetture moderne. Questo processo implica decisioni complesse su quali applicazioni migrare, quali riscrivere e quali conservare sul mainframe.

I fattori chiave per la modernizzazione includono l'agilità e la riduzione dei costi:

- **Agilità e time-to-market.** I sistemi moderni consentono processi di approvvigionamento più rapidi e risposte più rapide alle mutevoli richieste del mercato. L'adozione di SysOps pratiche DevOps e procedure può migliorare significativamente la produttività e la velocità di implementazione.
- **Riduzione dei costi.** La modernizzazione spesso porta a una riduzione dei costi dell'infrastruttura attraverso:
 - Pay-as-you-go modelli, che allineano i costi all'utilizzo effettivo.
 - Costi di licenza ridotti associati ai sistemi legacy.
 - Maggiore elasticità, che offre una migliore allocazione delle risorse.
 - Configurazioni attive-attive e ad alta disponibilità, che migliorano la resilienza del sistema ottimizzando al contempo l'utilizzo delle risorse.

Sulla base di questi fattori di business, la ripiattaforma delle applicazioni COBOL è considerata un approccio strategico alla modernizzazione. È possibile utilizzare un database condiviso per seguire un percorso di migrazione graduale che bilanci la necessità di modernizzazione con l'imperativo di mantenere la continuità aziendale. Questo metodo consente di sfruttare i vantaggi delle architetture moderne preservando al contempo l'affidabilità delle applicazioni COBOL. Di conseguenza, è possibile ottenere agilità, efficienza dei costi e innovazione mitigando al contempo i rischi associati a transizioni brusche e su larga scala. L'approccio condiviso al database Db2 descritto in questa guida fornisce un ponte tra i sistemi legacy e le piattaforme moderne e consente un processo di modernizzazione più fluido e controllato.

In questa guida:

- [Modernizzazione del mainframe AWS](#)
- [Il processo di trasformazione](#)
- [Best practice](#)
- [Fasi successive](#)
- [Risorse](#)
- [Storia del documento](#)

Modernizzazione del mainframe AWS

Note

Modernizzazione del mainframe AWS Il servizio (Managed Runtime Environment experience) non è più aperto a nuovi clienti. Per funzionalità simili a Modernizzazione del mainframe AWS Service (Managed Runtime Environment experience), esplora Modernizzazione del mainframe AWS Service (Self-Managed Experience). I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta [Modifica della Modernizzazione del mainframe AWS disponibilità](#).

Il [Modernizzazione del mainframe AWS servizio](#) consente di migrare le applicazioni mainframe legacy in un ambiente nativo del cloud, preservare la logica e gli investimenti aziendali esistenti, utilizzare strumenti automatizzati e servizi di runtime gestiti, ottimizzare le prestazioni delle applicazioni e ridurre i costi operativi. Questo servizio semplifica il processo di modernizzazione, in modo da poter sfruttare la potenza del cloud mantenendo il valore dei sistemi mainframe principali. AWS offre due approcci chiave alla modernizzazione del mainframe: il replatforming e il refactoring automatizzato.

Ridefinizione della piattaforma

[Modernizzazione del mainframe AWS Replatform with Rocket Software](#) (precedentemente Micro Focus) offre una potente opzione di ripiattoforma per le aziende che desiderano migrare le proprie applicazioni mainframe sul cloud con interruzioni minime. Questa soluzione consente di ricompilare ed eseguire il COBOL e le applicazioni esistenti senza richiedere modifiche significative al codice. PL/ I AWS

I principali vantaggi della soluzione AWS Replatform with Rocket Software includono:

- Conservazione della logica aziendale e degli investimenti esistenti
- Rischi ridotti e tempi di commercializzazione più rapidi
- Scalabilità e prestazioni migliorate sull'infrastruttura AWS
- Accesso a strumenti e pratiche di sviluppo moderni

È possibile utilizzare questa soluzione per mantenere i linguaggi di programmazione mainframe familiari, sfruttando al contempo la flessibilità, l'economicità e l'innovazione di Cloud AWS

Rifattorizzazione automatizzata

Per un approccio più trasformativo rispetto al replatforming, puoi utilizzare [AWS Blu Age](#), che offre il refactoring automatizzato delle applicazioni mainframe verso applicazioni native del cloud basate su Java. Questa soluzione consente di modernizzare i sistemi legacy in modo più completo e convertirli in applicazioni in grado di sfruttare appieno le tecnologie native del cloud.

I principali vantaggi di Blu Age includono: AWS

- Conversione di codice legacy in applicazioni Java moderne e gestibili
- Trasformazione automatizzata che riduce lo sforzo manuale e i potenziali errori
- Creazione di applicazioni native per il cloud ottimizzate per Servizi AWS
- Agilità migliorata e integrazione più semplice con le tecnologie moderne

AWS Blu Age ti aiuta a migrare le tue applicazioni e a prepararle per il cloud, per aprire nuove possibilità di innovazione e crescita. Per ulteriori informazioni su questo approccio, consulta [Rifattorizzazione automatica delle applicazioni con AWS Blu Age nella documentazione](#).

Modernizzazione del mainframe AWS

Vantaggi del replatforming

Questa guida illustra un approccio per la ripiattoforma delle applicazioni COBOL mainframe su AWS. Questo approccio mira a modernizzare i sistemi legacy mantenendo temporaneamente IBM Db2 per semplificare il processo di transizione. z/OS Mantenendo inizialmente la struttura del database esistente, è possibile ridurre la complessità e i rischi durante la migrazione. Questo approccio graduale consente di trarre vantaggio dalla scalabilità e dalla convenienza economica, Cloud AWS preservando al contempo l'integrità dei dati critici. I vantaggi della ripiattoforma graduale includono quanto segue:

- **Modernizzazione accelerata:** il replatforming e il refactoring in genere richiedono meno tempo e risorse rispetto alla riprogettazione di un'applicazione legacy nel cloud, perché non comportano la riscrittura dell'intera applicazione. Questo approccio supporta anche una transizione più graduale che consente alle organizzazioni di modernizzarsi al proprio ritmo, beneficiando immediatamente della scalabilità e della convenienza di Cloud AWS.

- **Mitigazione del rischio:** Replatforming offre diversi vantaggi rispetto al refactoring per molte organizzazioni. Le aziende possono mantenere il COBOL e le PL/I codebase esistenti, preservare anni di logica aziendale e ridurre al minimo il rischio associato a modifiche al codice su larga scala.
- **Continuità dei dati e migrazione graduale:** un vantaggio significativo del replatforming è la possibilità di conservare inizialmente i dati in Db2 o nel formato di dati originale. z/OS Questa strategia evita la necessità di processi di migrazione dei dati immediati, complessi e potenzialmente rischiosi. Mantenendo i dati nell'ambiente originale durante la fase iniziale, è possibile preservarne l'integrità, ridurre i tempi di inattività e minimizzare il rischio di perdita o danneggiamento dei dati durante il processo di modernizzazione. In un secondo momento, puoi pianificare una migrazione dei dati controllata e graduale verso database nativi del cloud che prevede test e convalide approfonditi mentre l'applicazione continua a funzionare nell'ambiente riplatformato.
- **Flessibilità e preparazione al futuro:** per le aziende che hanno investimenti significativi in competenze e applicazioni mainframe, il replatforming offre un percorso pragmatico verso la modernizzazione che bilancia innovazione e continuità. Offre la flessibilità necessaria per conservare inizialmente le strutture di dati e i metodi di accesso critici, ponendo al contempo le basi per le future iniziative di modernizzazione, inclusa l'eventuale migrazione dei dati verso soluzioni completamente native del cloud.

Le organizzazioni possono seguire l'approccio replatforming per modernizzarsi al proprio ritmo e soddisfare le esigenze immediate, pianificando al contempo obiettivi di trasformazione digitale a lungo termine. Questo approccio offre inoltre alle aziende l'opportunità di formare il proprio personale sui servizi nativi del cloud.

Il processo di trasformazione

La modernizzazione del mainframe è un passaggio fondamentale per le organizzazioni che vogliono sfruttare i vantaggi del cloud computing preservando al contempo le loro preziose applicazioni legacy. Questa trasformazione presenta sfide significative. Le applicazioni mainframe sono in genere altamente accoppiate e presentano interdipendenze complesse che si sono evolute nel corso di decenni di attività. Questa complessità richiede un approccio attento e metodico alla modernizzazione.

