



Test di applicazioni serverless su AWS

# AWS Linee guida prescrittive



---

# AWS Linee guida prescrittive: Test di applicazioni serverless su AWS

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

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

---

# Table of Contents

Introduzione .....	1
Panoramica di .....	1
Prerequisiti .....	2
Definizioni .....	2
Obiettivi .....	2
Aumentare la qualità del software .....	2
Ridurre il tempo necessario per implementare o modificare le funzionalità .....	5
Tecniche di test per applicazioni serverless .....	6
Test con mock .....	6
Test di emulazione .....	7
Test nel cloud .....	8
Sfide legate al test di applicazioni serverless .....	10
Esempio: una funzione Lambda che crea un bucket Amazon S3 .....	10
Esempio: una funzione Lambda che elabora i messaggi da Amazon SQS .....	11
Best practice per testare le applicazioni serverless .....	12
Dai priorità ai test nel cloud .....	12
Usa dei mock se necessario .....	12
Comprendi i compromessi dei test di emulazione .....	13
Amplia i test attraverso i confini naturali .....	14
Identifica i confini dell'architettura .....	14
Separare il codice Lambda dalla logica aziendale .....	14
Tratta i confini come contratti .....	14
Utilizza test harness per flussi di lavoro asincroni .....	15
Organizza ambienti cloud per l'isolamento degli sviluppatori .....	16
Accelera i cicli di feedback .....	16
Domande frequenti .....	17
Ho una funzione Lambda che esegue calcoli e restituisce un risultato senza chiamare altri servizi. Devo davvero testarla nel cloud? .....	17
In che modo i test nel cloud possono contribuire ai test unitari? Se è nel cloud e si connette ad altre risorse, non è un test di integrazione? .....	17
Risorse e passaggi successivi .....	19
Implementazioni di esempio .....	19
Approfondimenti .....	19
Riferimenti .....	19

---

Tools (Strumenti) .....	19
Collaboratori .....	20
Creazione .....	20
Revisione .....	20
Scrittura tecnica .....	20
Cronologia dei documenti .....	21
Glossario .....	22
# .....	22
A .....	23
B .....	26
C .....	28
D .....	31
E .....	35
F .....	37
G .....	39
H .....	40
I .....	41
L .....	44
M .....	45
O .....	49
P .....	52
Q .....	54
R .....	55
S .....	58
T .....	62
U .....	63
V .....	64
W .....	64
Z .....	65
.....	lxvii

# Test di applicazioni serverless su AWS

Amazon Web Services ([???](#)collaboratori)

Marzo 2026 (cronologia dei [documenti](#))

Questa guida illustra le metodologie per testare applicazioni serverless, descrive le sfide che potresti incontrare durante i test e introduce le best practice. Queste tecniche di test hanno lo scopo di aiutarti a iterare più rapidamente e a rilasciare il codice con maggiore sicurezza.

Questa guida è destinata agli sviluppatori che desiderano stabilire strategie di test per le proprie applicazioni serverless. Puoi utilizzare la guida come punto di partenza per conoscere le strategie di test e poi visitare il [repository Serverless Test Samples](#) per vedere esempi di test che seguono i modelli e le procedure consigliate descritti in questa guida. Questa guida descrive le metodologie di test serverless, descrive le sfide che i clienti incontrano quando testano le applicazioni serverless e introduce le migliori pratiche per testare le applicazioni serverless. Queste tecniche hanno lo scopo di aiutare gli sviluppatori a iterare più rapidamente e rilasciare con maggiore sicurezza.

## Panoramica di

I test automatizzati sono investimenti fondamentali che aiutano a garantire la qualità delle applicazioni e la velocità di sviluppo. I test accelerano anche il feedback degli sviluppatori. In qualità di sviluppatore, vuoi essere in grado di iterare rapidamente sulla tua applicazione e ottenere feedback sulla qualità del codice. Molti sviluppatori sono abituati a scrivere applicazioni da implementare in un ambiente desktop, direttamente nel sistema operativo o all'interno di un ambiente basato su container. Quando si lavora in ambienti desktop o basati su container, in genere si scrivono test su un codice ospitato interamente sul desktop. Tuttavia, nelle applicazioni serverless, i componenti dell'architettura potrebbero non essere implementabili in un ambiente desktop, ma potrebbero invece esistere solo nel cloud. Un'architettura basata sul cloud può includere livelli di persistenza, sistemi di messaggistica, costrutti di sicurezza e altri componenti. APIs Quando si scrive un codice applicativo basato su questi componenti, potrebbe essere difficile determinare il modo migliore per progettare ed eseguire i test.

Questa guida ti aiuta ad allinearti a una strategia di test che riduce l'attrito e la confusione e aumenta la qualità del codice.