Le organizzazioni devono affrontare le seguenti fasi chiave per una transizione di successo:

- **Pianificazione:** questa fase prevede una scoperta completa dei sistemi esistenti e la definizione delle priorità degli sforzi di modernizzazione. Organizations valuta la propria infrastruttura attuale, identifica le applicazioni critiche e determina quali sistemi devono essere modernizzati per primi.
- **Costruzione:** durante questa fase, le organizzazioni creano processi per migrare le applicazioni e sviluppare nuovi sistemi e infrastrutture. Ciò comporta la progettazione e l'implementazione dell'architettura modernizzata e la compilazione del codice sorgente.
- **Esecuzione:** questo passaggio consiste nella creazione degli ambienti di runtime per ospitare le applicazioni riplatformate. Implica la configurazione dell'hardware, del software e dell'infrastruttura cloud necessari per supportare i sistemi modernizzati e garantire che possano funzionare in modo efficiente nel nuovo ambiente.
- **Test:** questa fase include una rigorosa convalida dei sistemi modernizzati per verificare che tutti i requisiti funzionali e prestazionali siano stati soddisfatti. Vengono condotti test approfonditi per verificare l'integrità dei dati, la compatibilità del sistema e le prestazioni complessive del nuovo ambiente.
- **Cutover:** la fase finale si concentra sull'implementazione di strategie per una transizione fluida e sul controllo del passaggio dal mainframe legacy all'ambiente modernizzato. Ciò include un'attenta pianificazione del programma di migrazione e dei piani di emergenza per ridurre al minimo le interruzioni delle operazioni aziendali.

Le seguenti sezioni illustrano queste fasi in dettaglio:

- [Pianificazione](#)
- [Edificio](#)
- [In esecuzione](#)

- [Test in corso](#)
- [Cutover](#)

Pianificazione

Per soddisfare in modo efficace i requisiti delle applicazioni mainframe legacy, le organizzazioni spesso iniziano con una valutazione completa del proprio ambiente mainframe.

Scoperta delle applicazioni

Uno strumento potente in questa fase iniziale è [Rocket Enterprise Analyzer](#), che fornisce informazioni approfondite sulla struttura, le dipendenze e la complessità delle applicazioni mainframe. Questo strumento consente di determinare l'ambito delle attività di modernizzazione, i potenziali rischi e le opportunità di ottimizzazione.

Un aspetto cruciale da scoprire è l'intricata rete di dipendenze dei dati all'interno dei sistemi mainframe. Queste dipendenze sono spesso nascoste sotto strati di codice legacy e possono influire in modo significativo sugli sforzi di modernizzazione. Mappando il modo in cui diverse applicazioni e moduli interagiscono con varie fonti di dati, puoi comprendere meglio i potenziali effetti di qualsiasi modifica che intendi implementare.

Dipendenze tra i dati

Una valutazione approfondita delle dipendenze dei dati può rivelare informazioni critiche sul flusso di dati, sulla qualità dei dati e sulla governance dei dati all'interno dell'ambiente mainframe. Questa conoscenza è preziosa per pianificare strategie di migrazione dei dati, garantire l'integrità dei dati durante la modernizzazione e identificare opportunità di ottimizzazione dei dati. Acquisendo un quadro chiaro dei dati, è possibile prendere decisioni più informate su quali approcci di modernizzazione saranno più efficaci e meno dannosi per le operazioni esistenti.

Un'analisi dall'alto verso il basso che identifichi l'utilizzo delle tabelle in base alle transazioni o ai job Job Control Language (JCL) è fondamentale per creare una pianificazione e una definizione delle priorità a ondate. Questo approccio chiarisce le relazioni tra i diversi componenti dei sistemi mainframe e consente di sviluppare un approccio strategico e graduale alla modernizzazione. Identificando a quali tabelle si accede più frequentemente e con quali processi, è possibile dare priorità agli sforzi di modernizzazione: è possibile concentrarsi innanzitutto sulle aree ad alto impatto e garantire una transizione più fluida con interruzioni minime delle operazioni aziendali critiche.

Oltre a utilizzare Rocket Enterprise Analyzer per scoprire le dipendenze tra i dati, molte organizzazioni utilizzano anche soluzioni personalizzate per ottenere informazioni più approfondite sui propri ambienti mainframe. Questi strumenti interni spesso sfruttano la ricchezza di informazioni disponibili nel catalogo IBM Db2 e nei record di System Management Facility (SMF).

Benchmark sulla capacità

Una fase della pianificazione del progetto di ripiattoforma del mainframe consiste nel raccogliere informazioni dettagliate sull'attuale consumo di carichi di lavoro. Questi dati vi aiuteranno a prevedere e fornire con precisione la capacità iniziale richiesta nell'ambiente cloud di destinazione. Ad esempio, ti consigliamo di raccogliere milioni di dati sul consumo orario di istruzioni al secondo (MIPS) sia per le transazioni online che per le transazioni in batch dai job IBM Customer Information Control System (CICS) o Information Management System (IMS) e Job Control Language (JCL).

IBM offre una vasta gamma di [modelli di prezzo](#) per MIPS nell'elaborazione mainframe e molti di questi modelli sono incentrati sull'utilizzo di picco. Tra questi modelli basati sui picchi, il più comune è il modello di picco continuo di quattro ore.

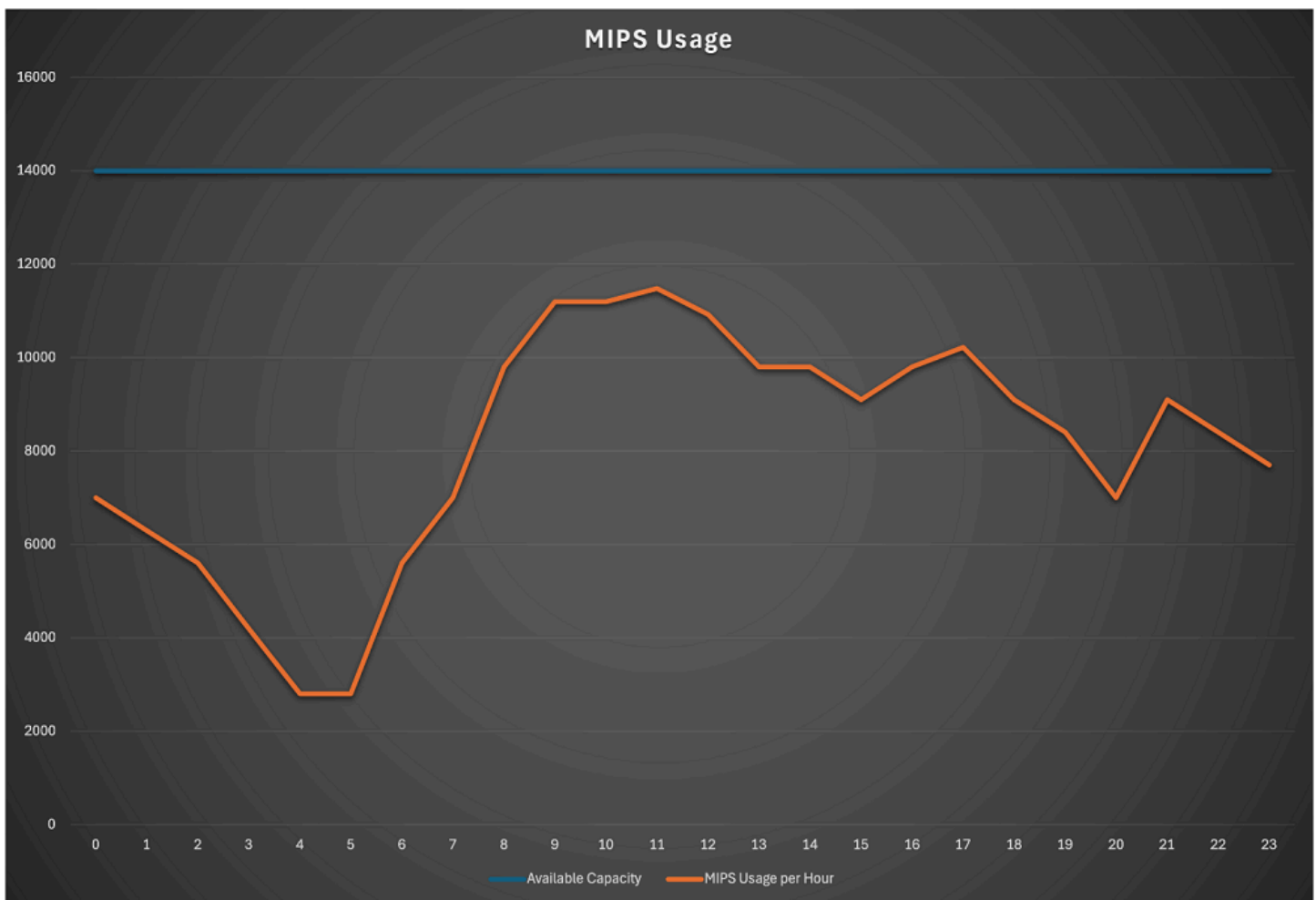
I costi del mainframe comprendono cinque aree chiave che influiscono in modo significativo sulle spese complessive:

- Le licenze software sono spesso un componente importante. Copre sistemi operativi, middleware, database e varie applicazioni e i costi sono talvolta legati alla capacità o all'utilizzo delle macchine.
- Le spese hardware comprendono l'acquisto o il leasing iniziale di apparecchiature mainframe, la manutenzione continua e gli aggiornamenti.
- I costi di storage possono essere notevoli a causa delle enormi quantità di dati gestiti e coinvolgono sistemi a disco, librerie a nastro e il software di gestione associato.
- Le spese per il personale coprono gli stipendi di professionisti specializzati nel settore mainframe, come programmatori di sistema e amministratori di database.
- Le misure di disaster recovery e di continuità aziendale, tra cui sistemi di backup, hardware ridondante e strutture di ripristino offsite, rappresentano un investimento significativo per garantire un'elevata disponibilità e un ripristino rapido.

Queste cinque categorie di costi, insieme ai costi basati su MIPS, costituiscono la base della maggior parte dei budget per i mainframe. Tuttavia, le loro proporzioni relative possono variare notevolmente a seconda delle dimensioni dell'organizzazione, del settore e degli specifici modelli di utilizzo del mainframe.

I dati MIPS orari sono fondamentali per acquisire una comprensione completa dei modelli di carico di lavoro e delle prestazioni del mainframe. A differenza delle medie giornaliere o mensili, i dati orari forniscono informazioni granulari che rivelano le variegiate fluttuazioni dell'utilizzo delle risorse del sistema nel corso della giornata. Questo livello di dettaglio è prezioso per valutare con precisione le esigenze di prestazioni e capacità dell'applicazione nel cloud.

Analizzando i dati MIPS orari, è possibile identificare i periodi di picco di utilizzo, individuare le tendenze e individuare potenziali rallentamenti che potrebbero essere oscurati nei dati aggregati, come mostrato nel diagramma seguente. Questa granularità consente una pianificazione più precisa della capacità, aiuta a ottimizzare l'allocazione delle risorse e può potenzialmente portare a risparmi sui costi e a una migliore efficienza del sistema.



I dati MIPS orari fungono anche da strumento essenziale di benchmark delle prestazioni. Stabilisce una base dettagliata delle prestazioni del sistema, che è particolarmente utile quando si pianificano o si valutano modifiche al sistema come migrazioni o aggiornamenti. Confrontando i dati MIPS orari precedenti e successivi alla modifica, è possibile misurare con precisione l'impatto di queste

modifiche sulle prestazioni del sistema e garantire che il mainframe continui a soddisfare le esigenze dell'organizzazione.

Per raccogliere dati MIPS su base oraria, sono disponibili diverse opzioni. Un approccio consiste nell'utilizzare direttamente i record SMF. Questi record forniscono una grande quantità di informazioni sull'attività del sistema e sull'utilizzo delle risorse. In alternativa, è possibile utilizzare strumenti specializzati come IBM Sub-Capacity Reporting Tool (SCRT), che può semplificare il processo di raccolta e analisi dei dati MIPS.

Indipendentemente dal metodo scelto, è importante raccogliere i dati per un periodo prolungato, idealmente per diversi mesi. Questo periodo di raccolta prolungato consente di tenere conto delle variazioni cicliche del carico di lavoro, come picchi di end-of-month elaborazione o fluttuazioni stagionali. Acquisendo questi modelli a lungo termine, è possibile sviluppare un quadro più accurato e completo delle caratteristiche prestazionali del mainframe, che consente un processo decisionale più informato e una gestione della capacità più efficace.

Pianificazione delle onde

È possibile utilizzare le informazioni raccolte per assegnare priorità strategiche alle iniziative di ripiattoforma del mainframe. Un approccio prudente consiste nell'iniziare con carichi di lavoro meno critici, come transazioni commerciali non essenziali o lavori in batch, per consentire ai team di acquisire esperienza e perfezionare i processi con il minimo rischio per le operazioni essenziali. Inoltre, considerare i carichi di lavoro di sola lettura come primi candidati per la migrazione può essere vantaggioso, poiché questi carichi di lavoro in genere comportano meno complessità e un minor rischio di incoerenze tra i dati. Questo approccio consente di creare fiducia e slancio nelle attività di ripiattoforma.

Inoltre, il raggruppamento di carichi di lavoro che condividono tabelle Db2 per operazioni di scrittura o aggiornamento può semplificare il processo di migrazione. Identificando questi carichi di lavoro interconnessi, è possibile pianificare ondate di migrazione coerenti che mantengano l'integrità dei dati e riducano al minimo la necessità di soluzioni provvisorie complesse. Questa strategia non solo riduce il rischio di conflitti di dati, ma ottimizza anche la tempistica complessiva della ripiattoforma affrontando contemporaneamente i componenti correlati. In definitiva, questo approccio di prioritizzazione basato sui dati garantisce una considerazione equilibrata di criticità, complessità e interdipendenza e porta a un processo di modernizzazione del mainframe più efficiente e di successo.

In fase di creazione

L'utilizzo di un database Db2 condiviso consente l'esecuzione simultanea di applicazioni identiche o coerenti in ambienti mainframe e cloud. Questo approccio offre diversi vantaggi quando si mantiene la stessa versione dell'applicazione su entrambe le piattaforme e offre maggiore flessibilità e affidabilità nelle operazioni.

Uno dei principali vantaggi di questa strategia è la capacità di implementare un piano di rollback efficace. In caso di problemi durante la migrazione o l'implementazione, la disponibilità della stessa versione dell'applicazione consente un rapido ritorno allo stato precedente e riduce al minimo i tempi di inattività e le potenziali incongruenze nei dati.

Coerenza delle applicazioni

Il mirroring dei componenti delle applicazioni da un gestore di controllo del codice sorgente distribuito al mainframe è un approccio strategico durante il processo di ripiattaforma. Questo metodo supporta l'uso di moderni strumenti di gestione del codice sorgente mantenendo al contempo la sincronizzazione con l'ambiente mainframe. Questo processo di mirroring è temporaneo e dura solo fino a quando il carico di lavoro non è completamente funzionale nella produzione sulla piattaforma distribuita.