## Prerequisiti

Questa guida presuppone che tu abbia familiarità con le nozioni di base dei test automatizzati, incluso il modo in cui vengono utilizzati i test automatizzati del software per garantire la qualità del software. La guida fornisce un'introduzione di alto livello a una strategia di test delle applicazioni serverless e non richiede alcuna esperienza pratica nella scrittura di test.

## Definizioni

Questa guida usa i termini seguenti:

- Test unitari, sono test eseguiti isolatamente sul codice per un singolo componente dell'architettura.
- I test di integrazione vengono eseguiti su due o più componenti architettonici, in genere in un ambiente cloud.
- End-to-end i test verificano i comportamenti su intere applicazioni o flussi di lavoro.
- Gli emulatori sono applicazioni (spesso fornite da terze parti) progettate per simulare un servizio cloud senza fornire o richiamare alcuna risorsa cloud.
- Mock (chiamati anchefalsi), sono implementazioni in un'applicazione di test che sostituiscono una dipendenza con una simulazione di tale dipendenza.

## Obiettivi

Le best practice riportate in questa guida hanno lo scopo di aiutarti a raggiungere due obiettivi principali:

- Aumentare la qualità delle applicazioni serverless
  - Test ai confini dell'architettura
  - Test ai limiti del codice
- Ridurre il tempo necessario per implementare o modificare le funzionalità

## Aumentare la qualità del software

La qualità di un'applicazione dipende in larga misura dalla capacità degli sviluppatori di testare una varietà di scenari per verificarne la funzionalità. Quando non si implementano test automatici o, più in

genere, se i test non coprono adeguatamente gli scenari richiesti, la qualità dell'applicazione non può essere determinata o garantita.

In un'architettura basata su server, i team possono definire facilmente l'ambito di test: qualsiasi codice che viene eseguito sul server applicativo deve essere testato. Gli altri componenti che richiamano il server, o le dipendenze richiamate dal server, sono spesso considerati esterni e non possono essere testati dal team responsabile dell'applicazione sul server.

Le applicazioni serverless sono spesso costituite da unità di lavoro più piccole, ad esempio funzioni AWS Lambda, eseguite nel proprio ambiente. Probabilmente, i team saranno responsabili di molte di queste unità più piccole all'interno di una singola applicazione. Alcune funzionalità delle applicazioni possono essere delegate interamente a servizi gestiti come Amazon Simple Storage Service (Amazon S3) o Amazon Simple Queue Service (Amazon SQS) senza utilizzare alcun codice sviluppato internamente. I modelli tradizionali basati su server per il test del software possono escludere i servizi gestiti considerandoli esterni all'applicazione. Ciò può portare a una copertura inadeguata, in cui gli scenari critici potrebbero essere limitati a test esplorativi manuali o a pochi casi di test di integrazione in cui il risultato varia a seconda dell'ambiente. Pertanto, l'adozione di strategie di test che comprendano comportamenti dei servizi gestiti e configurazioni cloud può migliorare la qualità del software.

## Test ai confini dell'architettura

Man mano che le applicazioni serverless crescono, si diffondono naturalmente su più componenti architettonici. Sebbene ciò utilizzi funzionalità AWS distribuite, può rendere difficile la comprensione end-to-end del comportamento.

### Identificazione dei confini naturali

Quando progetti la tua architettura seguendo le migliori pratiche serverless (una funzione = un lavoro, disaccoppiamento), noterai i confini naturali tra i sottosistemi. Questi limiti rappresentano punti di separazione logica nell'applicazione.

### I confini come contratti di test

Questi confini architettonici sono ottimi candidati per testare i bordi. Tratta ogni limite come un contratto e verifica che si comporti secondo le specifiche definite. Pensate a questi limiti come a delle giunture nella vostra applicazione in cui inserire la convalida dei test.

### Vantaggi principali

Di seguito sono riportati i principali vantaggi del test ai confini dell'architettura:

- **Ambito di test mirato:** testa i sottosistemi in modo indipendente senza dover comprendere l'intera applicazione.
- **Convalida del contratto:** assicurati che ogni limite mantenga il comportamento previsto man mano che il sistema si evolve
- **Strumentazione a doppio scopo:** questi stessi limiti offrono eccellenti vantaggi in termini di osservabilità durante la produzione
- **Test harness:** consente di testare sistemi serverless asincroni. Ti aiuta a testare architetture basate sugli eventi acquisendo e convalidando gli eventi mentre fluiscono nel sottosistema.

## Test ai limiti del codice

Definisci confini chiari del codice separando il codice dell'infrastruttura, come il codice Lambda, dalla tua logica aziendale principale. Questa separazione crea ambiti di test distinti che semplificano la strategia di test.