Migrando il codice sorgente dell'applicazione replatformata verso uno strumento distribuito di gestione delle modifiche, è possibile sfruttare i numerosi vantaggi offerti dai moderni gestori di codice sorgente. Ciò include:

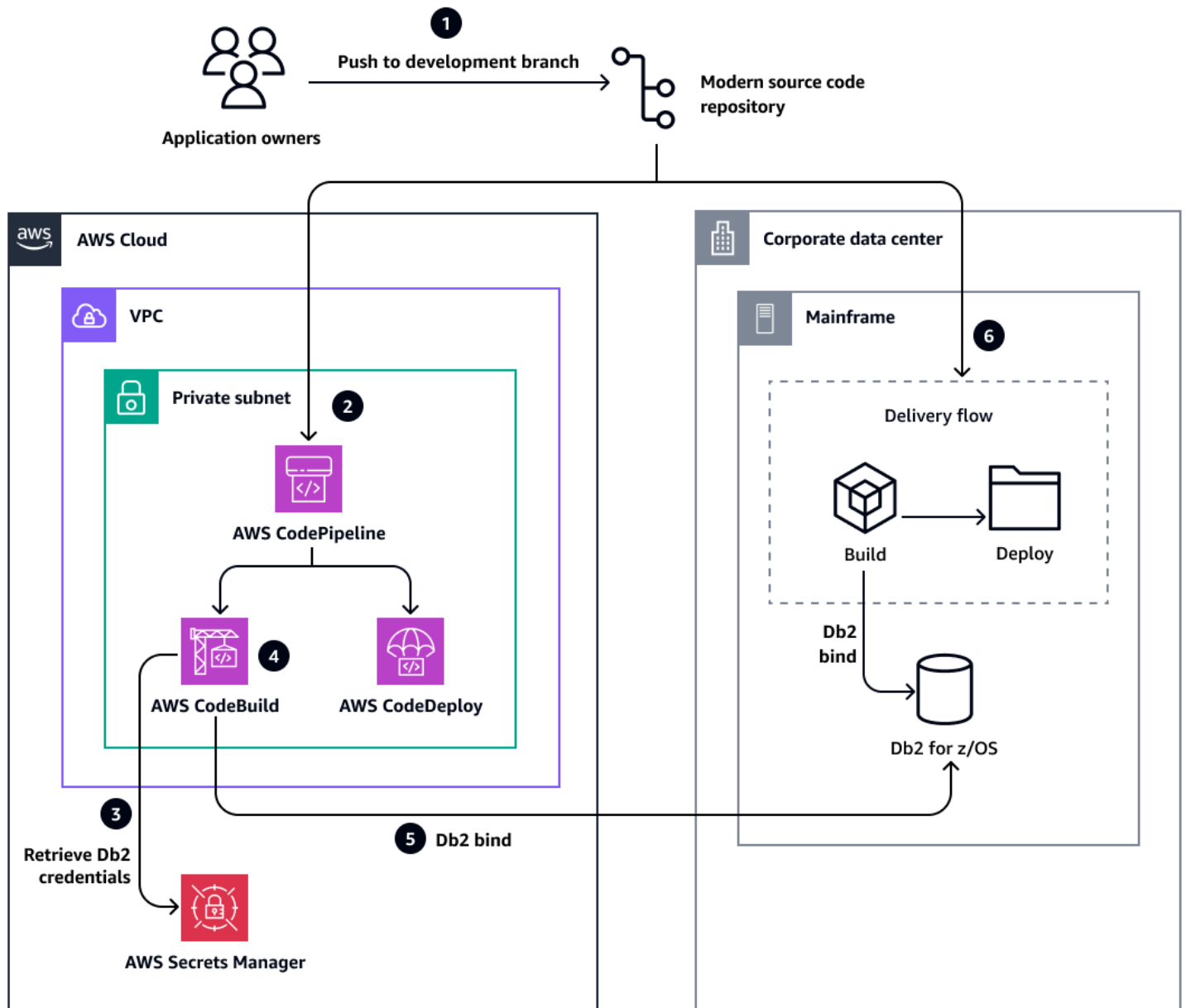
- **Collaborazione migliorata:** gli strumenti distribuiti spesso forniscono un supporto migliore per la collaborazione in team, includendo funzionalità come pull request, revisioni del codice e strategie di ramificazione.
- **Controllo delle versioni migliorato:** i sistemi moderni offrono un controllo delle versioni più granulare e semplificano il monitoraggio delle modifiche e la gestione delle diverse versioni del codice.
- **Integrazione con CI/CD le pipeline:** molti strumenti distribuiti si integrano perfettamente con le pipeline di integrazione e distribuzione continua (CI/CD), che semplificano il processo di sviluppo.
- **Migliore visibilità e tracciabilità:** questi strumenti spesso forniscono dashboard e funzionalità di reporting superiori e offrono una visione più approfondita del processo di sviluppo.
- **Support per pratiche di sviluppo moderne:** i sistemi distribuiti sono in genere più adatti per metodologie e DevOps pratiche agili.

Il processo di mirroring prevede la sincronizzazione del codice dal gestore di controllo del codice sorgente distribuito al mainframe. Ciò garantisce che entrambi gli ambienti rimangano coerenti durante il periodo di transizione. Tuttavia, è necessario implementare il mirroring come sincronizzazione unidirezionale, in cui gli aggiornamenti fluiscono dal sistema distribuito al mainframe, anziché bidirezionalmente. Questo approccio mantiene la coerenza e previene i potenziali conflitti che potrebbero derivare dagli aggiornamenti simultanei in entrambi gli ambienti.

Adottando questa strategia di mirroring, è possibile spostare gradualmente le attività di sviluppo verso la piattaforma distribuita, garantendo al contempo il mantenimento dell'ambiente mainframe. up-to-date Ciò garantisce una transizione più fluida e una rete di sicurezza durante il processo di ripiattaforma. Quando il carico di lavoro è completamente funzionale e stabile nell'ambiente di produzione distribuito, è possibile eliminare gradualmente il processo di mirroring e completare la migrazione al moderno sistema di gestione del codice sorgente.

Architecture

Il diagramma seguente mostra come un sistema di gestione del codice sorgente distribuito può rispecchiare i componenti delle applicazioni e mantenere la sincronizzazione tra l' Cloud AWS ambiente mainframe e quello mainframe. L' Cloud AWS ambiente utilizza CI/CD servizi come [AWS CodeBuild](#), [AWS CodePipeline](#), e [AWS CodeDeploy](#) per creare e distribuire l'applicazione.



In questo flusso di lavoro:

1. I proprietari dell'applicazione distribuiscono una nuova versione dell'applicazione nel ramo di sviluppo del repository del codice sorgente.
2. I trigger AWS CodePipeline della nuova versione.
3. AWS CodeBuild recupera le credenziali Db2 da. [Gestione dei segreti AWS](#)
4. CodeBuild compila l'applicazione.
5. CodeBuild usa Db2 z/OS per associare l'applicazione.
6. Il flusso di distribuzione del mainframe crea e distribuisce anche l'applicazione.

In esecuzione

Per garantire prestazioni ottimali e una bassa latenza tra l'applicazione basata sul cloud e il database locale, si consiglia di implementare [AWS Direct Connect](#). Questo servizio fornisce una connessione di rete dedicata tra AWS e il data center dell'organizzazione e offre prestazioni di rete più costanti e una latenza ridotta rispetto alle connessioni basate su Internet. Ciò è particolarmente importante per le operazioni di database che richiedono tempi di risposta rapidi.

Per ottenere un'elevata disponibilità (HA) ed elasticità per l'applicazione su cui è in esecuzione AWS, è possibile implementare un'architettura robusta utilizzando i seguenti componenti:

- **Elastic Load Balancing (ELB):** puoi implementare un sistema di bilanciamento del carico per distribuire il traffico in entrata su più istanze Amazon Elastic Compute Cloud (Amazon EC2) su cui viene eseguita l'applicazione. Ciò garantisce una distribuzione uniforme del carico di lavoro e fornisce un unico punto di ingresso per le richieste dei clienti.
- **Gruppo Auto Scaling:** le istanze EC2 che ospitano l'applicazione possono essere organizzate in un gruppo Auto Scaling. Ciò consente all'infrastruttura di regolare automaticamente il numero di istanze in base a metriche predefinite come l'utilizzo della CPU o il traffico di rete. Durante le ore di punta, è possibile avviare istanze aggiuntive per gestire un carico maggiore, mentre durante i periodi più silenziosi, è possibile chiudere le istanze non necessarie per ottimizzare i costi.
- **Istanze EC2:** l'applicazione può essere distribuita su istanze EC2 all'interno del gruppo Auto Scaling. Queste istanze devono essere distribuite su più zone di disponibilità per migliorare la tolleranza agli errori e garantire un'elevata disponibilità.
- **Implementazione Multi-AZ:** distribuendo le istanze dell'applicazione su più zone di disponibilità, il sistema è in grado di resistere al guasto di una singola zona di disponibilità senza un impatto significativo sulla disponibilità complessiva.

Questa architettura consente all'applicazione di scalare senza problemi in base alla domanda, mantenendo al contempo un'elevata disponibilità. Il sistema di bilanciamento del carico assicura che il traffico sia distribuito in modo uniforme tra le istanze integre e il gruppo Auto Scaling gestisce il numero di istanze in base al carico di lavoro effettivo.