Lo schema di confine

Stabilisci due limiti di codice chiari nelle tue funzioni Lambda:

- **Limite esterno (gestore Lambda):** un sottile strato adattatore che gestisce problemi specifici per AWS Lambda
- **Limite interno (business logic):** metodi di logica aziendale puri indipendenti dal runtime Lambda

Gestore come adattatore (ambito esterno)

Il gestore di funzioni Lambda dovrebbe essere un livello sottile che:

- Estrae i dati dagli oggetti in entrata e dagli oggetti event context
- Convalida i dati estratti
- Passa solo i dettagli rilevanti ai metodi di logica aziendale
- Restituisce i risultati nel formato previsto per Lambda

Logica aziendale (ambito interno)

La tua logica aziendale principale dovrebbe:

- Funziona indipendentemente dai dettagli specifici di Lambda

- Accetta input semplici e convalidati
- Restituisci output prevedibili
- Richiede dipendenze minime per l'inizializzazione

### Vantaggi dei test per ambito

- Test dei limiti interni: test unitari completi sulla logica aziendale senza complessità Lambda o configurazione dell'ambiente
- Test sui limiti esterni: test di integrazione mirati che convalidano la gestione degli eventi e l'estrazione dei dati a livello di adattatore
- Sovraccarico minimo per i test: per la maggior parte dei test non sono necessari ambienti complessi o dipendenze estese

Questo approccio basato sui limiti ti consente di testare la maggior parte del codice come funzioni pure, mantenendo i test Lambda minimi e mirati.

## Ridurre il tempo necessario per implementare o modificare le funzionalità

È possibile ridurre al minimo l'effetto dei bug del software e dei problemi di configurazione su costi e pianificazioni rilevando questi problemi durante un ciclo di sviluppo iterativo. Quando uno sviluppatore non riesce a rilevare questi problemi, più persone devono impegnarsi ulteriormente per identificarli.

Un'architettura serverless può includere servizi gestiti che forniscono funzionalità applicative critiche tramite chiamate API. Per questo motivo, il ciclo di sviluppo dovrebbe includere test che convalidino sia il percorso positivo (in cui le interazioni con questi servizi si comportano come previsto) sia il percorso triste (in cui le chiamate falliscono, restituiscono risposte impreviste o si comportano diversamente tra gli ambienti). Senza questi test, è possibile riscontrare problemi derivanti dalle differenze tra l'ambiente locale e l'ambiente distribuito. Quando ciò accade, è necessario dedicare più tempo al tentativo di riprodurre e verificare una correzione, poiché ogni iterazione ora richiede la convalida delle modifiche rispetto a un ambiente diverso dalla configurazione preferita.

Un'adeguata strategia di test serverless migliora i tempi di iterazione fornendo risultati accurati per i test che includono chiamate ad altri servizi.

# Tecniche di test per applicazioni serverless

Esistono tre approcci principali per eseguire test su codice applicativo serverless: test fittizi, test di emulazione e test nel cloud. In genere è possibile trovare una combinazione di questi approcci in uso nell'ambito di un singolo progetto. Ad esempio, potresti utilizzare i test simulati quando sviluppi il codice a livello locale, i test di emulazione per replicare più fedelmente il comportamento del servizio prima della distribuzione e i test sul cloud come parte di un processo notturno di integrazione e distribuzione continua (CI/CD).

## Test con mock

Il test con mock è una strategia che consiste nel creare oggetti sostitutivi nel codice che simulino il comportamento dei servizi cloud. Ad esempio, puoi scrivere un test che utilizza un mock del servizio Amazon S3 e puoi configurare quel mock per restituire un oggetto di risposta specifico ogni volta che viene chiamato `PutObject` il metodo. Quando viene eseguito un test, il mock restituisce la risposta senza chiamare Amazon S3 o altri endpoint del servizio.

Questi oggetti sostitutivi vengono spesso generati da un framework fittizio per ridurre la quantità di implementazione necessaria per un determinato test. Alcuni framework fittizi sono generici e altri sono progettati specificamente per AWS SDKs

I mock, insieme agli stub, rientrano nella categoria più ampia dei fake. I mock differiscono dagli emulatori (discussi più avanti in questa sezione) in quanto i mock vengono in genere creati o configurati da uno sviluppatore come parte del codice di test, mentre gli emulatori sono applicazioni autonome che espongono APIs (come REST APIs) allo stesso modo dei sistemi che emulano.

Di seguito puoi trovare alcuni vantaggi legati all'utilizzo dei mock:

- I simulatori possono simulare servizi di terze parti che sfuggono al controllo dell'applicazione, come APIs i fornitori di software as a service (SaaS), senza bisogno di accedere direttamente a tali servizi.
- I mock sono anche utili per testare le condizioni di errore, soprattutto quando tali condizioni (come nel caso delle interruzioni del servizio) sono difficili da simulare, come un'interruzione del servizio.
- Come gli emulatori, i framework fittizi possono fornire uno sviluppo locale rapido dopo essere stati configurati.

- I mock possono fornire un comportamento sostitutivo praticamente per qualsiasi tipo di oggetto, quindi le strategie di simulazione possono riguardare una più ampia varietà di servizi rispetto agli emulatori.
- Quando diventano disponibili nuove funzionalità o comportamenti, i test simulati possono reagire più rapidamente utilizzando (o ricorrendo a) un framework fittizio generico, in grado di simulare le nuove funzionalità non appena l'SDK aggiornato diventa disponibile. AWS

I test con mock presentano i seguenti svantaggi:

- Generalmente, i mock richiedono un notevole impegno per l'impostazione e la configurazione, in particolare quando si cerca di determinare i valori restituiti da servizi diversi al fine di simulare correttamente le risposte.
- Poiché i mock sono scritti o configurati dagli sviluppatori che scrivono i test, si tratta di una responsabilità aggiuntiva per gli sviluppatori.
- Potrebbe essere necessario accedere al cloud per comprendere APIs e restituire i valori dei servizi.
- I mock possono anche essere difficili da gestire, perché richiedono aggiornamenti ogni volta che le firme delle API cloud simulate cambiano, gli schemi di valori restituiti si evolvono o la logica dell'applicazione testata viene estesa per effettuare chiamate a new APIs. Queste modifiche creano un carico di lavoro aggiuntivo per lo sviluppo dei test per gli sviluppatori.
- I test simulati potrebbero essere superati localmente ma fallire nel cloud perché simulano, anziché replicare, il comportamento di servizi reali, lasciando inosservati i problemi specifici dell'ambiente.
- I framework fittizi, come gli emulatori, non sono in grado di rilevare violazioni delle policy AWS Identity and Access Management (IAM), limiti delle quote di servizio o vincoli di dimensione del payload, né attivano servizi ausiliari come Amazon AWS CloudTrail o Amazon GuardDuty che possono influire sul comportamento delle CloudWatch applicazioni in un ambiente distribuito.
- Sebbene i mock siano più efficaci nel simulare ciò che farà un'applicazione quando l'autorizzazione fallisce o viene superata una quota, i test fittizi non sono in grado di determinare quale risultato si verificherà effettivamente quando il codice viene distribuito nell'ambiente di produzione.

## Test di emulazione

I test di emulazione sono abilitati da applicazioni eseguite localmente note come emulatori che assomigliano. Servizi AWS Gli emulatori hanno APIs caratteristiche simili alle loro controparti cloud e forniscono valori di ritorno simili. Possono anche simulare cambiamenti di stato avviati dalle chiamate

API. Ad esempio, un emulatore Amazon S3 potrebbe archiviare un oggetto sul disco locale quando viene chiamato il `PutObject` metodo. Quando `GetObject` viene chiamato con la stessa chiave, l'emulatore legge lo stesso oggetto dal disco e lo restituisce.

I vantaggi dei test di emulazione sono molteplici:

- Offrono l'ambiente più familiare per gli sviluppatori che sono abituati a sviluppare codice esclusivamente in un ambiente locale. Ad esempio, se avete dimestichezza con lo sviluppo di un'applicazione di livello n, potreste avere un motore di database e un server Web, simili a quelli in esecuzione in produzione, in esecuzione sul computer locale per fornire funzionalità di test rapide, locali e isolate.
- Non richiede alcuna modifica all'infrastruttura cloud (come gli account cloud degli sviluppatori), quindi è facile da implementare con i modelli di test esistenti. I test di emulazione hanno il vantaggio di offrire iterazioni di sviluppo locali rapide.

Tuttavia, gli emulatori presentano anche diversi svantaggi:

- Spesso sono difficili da configurare e replicare, soprattutto quando vengono utilizzati come parte delle CI/CD pipeline. Questo potrebbe aumentare il carico di lavoro e la manutenzione per il personale IT o per gli sviluppatori che gestiscono le proprie installazioni software.
- Le funzionalità emulate sono APIs in genere in ritardo rispetto alle modifiche dei servizi e potrebbero ostacolare l'adozione di nuove funzionalità fino all'aggiunta del supporto.
- Gli emulatori potrebbero richiedere supporto, aggiornamenti, correzioni di bug o miglioramenti della parità delle funzionalità, che sono di responsabilità dell'autore dell'emulatore (spesso si tratta di società terze).
- I test che si basano sugli emulatori possono fornire risultati positivi a livello locale, ma potrebbero fallire nel cloud a causa delle interazioni con altri aspetti, AWS come le politiche e le quote IAM, che generalmente non vengono emulate.
- Alcuni Servizi AWS non dispongono di emulatori, quindi se ti affidi solo all'emulazione, potresti non avere un'opzione di test soddisfacente per alcune parti dell'applicazione.

## Test nel cloud

I test nel cloud sono il processo di esecuzione di test su codice implementato in un ambiente cloud anziché in un ambiente desktop. Se ci si affida allo sviluppo di applicazioni native del cloud, il valore dei test nel cloud aumenta. Esempio:

- Puoi eseguire test nel cloud sulle funzionalità APIs e sui servizi più recenti, assicurandoti che riflettano i valori e i comportamenti restituiti più recenti mentre AWS continui a lanciare nuovi servizi e funzionalità.
- I test possono riguardare le policy IAM, le Service Quotas, le configurazioni e tutti i servizi.
- Il tuo ambiente di test cloud è quello che somiglia di più al tuo ambiente di runtime di produzione, quindi è probabile che i test eseguiti nel cloud ottengano risultati più coerenti man mano che avanzano negli ambienti.