Per migliorare ulteriormente l'affidabilità, puoi implementare un robusto sistema di monitoraggio e avviso utilizzando [Amazon CloudWatch](#) per rilevare e rispondere tempestivamente a eventuali problemi o guasti di prestazioni. Inoltre, test regolari delle funzionalità di scalabilità automatica e degli

scenari di failover garantiranno che il sistema si comporti come previsto in varie condizioni di carico e potenziali guasti.

Adottando questo approccio, è possibile trarre vantaggio dalla scalabilità e dalla flessibilità del mantenimento al Cloud AWS contempo di una connessione sicura al database Db2 locale. Questa configurazione ibrida rappresenta un percorso eccellente verso una migrazione completa al cloud e fornisce una transizione graduale e una mitigazione del rischio durante tutto il processo.

Impegno in due fasi (2 PC)

[Modernizzazione del mainframe AWS Replatform with Rocket Software](#) offre supporto per transazioni di commit in due fasi (2PC) attraverso l'implementazione dell'architettura estesa (XA). Questa funzionalità è fondamentale per mantenere l'integrità dei dati nei sistemi distribuiti, in particolare negli ambienti mainframe in cui le transazioni complesse spesso si estendono su più risorse.

L'architettura XA, integrata in AWS Replatform with Rocket Software, consente il coordinamento delle transazioni tra diverse risorse come database e code di messaggi. Questa integrazione garantisce che tutte le parti di una transazione distribuita vengano confermate o ripristinate all'unisono, per mantenere la coerenza in tutto il sistema.

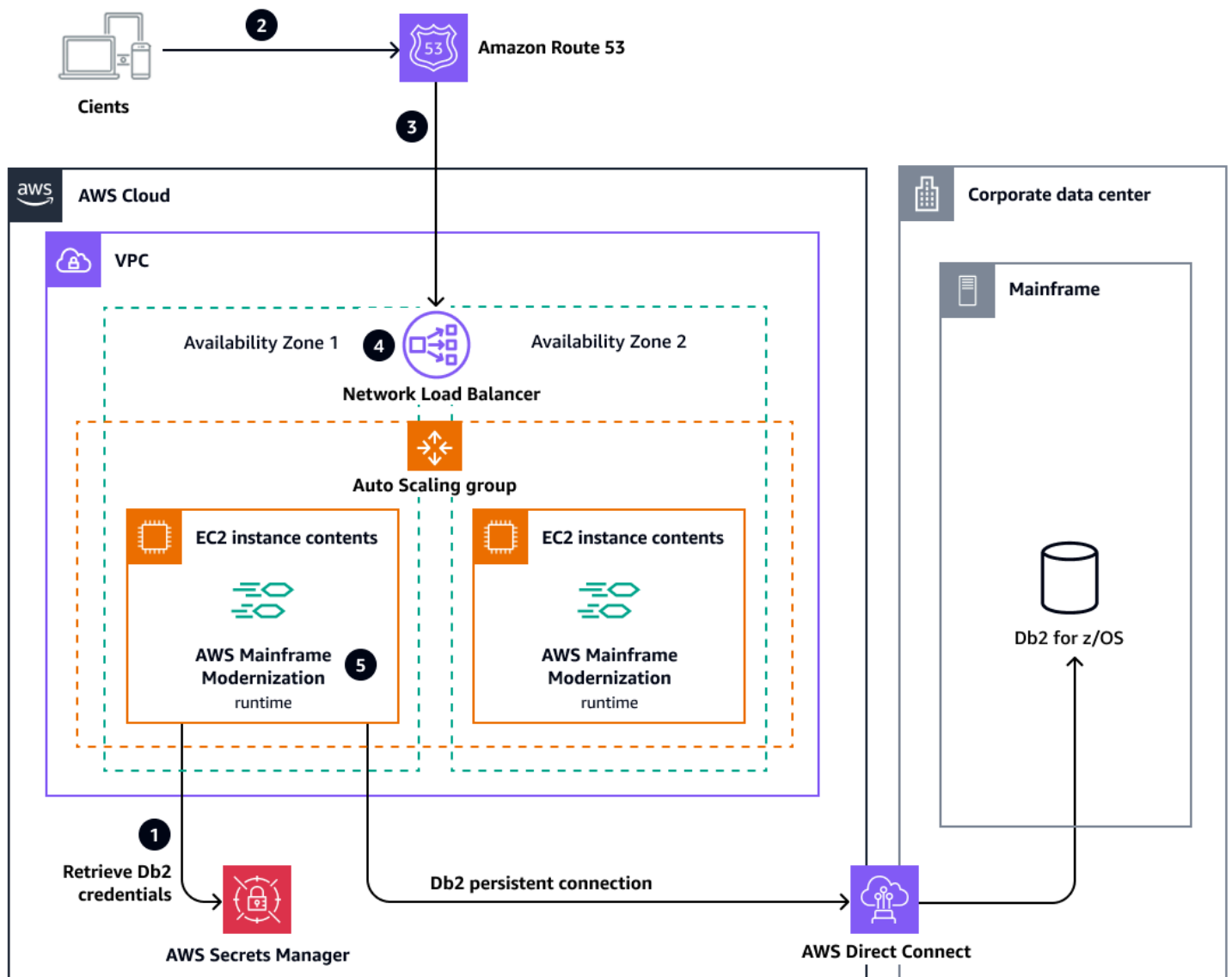
Il processo 2PC consiste in due fasi:

- Fase di preparazione: il gestore delle transazioni interroga tutti i gestori delle risorse coinvolti nella transazione per assicurarsi che siano pronti a impegnarsi.
- Fase di impegno: se tutti i gestori delle risorse rispondono positivamente, il gestore delle transazioni ordina loro di eseguire le modifiche. Se uno dei gestori delle risorse non può impegnarsi, a tutti i manager viene chiesto di annullare le modifiche.

Utilizzando XA, AWS Replatform with Rocket Software fornisce una soluzione affidabile e scalabile per la gestione di transazioni complesse e distribuite in ambienti mainframe modernizzati. Questa funzionalità è essenziale per le organizzazioni che desiderano migrare le proprie applicazioni mainframe sul cloud senza compromettere l'integrità o le prestazioni transazionali.

Infrastruttura di runtime

Il diagramma seguente mostra un ambiente ad alta disponibilità ed elastico Cloud AWS che include due zone di disponibilità, istanze EC2 in un gruppo Auto Scaling, un Network Load Balancer e una connessione dedicata tra gli ambienti mainframe e tramite AWS Direct Connect.



In questa architettura:

1. All'avvio del Modernizzazione del mainframe AWS runtime, recupera le credenziali Db2 [Gestione dei segreti AWS](#) e apre una connessione persistente con Db2 for z/OS.

Note

Modernizzazione del mainframe AWS Il servizio (Managed Runtime Environment experience) non è più aperto a nuovi clienti. Per funzionalità simili a Modernizzazione del mainframe AWS Service (Managed Runtime Environment experience), esplora Modernizzazione del mainframe AWS Service (Self-Managed Experience). I clienti

esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta [Modifica della Modernizzazione del mainframe AWS disponibilità](#).

2. I client associano l'indirizzo Network Load Balancer in [Amazon Route 53](#).
3. Route 53 reindirizza le transazioni al Network Load Balancer.
4. Il Network Load Balancer distribuisce le transazioni su più istanze EC2.
5. Il carico di lavoro su cui è in esecuzione Modernizzazione del mainframe AWS interagisce con Db2 utilizzando una connessione persistente tramite z/OS AWS Direct Connect

Test in corso

Quando si ripiattoforma un'applicazione COBOL mantenendo Db2 z/OS come database condiviso, è fondamentale garantire che il nuovo sistema funzioni in modo equivalente all'originale. Questo ambiente ibrido presenta sfide e opportunità uniche per i test. La seguente strategia delinea un approccio completo ai test di equivalenza funzionale ed è progettata per convalidare le prestazioni, l'integrità dei dati e la perfetta integrazione dell'applicazione ripiattoformata con il database Db2 for z/OS esistente.

Inizia identificando i processi e le transazioni aziendali critici che devono essere confrontati tra i sistemi. Quindi, crea un piano di test dettagliato con scenari specifici che valuteranno efficacemente l'equivalenza funzionale di queste transazioni. Infine, sviluppa set di dati di test completi che coprano tutti gli scenari identificati e assicurati che siano identici per entrambi i sistemi per consentire un confronto accurato.