I test nel cloud presentano tuttavia alcuni svantaggi. tra cui i seguenti:

- Le implementazioni in ambienti cloud richiedono in genere più tempo rispetto alle implementazioni in ambienti desktop. Strumenti come [AWS Serverless Application Model \(AWS SAM\) Accelerate](#), [AWS Cloud Development Kit \(AWS CDK\) watch mode](#) e [SST](#) aiutano a ridurre la latenza associata alle iterazioni di implementazione del cloud.
- I test nel cloud possono comportare costi di servizio aggiuntivi, a differenza dei test locali, che sfruttano l'ambiente di sviluppo locale.
- I test nel cloud potrebbero essere meno fattibili se non disponi di un accesso a Internet ad alta velocità.
- Nei settori regolamentati, le politiche di sicurezza aziendali possono limitare l'accesso degli sviluppatori agli ambienti cloud, rendendo difficile o impossibile l'esecuzione di test sul cloud come parte di un flusso di lavoro di sviluppo locale.
- I limiti dell'ambiente vengono spesso tracciati a livello di stack negli account condivisi per gli ambienti di sviluppo, a volte utilizzando strategie di tipo namespace come l'uso di prefissi per identificare la proprietà. Generalmente, per gli ambienti di preproduzione e produzione vengono fissati dei limiti a livello di account per isolare i carichi di lavoro dai problemi di tipo noisy neighbor, per favorire i controlli di sicurezza con privilegi minimi e per proteggere i dati sensibili. La necessità di creare ambienti isolati potrebbe comportare oneri aggiuntivi per i DevOps team, soprattutto se lavorano in un'azienda con controlli rigorosi su account e infrastruttura.

## Sfide legate al test di applicazioni serverless

Quando utilizzate emulatori e chiamate simulate per testare un'applicazione serverless sul desktop locale, potreste riscontrare delle incongruenze nei test man mano che il codice progredisce da un ambiente all'altro nella pipeline. CI/CD Gli unit test creati sul desktop per convalidare la logica di business dell'applicazione potrebbero non includere o rappresentare accuratamente gli aspetti critici dei servizi cloud. I test completi non possono essere eseguiti localmente in modo isolato. Richiedono la verifica delle autorizzazioni e delle configurazioni tra più servizi.

Le sezioni seguenti descrivono le sfide che potresti incontrare durante l'implementazione di una strategia di test sul cloud. Le sezioni seguenti descrivono le sfide che i clienti incontrano quando cercano di implementare una strategia di test sul cloud e le nostre linee guida sulle migliori pratiche per ottenere una copertura efficace dei test.

### Esempio: una funzione Lambda che crea un bucket Amazon S3

Se la logica di una funzione Lambda dipende dalla creazione di un bucket Amazon S3, un test completo dovrebbe confermare che Amazon S3 è stato chiamato correttamente e che il bucket è stato creato con successo. In una configurazione di test con mock, potresti simulare una risposta con esito positivo e potenzialmente aggiungere un caso di test per gestire una risposta non esito negativo. In uno scenario di test di emulazione, è possibile chiamare l'CreateBucketAPI, ma l'identità che effettua la chiamata non avrà origine dal servizio Lambda che assume un ruolo e verrà invece utilizzata un'autenticazione segnaposto: si tratta spesso del ruolo o dell'identità utente più permissivi.

Le configurazioni di simulazione ed emulazione discusse in precedenza molto probabilmente eseguiranno test su cosa farà la funzione Lambda se la chiamata ad Amazon S3 va a buon fine (o meno). Tuttavia, questi test non riusciranno a stabilire se la funzione Lambda è in grado di creare correttamente il bucket, data la configurazione della funzione. Questa configurazione è probabilmente rappresentata dall'infrastruttura come codice (IaC) per prodotti e servizi come AWS CloudFormation AWS SAM, o Terraform. HashiCorp Un possibile problema è che il ruolo assegnato alla funzione non dispone di una policy allegata che consenta l'`s3:CreateBucket` azione e, pertanto, la funzione fallirà sempre quando viene distribuita in un ambiente cloud.

# Esempio: una funzione Lambda che elabora i messaggi da Amazon SQS

Se una coda Amazon Simple Queue Service (Amazon SQS) è l'origine di una funzione Lambda, un test completo dovrebbe verificare che la funzione Lambda venga richiamata correttamente quando un messaggio viene messo in coda. I test di emulazione e i test simulati sono generalmente configurati per eseguire direttamente il codice della funzione Lambda e per simulare l'integrazione di Amazon SQS passando un payload di eventi JSON (o un oggetto deserializzato) come input del gestore di funzioni.

I test locali che simulano l'integrazione di Amazon SQS verificheranno cosa farà la funzione Lambda quando viene chiamata da Amazon SQS con un determinato payload, ma non garantiranno che Amazon SQS richiami correttamente la funzione Lambda quando viene distribuita in un ambiente cloud.

Di seguito troverai alcuni esempi di problemi di configurazione che potresti riscontrare con Amazon SQS e Lambda:

- Il timeout di visibilità di Amazon SQS è troppo basso e comporta più chiamate quando ne era prevista una sola.
- Il ruolo di esecuzione della funzione Lambda non consente la lettura dei messaggi dalla coda (tramite `sqs:ReceiveMessages`, `sqs:DeleteMessage`, o). `sqs:GetQueueAttributes`
- L'evento di esempio passato alla funzione Lambda supera la quota relativa alla dimensione dei messaggi di Amazon SQS. Pertanto, il test non è valido perché Amazon SQS non sarebbe mai in grado di inviare un messaggio di tali dimensioni.

Come mostrano questi esempi, con molta probabilità si otterranno risultati inattendibili dai test che riguardano la logica aziendale ma non le configurazioni tra i servizi cloud.

# Best practice per testare le applicazioni serverless

Le sezioni seguenti descrivono le best practice per ottenere una copertura efficace durante il test di applicazioni serverless.

## Dai priorità ai test nel cloud

Per applicazioni ben progettate, puoi utilizzare una varietà di tecniche di test per soddisfare una serie di requisiti e condizioni. Tuttavia, sulla base degli strumenti attuali, ti consigliamo di concentrarti il più possibile sui test nel cloud. Sebbene i test nel cloud possano creare latenza per gli sviluppatori, aumentare i costi e talvolta richiedere investimenti in DevOps controlli aggiuntivi, questa tecnica offre la copertura di test più affidabile, accurata e completa.

Devi avere accesso ad ambienti isolati in cui eseguire i test. Idealmente, ogni sviluppatore dovrebbe disporre di un programma dedicato Account AWS per evitare problemi di denominazione delle risorse che possono verificarsi quando più sviluppatori che lavorano sullo stesso codice cercano di distribuire o richiamare chiamate API su risorse con nomi identici. Questi ambienti devono essere configurati con avvisi e controlli adeguati per evitare spese inutili. Ad esempio, è possibile limitare il tipo, il livello o la dimensione delle risorse che possono essere create e impostare avvisi e-mail quando i costi stimati superano una determinata soglia.

Se devi dividerne una Account AWS con altri sviluppatori, i processi di test automatici dovrebbero denominare le risorse in modo univoco per ogni sviluppatore. Ad esempio, gli script di aggiornamento o i file di configurazione TOML che causano i comandi AWS SAM CLI [sam deploy](#) o [sam sync](#) possono specificare automaticamente un nome di stack che include il nome utente dello sviluppatore locale.

I test nel cloud sono utili per tutte le fasi del test, inclusi test unitari, test di integrazione e test. end-to-end

## Usa dei mock se necessario

I framework fittizi sono uno strumento prezioso per scrivere test unitari rapidi. Questi sono particolarmente utili quando i test riguardano logiche aziendali interne complesse, come calcoli o simulazioni matematiche o finanziarie. Cerca test unitari che prevedono un gran numero di casi di test o varianti di input, in cui tali input non modificano lo schema o il contenuto delle chiamate ad altri servizi cloud. La creazione di test fittizi per questi scenari può migliorare i tempi di iterazione degli sviluppatori.

Il codice verificato dai test unitari che usano i test fittizi dovrebbe essere verificato anche dai test nel cloud. Ciò è necessario perché i mock sono ancora in esecuzione sul laptop o sulla macchina di compilazione di uno sviluppatore e l'ambiente potrebbe essere configurato in modo diverso rispetto a quello nel cloud. Ad esempio, il codice potrebbe includere AWS Lambda funzioni che utilizzano più memoria o richiedono più tempo di quello che Lambda è configurato per allocare quando viene eseguito con determinati parametri di input. Oppure il codice potrebbe includere variabili di ambiente che non sono configurate nello stesso modo (o non sono affatto configurate) e le differenze potrebbero causare un comportamento diverso o un errore del codice.

Non utilizzare simulazioni di servizi cloud per convalidare la corretta implementazione di tali integrazioni di servizi. Sebbene possa essere accettabile simulare un servizio cloud quando si testano altre funzionalità, è consigliabile testare le chiamate ai servizi cloud nel cloud per convalidare la configurazione e l'implementazione funzionale corrette.