Ambiente di origine

- Istantanea iniziale (prima istantanea):
 - Assicurati che la tabella di dati non venga utilizzata da altre applicazioni durante il test, poiché ciò può influire sul test di equivalenza.
 - Scatta un'istantanea del Db2 per le z/OS tabelle utilizzate dalla transazione prima di eseguire qualsiasi test.
- Test del sistema di origine:
 - Esegui la suite completa di test sull'applicazione COBOL originale.
 - Registra tutte le transazioni, gli input e gli output.
 - Monitora le prestazioni del sistema e l'utilizzo delle risorse.

- Istantanea di test post-source (seconda istantanea):
 - Scatta un'altra istantanea di Db2 for z/OS database dopo aver completato i test del sistema di origine.

Ambiente di destinazione

- Reimpostazione del database:
 - Ripristina il database allo stato iniziale utilizzando la prima istantanea.
- Test del sistema Target (ambiente riplatformer):
 - Esegui la stessa suite di test sull'applicazione replatformed.
 - Assicurati che tutti i test del sistema di destinazione utilizzino gli stessi input dei test del sistema di origine.
 - Monitora le prestazioni del sistema e l'utilizzo delle risorse.
- Istantanea del test post-target (terza istantanea):
 - Scatta un'istantanea finale di Db2 for z/OS database dopo aver completato i test del sistema di destinazione.

Analisi

- Confronto e analisi:
 - Confronta la seconda e la terza istantanea per identificare eventuali discrepanze nei dati.
 - Analizza i risultati dei test e confronta gli output dei sistemi di origine e di destinazione.
 - Valuta le metriche delle prestazioni tra i due ambienti.
- Test di integrazione:
 - Eseguite test che coinvolgono sia l'applicazione riplatformata che tutti i componenti COBOL rimanenti.
 - Verifica la perfetta interazione tra i due ambienti.
- Test di failover e ripristino:
 - Scenari di test in cui un ambiente fallisce e l'altro ambiente prende il sopravvento.
 - Garantisci la coerenza e l'integrità dei dati durante le situazioni di failover.
- Test di carico e stress:
 - Esegui test con carichi diversi per valutare le prestazioni del sistema ibrido sotto stress.

- Identifica eventuali rallentamenti o problemi di prestazioni in entrambi gli ambienti.
- Documentazione e reportistica:
 - Documenta tutti i risultati dei test, le discrepanze e le metriche delle prestazioni.
 - Prepara un rapporto completo che confronti i sistemi di origine e di destinazione.

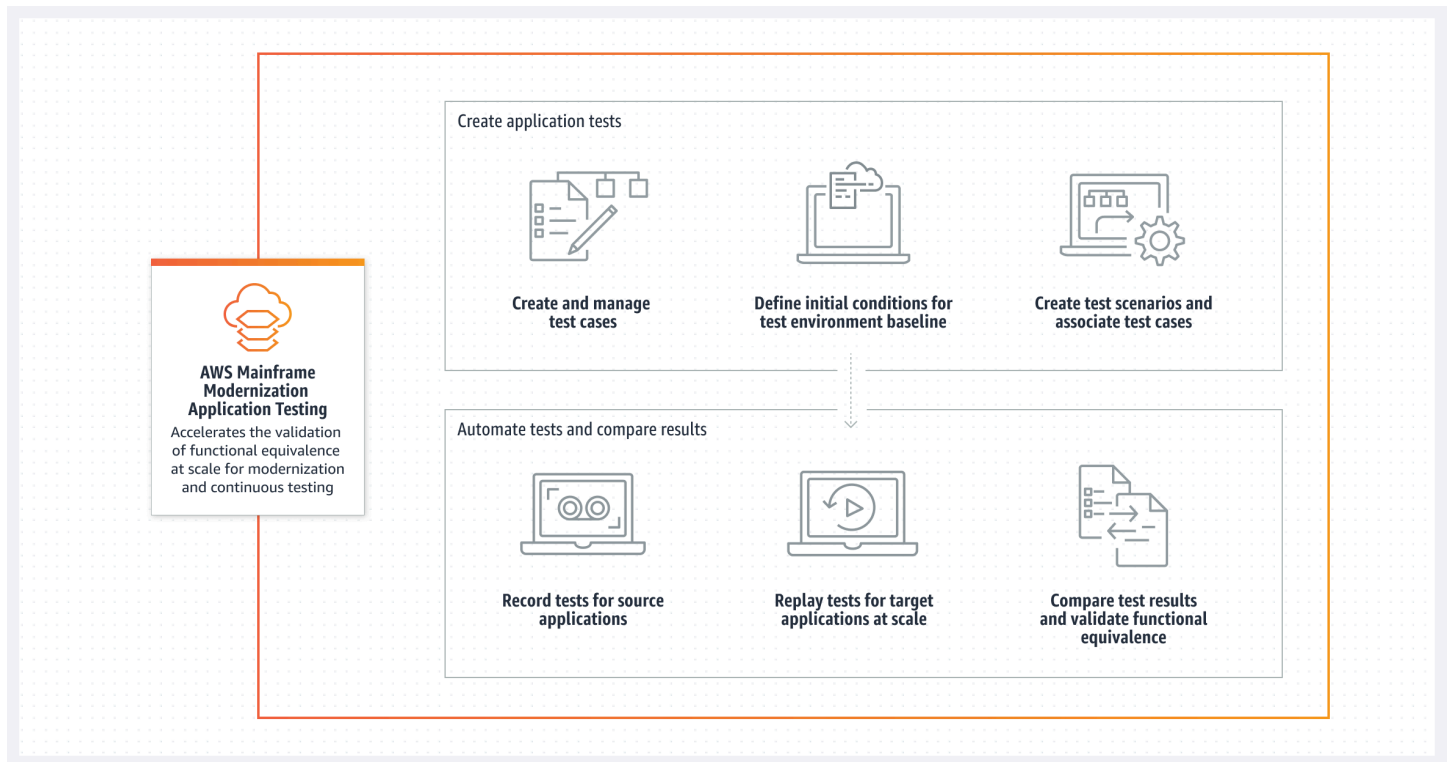
Test dell'applicazione in Modernizzazione del mainframe AWS

Il [AWS Mainframe Modernization Application Testing](#) servizio automatizza l'esecuzione dei test delle applicazioni su larga scala. AWS Test dell'applicazione aiuta a ottimizzare e ridurre i costi dei progetti di modernizzazione delle applicazioni mainframe e di test.

Note

Modernizzazione del mainframe AWS Il servizio (Managed Runtime Environment experience) non è più aperto a nuovi clienti. Per funzionalità simili a Modernizzazione del mainframe AWS Service (Managed Runtime Environment experience), esplora Modernizzazione del mainframe AWS Service (Self-Managed Experience). I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta [Modifica della Modernizzazione del mainframe AWS disponibilità](#).

Il diagramma seguente mostra come AWS Test dell'applicazione funziona ad alto livello.



Il processo prevede le seguenti fasi:

1. Crea e gestisci i casi di test, che sono l'unità più piccola di azioni di test. Identifica i tipi di dati che meglio rappresentano l'equivalenza funzionale tra il sistema di origine e quello di destinazione.
2. Definisci la configurazione dell'ambiente di test specificando CloudFormation modelli e attributi aggiuntivi.
3. Crea suite di test, che sono raccolte di casi di test.
4. Carica e riproduci set di dati: acquisisci i set di dati di input e output sul mainframe, caricali e quindi riproduci lo scenario di test sul sistema di destinazione. AWS
5. Confronta i set di dati di origine e di destinazione. AWS Test dell'applicazione confronta automaticamente i set di dati di output dei sistemi di origine e di destinazione. Esaminali e valutali per identificare le discrepanze.

Per ulteriori informazioni, consulta la documentazione [Modernizzazione del mainframe AWS](#).

Conversione

Nella modernizzazione del mainframe, una delle sfide più critiche è ridurre al minimo i tempi di inattività e i rischi durante la transizione verso una nuova piattaforma. La strategia blue/green di implementazione offre un approccio potente e flessibile alla migrazione dei sistemi.