I mock possono aggiungere valore ai test unitari, soprattutto quando si testano frequentemente un gran numero di casi. Questo vantaggio è ridotto per i test di integrazione, poiché il livello di impegno necessario per implementare i mock necessari aumenta con il numero di punti di connessione. End-to-end test non dovrebbero utilizzare simulazioni, poiché generalmente si occupano di stati e logiche complesse che non possono essere facilmente simulate con framework fittizi.

## Comprendi i compromessi dei test di emulazione

Gli emulatori possono essere una scelta pratica per casi d'uso specifici. Ad esempio, un team di sviluppo con accesso a Internet limitato, incoerente o lento potrebbe scoprire che i test di emulazione sono il modo più affidabile per iterare sul codice prima di passare a un ambiente cloud.

Nella maggior parte delle altre circostanze, utilizza gli emulatori in modo selettivo. Quando fai molto affidamento su un emulatore, può diventare difficile incorporare nuove funzionalità di AWS servizio nei test fino a quando il fornitore dell'emulazione non rilascia un aggiornamento che garantisca la parità di funzionalità. Gli emulatori richiedono inoltre investimenti iniziali e continui per l'installazione e la configurazione dei sistemi di sviluppo e delle macchine di costruzione. Inoltre, molti servizi cloud non dispongono di emulatori; la scelta di una strategia incentrata sull'emulazione può precludere l'uso di tali servizi o produrre codice e configurazioni che non sono ben testati rispetto al comportamento reale dei servizi.

Se utilizzi i test di emulazione, completali il più possibile con i test sul cloud per verificare che siano presenti configurazioni cloud corrette e per testare le interazioni con servizi che possono essere simulati o simulati solo in un ambiente emulato.

I test di emulazione possono fornire un feedback rapido per i test unitari e, a seconda delle caratteristiche e della parità comportamentale del software di emulazione, possono supportare anche alcune integrazioni e test. end-to-end

## Amplia i test attraverso i confini naturali

Man mano che le applicazioni serverless si espandono su più componenti architettonici, emergono confini naturali attorno ai sottosistemi, specialmente quando si seguono le migliori pratiche come le funzioni monouso e il disaccoppiamento basato sugli eventi. Questi limiti fungono da punti di prova efficaci in cui è possibile convalidare i contratti tra i componenti.

### Identifica i confini dell'architettura

Cerca cuciture naturali nel design della tua applicazione:

- Tra servizi, ad esempio una EventBridge regola Amazon che collega un editore a un consumatore
- Ai margini delle API, come gli endpoint Amazon API Gateway che fronteggiano le funzioni Lambda
- Riguardo ai flussi di lavoro, come l'orchestrazione di più servizi AWS Step Functions
- A livelli di storage, come i flussi di Amazon DynamoDB che attivano l'elaborazione a valle

### Separare il codice Lambda dalla logica aziendale

Semplifica i test isolando il codice Lambda dalla logica aziendale principale. Il gestore Lambda dovrebbe fungere da sottile adattatore tra il AWS runtime e la logica dell'applicazione. Dovrebbe estrarre e convalidare i dati degli eventi e quindi delegarli a una funzione testabile che non ha dipendenze Lambda. Ciò rende la logica aziendale portabile, più facile da ragionare e semplice da testare senza simulare oggetti Lambda o configurare ambienti complessi.

### Tratta i confini come contratti

Esegui i test al confine, non attraverso di esso. Convalida ciò che attraversa il limite senza richiedere l'intero sistema a valle. Questi stessi limiti fungono anche da ganci di osservabilità nella produzione. Le giunture architettoniche in cui esegui i test possono essere strumentate per il monitoraggio utilizzando Amazon CloudWatch Logs, AWS X-Ray trace ed eventi. EventBridge

## Utilizza test harness per flussi di lavoro asincroni

Le applicazioni serverless si basano spesso su modelli asincroni, in cui gli eventi attivano l'elaborazione, i messaggi scorrono attraverso le code e i flussi di lavoro si estendono su più servizi senza risposte immediate. Non puoi semplicemente richiamare una funzione e controllare un valore restituito. Il risultato può apparire successivamente in un database, in un flusso di log o in un altro servizio.

Un test harness è un'infrastruttura di test implementata insieme all'applicazione per osservare e convalidare questo comportamento asincrono. I test harness in genere includono:

- Listener di eventi che si iscrivono agli stessi eventi prodotti dall'applicazione
- Meccanismi di storage (come tabelle DynamoDB o bucket Amazon S3) in cui è possibile acquisire i risultati dei test
- Logica di polling nel codice di test che attende la visualizzazione dei risultati previsti

Il codice di test avvia un evento, attende il completamento del flusso di lavoro, quindi interroga il test harness per verificare che si sia verificato il risultato previsto.