L'implementazione blu/verde è una tecnica che riduce i tempi di inattività e i rischi eseguendo due ambienti di produzione identici denominati blu e verde. Ecco come funziona nel contesto della modernizzazione del mainframe:

- Ambiente blu: questo è il tuo attuale sistema mainframe che gestisce tutto il traffico di produzione.
- Ambiente ecologico: questa è la tua nuova piattaforma modernizzata pronta a subentrare. AWS

La strategia blue/green Cutover include le seguenti fasi: fornitura, messa in funzione, ripristino in caso di problemi e conclusione.

Fornitura

In questa fase, esegui il provisioning del nuovo ambiente (verde) AWS seguendo questi passaggi:

1. Ripiatmosfera l'ambiente: la zona ospitata da [Route 53](#) deve contenere un [record DNS](#) che punti all'ambiente mainframe (blu).
2. Verifica la connettività: assicurati una connessione corretta tra i tuoi gestori di transazioni Account AWS e quelli locali e Db2 for database. z/OS
3. Esegui test di fumo: utilizza l'indirizzo del AWS load balancer per accedere all'ambiente replatformed ed esegui test di fumo completi per verificare quanto segue:
 - Tutti i carichi di lavoro previsti sono disponibili.
 - 3270 transazioni vengono elaborate correttamente.
 - Le interazioni dei dati con Db2 for z/OS funzionano come previsto.

Trasmetti in diretta

In questa fase, sposti il traffico verso l'ambiente verde e controlli le modifiche.

1. Utilizza le politiche di routing del traffico in Route 53 per spostare il traffico:
 - Opzione A: puoi spostare il traffico tutto in una volta.

- Opzione B: In alternativa, è possibile utilizzare una distribuzione ponderata graduale.

2. Monitora e convalida:

- Osserva attentamente l' AWS ambiente mentre il traffico cambia.
- Controlla l'elaborazione delle transazioni 3270.
- Verifica Db2 per la z/OS comunicazione.
- Monitora i problemi di prestazioni.
- Chiedi agli utenti di convalidare i risultati delle transazioni.

Rollback

In caso di problemi, puoi aggiornare rapidamente Route 53 per reindirizzare il traffico verso l'ambiente mainframe locale (blu).

È necessario esaminare e risolvere i problemi prima di tentare un altro cutover.

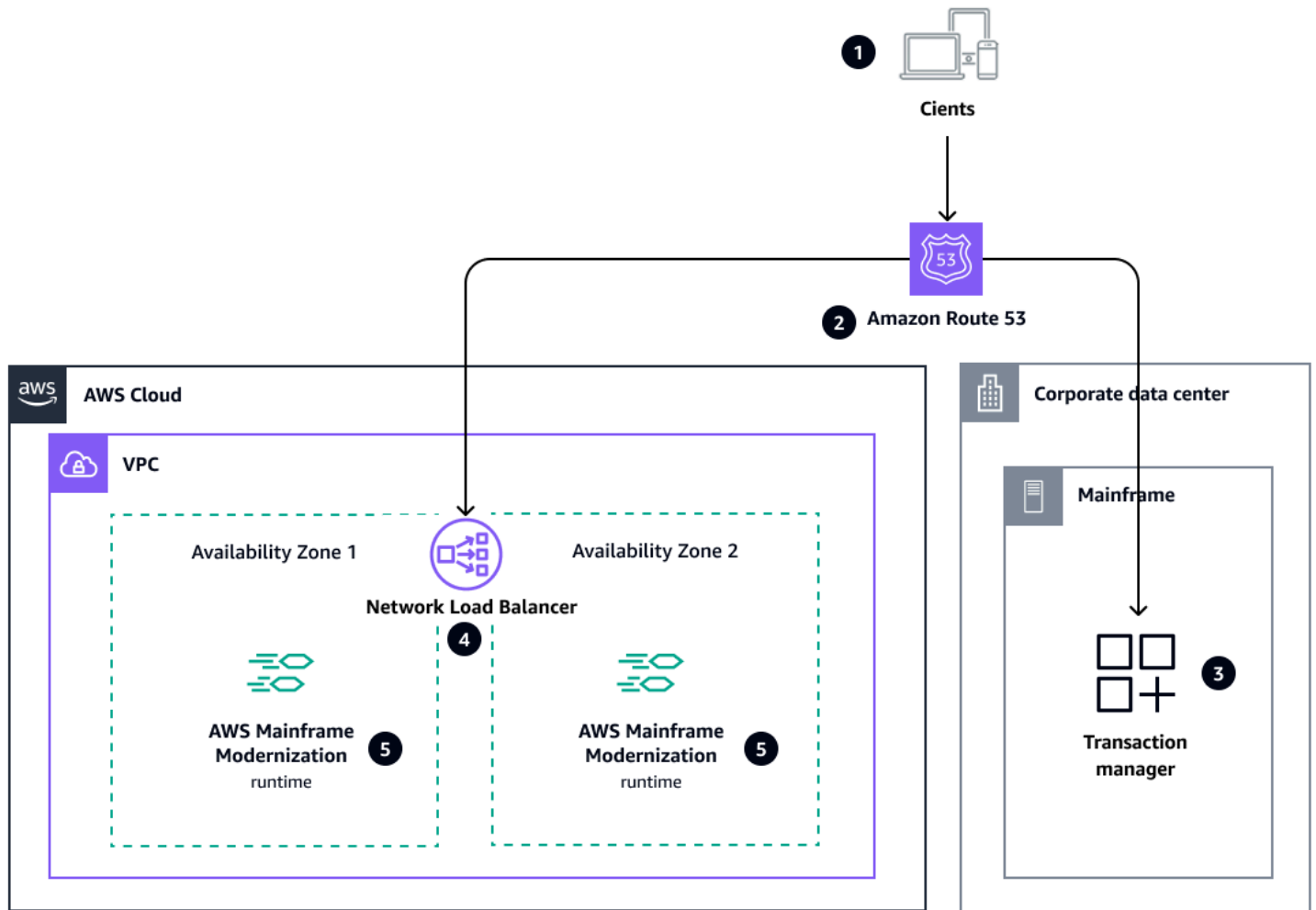
Concludi

Dopo aver monitorato il traffico e verificato che il tuo ambiente verde funzioni correttamente, puoi aumentare gradualmente il traffico dell'applicazione verso AWS.

Dopo un periodo stabile, è possibile disattivare l'ambiente di transazione mainframe (blu) e mantenere Db2 for z/OS database in locale.

Architecture

Il diagramma seguente illustra il flusso di cutover.



Il processo di cutover consiste nei seguenti elementi:

1. Le applicazioni client, i frontend e i backend per i frontend (BFFs) inviano le transazioni al nome di dominio Route 53.
2. Route 53 indirizza la connessione al gestore delle transazioni del mainframe o al Network Load Balancer, a seconda della politica di routing definita.
3. Il gestore delle transazioni elabora le transazioni inviate al mainframe.
4. Il Network Load Balancer distribuisce le transazioni negli ambienti di ripiattoforma disponibili per l'elaborazione.
5. Gli ambienti di Modernizzazione del mainframe AWS ripiattoforma elaborano le richieste.

Best practice

Questa sezione descrive una serie di best practice per affrontare le principali sfide legate alla ripiattoforma dei carichi di lavoro mainframe in ambienti cloud mantenendo il database su Db2 for z/OS.

Latenza di rete

Per prevedere con precisione l'impatto sulla latenza della separazione dell'applicazione dal database Db2 durante un processo di ripiattoforma, si consiglia di condurre una valutazione approfondita del numero di chiamate Db2 sia per le transazioni che per i processi batch. Questa valutazione deve essere eseguita utilizzando dati di traccia e deve includere i seguenti passaggi:

- **Raccogli dati di traccia:** raccogli tracce dettagliate delle transazioni rappresentative e dei lavori in batch e assicurati che le tracce acquisiscano tutte le interazioni Db2, incluse le entrate e le uscite.
- **Analizza i dati di traccia:** conta il numero di entrate e uscite Db2 per ogni transazione e processo batch e calcola il numero medio di interazioni Db2 per transazione e processo batch.
- **Misura i tempi di risposta attuali:** verifica se l'accesso a Db2 è in linea con il contratto sul livello di servizio (SLA) dell'applicazione.
- **Stima della latenza di rete:** determina la latenza di rete prevista tra l'applicazione ripiattoformata e il database Db2. Prendi in considerazione fattori come la distanza fisica, l'infrastruttura di rete e i potenziali colli di bottiglia.
- **Calcola l'impatto potenziale:** per ogni transazione e processo batch, moltiplica il numero di entrate e uscite da Db2 per la latenza di rete stimata. Aggiungi questo tempo calcolato ai tempi di risposta correnti per prevedere il nuovo tempo di elaborazione totale.
- **Valuta i risultati:** valuta se l'aumento di latenza previsto è accettabile per i requisiti aziendali e identifica eventuali transazioni o processi che potrebbero richiedere l'ottimizzazione o la riprogettazione per mitigare i problemi di latenza.
- **Prendi in considerazione le strategie di mitigazione:** esplora opzioni come il pool di connessioni, la memorizzazione nella cache o il recupero di dati in batch per ridurre il numero di interazioni Db2 individuali. Valuta la possibilità di spostare i dati a cui si accede di frequente più vicino al livello dell'applicazione.

Seguendo questi passaggi, sarai in grado di prendere decisioni basate sui dati sulla fattibilità della tua strategia di replatforming e identificare eventuali problemi di prestazioni prima che abbiano un impatto sull'ambiente di produzione. Questo approccio contribuirà a garantire una transizione fluida mantenendo al contempo livelli di prestazioni accettabili per le applicazioni dipendenti dal database.

Sicurezza

- Proteggi la build della tua applicazione: utilizza una sottorete privata nel cloud privato virtuale (VPC) da AWS CodeBuild eseguire per garantire l'isolamento e una maggiore sicurezza. Implementa un contesto affidabile Db2 dalla CodeBuild sottorete CIDR per un accesso sicuro al database durante il processo di creazione.
- Proteggi il tuo ambiente di runtime: utilizza un contesto affidabile Db2 dalla sottorete di runtime CIDR per connessioni sicure al database.
- Gestisci le credenziali del database in modo sicuro: implementa una pianificazione regolare di rotazione delle credenziali per ridurre al minimo il rischio di accesso non autorizzato. Archivia le credenziali Db2 in modo sicuro in. Gestione dei segreti AWS
- Stabilisci la sicurezza della rete: implementa solide regole di segmentazione della rete e firewall per proteggere sia gli ambienti di compilazione che quelli di runtime. Utilizza la combinazione corretta di AWS Direct Connect e AWS Site-to-Site VPN per raggiungere il livello di sicurezza necessario.
- Applica la crittografia: applica la crittografia per i dati in transito tra l'applicazione e Db2 for z/OS.

Governance delle applicazioni

- Stabilisci una fonte affidabile: stabilisci la nuova gestione della configurazione del software (SCM), ad esempio, GitHub come unica fonte di verità per il codice dell'applicazione migrato. Ciò garantisce la coerenza ed elimina le discrepanze di versione tra gli ambienti cloud e mainframe durante il periodo di transizione.
- Aggiorna il processo di gestione delle modifiche: aggiorna il processo di gestione delle modifiche per prendere in considerazione le modifiche al codice in questo nuovo paradigma a doppio ambiente. Questo processo dovrebbe includere:
 - Flussi di lavoro di approvazione chiari per le modifiche al codice.
 - Revisioni obbligatorie del codice prima di unirlo al ramo principale.
 - Procedure di distribuzione sincronizzate per garantire che entrambi gli ambienti ricevano aggiornamenti contemporaneamente.

- Meccanismi di ripristino in caso di problemi in entrambi gli ambienti.

Elasticità

L'elasticità del cloud computing introduce un cambio di paradigma che altera in modo significativo la struttura dei costi e la gestione delle risorse del mainframe. A differenza dell'ambiente mainframe tradizionale, che ha capacità fissa e modelli di prezzo basati sui picchi, le piattaforme cloud offrono una scalabilità dinamica e un pay-as-you-go approccio che può potenzialmente portare a notevoli risparmi sui costi e a una migliore efficienza operativa.

In un ambiente cloud, le organizzazioni possono aumentare o diminuire le proprie risorse di elaborazione in tempo reale in base alla domanda effettiva, eliminando così la necessità di un overprovisioning per far fronte ai picchi di carico. Questa elasticità consente alle aziende di pagare solo per le risorse che consumano invece di investire in costose licenze hardware e software per gestire picchi di utilizzo occasionali.

[Per informazioni dettagliate su come funzionano i prezzi AWS, consulta Prezzi.AWS](#)

Fasi successive

La modernizzazione del mainframe è un'iniziativa complessa e critica che richiede conoscenze specializzate e soluzioni avanzate. Potete accelerare il processo di modernizzazione e ottenere risultati aziendali più rapidi attraverso [partnership strategiche](#) che vi aiuteranno nelle seguenti attività:

- Valuta e stabilisci le priorità: esamina le tue applicazioni mainframe e identifica quelle adatte alla ripiattoforma mantenendo il database su Db2 for z/OS. Prendi in considerazione fattori quali la complessità, la criticità aziendale e il potenziale ritorno sull'investimento (ROI).
- Sviluppa una strategia di migrazione: crea un piano dettagliato per la ripiattoforma delle applicazioni selezionate, che includa tempistiche, allocazione delle risorse e strategie di mitigazione del rischio.
- Valuta strumenti e tecnologie: ricerca e seleziona strumenti e tecnologie appropriati per facilitare il processo di ripiattoforma, come le piattaforme di modernizzazione delle applicazioni o gli strumenti di conversione del codice.
- Interagisci con gli esperti: prendi in considerazione la possibilità di collaborare con specialisti della modernizzazione del mainframe o con società di consulenza che hanno esperienza in progetti di ripiattoforma.
- Prova di fattibilità: inizia con un proof of concept su piccola scala per convalidare il tuo approccio e identificare potenziali sfide prima di passare ad applicazioni più grandi.
- Test e convalida: sviluppate una strategia di test completa per garantire che le applicazioni ripiattoformate funzionino correttamente e mantengano l'integrità dei dati con il vostro database Db2 for database esistente. z/OS
- Formazione e trasferimento delle conoscenze: preparate il vostro team per il nuovo ambiente fornendo corsi di formazione sulle applicazioni ripiattoformate e su eventuali nuovi strumenti o tecnologie introdotti.
- Implementazione graduale: prendete in considerazione un approccio graduale alla ripiattoforma, che prevede la migrazione graduale delle applicazioni monitorando le prestazioni e risolvendo eventuali problemi.
- Ottimizzazione continua: dopo la ripiattoforma, monitora e ottimizza continuamente le prestazioni delle applicazioni e le loro interazioni con Db2 for database per garantire un successo a lungo termine. z/OS
- Modernizza al tuo ritmo: ora che il carico di lavoro è in esecuzione AWS e sta già sfruttando i vantaggi del cloud, inizia a pianificare la fase di re-immaginazione della tua modernizzazione.

Resources

Per ulteriori informazioni sulla migrazione e la modernizzazione del mainframe, consulta le seguenti risorse.

AWS documentazione

- [Configurazione di Amazon Route 53 come servizio DNS](#)
- [Instradamento del traffico verso un sistema di bilanciamento del carico ELB](#)
- [Routing ponderato](#)
- [Ripiattoforma delle applicazioni con Rocket Software](#)

Riferimenti a Rocket Software

- [Interfaccia di chiamata esterna Micro Focus \(ECI\)](#)
- [Servizi Web CICS](#)

Riferimenti IBM

- [Contesti affidabili](#) (IBM Db2 per la documentazione) z/OS

Tools (Strumenti)

- [Server Rocket Enterprise](#)

AWS Modelli e guide di orientamento prescrittivo

- [Crea programmi COBOL Db2 utilizzando e Modernizzazione del mainframe AWSAWS CodeBuild](#)
- [DevOps per Modernizzazione del mainframe AWS](#)
- [Modernizzazione del mainframe: modelli di disaccoppiamento per la migrazione del codice dell'applicazione](#)

-
- [Proteggi e semplifica l'accesso degli utenti in un database federativo Db2 utilizzando contesti affidabili AWS](#)

Cronologia dei documenti

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

Modifica	Descrizione	Data
Pubblicazione iniziale	—	7 maggio 2025

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.](#)

ripiattoforma

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.