Di seguito sono riportate le best practice:

- Definisci in modo chiaro SLAs le operazioni asincrone: stabilisci la durata dei flussi di lavoro e usali come timeout di polling nei test
- Usa identificatori univoci per l'isolamento dei test: genera nomi di file, messaggi o token di correlazione univoci per ogni esecuzione del test per evitare IDs interferenze tra i test
- Implementa l'infrastruttura di test insieme alla tua applicazione: includi risorse di test harness nei tuoi infrastructure-as-code modelli in modo che rimangano sincronizzati man mano che l'applicazione si evolve
- Pulisci i dati dei test dopo l'esecuzione dei test: in questo modo si evita l'accumulo di artefatti di test nel tuo ambiente cloud

I test harness sono particolarmente utili per i test di integrazione che convalidano i flussi di lavoro su più servizi, i end-to-end test che verificano i percorsi completi degli utenti e le architetture basate sugli eventi in cui i servizi comunicano attraverso Amazon EventBridge SNS, Amazon SQS o Amazon Kinesis.

## Organizza ambienti cloud per l'isolamento degli sviluppatori

I test nel cloud richiedono ambienti isolati l'uno dall'altro. Quando gli sviluppatori condividono un unico account Account AWS, ad esempio un account di sviluppo in team, prendi in considerazione la possibilità di creare uno stack di applicazioni separato per ogni sviluppatore o ramo di funzionalità. In questo modo si isolano le risorse, si evitano le collisioni tra i nomi e si evitano contese tra le quote o problemi legati alla rumorosità dei vicini durante i test.

Utilizza AWS Systems Manager Parameter Store o strumenti simili per gestire configurazioni specifiche dello stack, come gli endpoint delle API e i nomi delle code. Per ridurre i costi, condividi risorse costose come i cluster Amazon Relational Database Service (Amazon RDS) tra stack di sviluppatori, mantenendo al contempo le risorse serverless leggere (come funzioni Lambda, fasi API Gateway e tabelle DynamoDB) isolate per stack.

Nei settori regolamentati, le politiche di sicurezza aziendali possono limitare l'accesso degli sviluppatori agli ambienti cloud, rendendo difficile l'esecuzione di test sul cloud come parte di un flusso di lavoro di sviluppo locale. In questi casi, i test di emulazione possono colmare il divario tra i test simulati locali e la convalida completa del cloud, anche se dovrebbero essere integrati dai test sul cloud ogni volta che l'accesso lo consente.

## Accelera i cicli di feedback

Quando esegui i test nel cloud, utilizza strumenti e tecniche per accelerare i cicli di feedback dello sviluppo. Ad esempio, utilizza [AWS SAM AWS CDK Accelerate e la](#) modalità watch per ridurre il tempo necessario per apportare modifiche al codice in un ambiente cloud. Gli esempi presenti nel [repository GitHub Serverless Test Samples](#) esplorano alcune di queste tecniche.

Ti consigliamo inoltre di creare e testare le risorse cloud dal tuo computer locale il prima possibile durante lo sviluppo, non solo dopo il check-in per il controllo del codice sorgente. Questa pratica consente di accelerare l'esplorazione e la sperimentazione durante lo sviluppo di soluzioni. Inoltre, la capacità di automatizzare l'implementazione da una macchina di sviluppo ti aiuta a scoprire i problemi di configurazione del cloud più rapidamente e riduce gli sprechi di tempo per gli aggiornamenti e l'approvazione delle modifiche al controllo del codice sorgente.

## Domande frequenti

Ho una funzione Lambda che esegue calcoli e restituisce un risultato senza chiamare altri servizi. Devo davvero testarla nel cloud?

Sì. AWS Lambda le funzioni hanno parametri di configurazione che potrebbero modificare l'esito del test. Tutto il codice della funzione Lambda dipende dal [timeout e dalle impostazioni di memoria](#), che potrebbero causare il fallimento della funzione se non sono impostate correttamente. [Le policy Lambda consentono anche la registrazione standard dell'output su Amazon CloudWatch](#). Anche se il codice non chiama CloudWatch direttamente, è necessaria un'autorizzazione per abilitare la registrazione e tale autorizzazione non può essere simulata o emulata con precisione.

In che modo i test nel cloud possono contribuire ai test unitari?  
Se è nel cloud e si connette ad altre risorse, non è un test di integrazione?

Definiamo test unitari come test che operano isolatamente su componenti dell'architettura. Questa definizione non preclude necessariamente l'uso di chiamate di servizio o altre comunicazioni di rete.

Molte applicazioni serverless dispongono di componenti dell'architettura che possono essere testati in modo isolato, anche nel cloud. Un esempio di base è una funzione Lambda che accetta alcuni input, li interpreta e invia un messaggio a una coda Amazon Simple Queue Service (Amazon SQS). Un test unitario di questa funzione verificherebbe probabilmente se i valori di input determinano la presenza di determinati valori nel messaggio in coda. Prendiamo in considerazione un test scritto utilizzando lo schema `arrange, act, assert`:

- **Arrange:** alloca le risorse (una coda per ricevere messaggi e la funzione in esame).
- **Agisci:** richiama la funzione in fase di test.
- **Assert:** recupera il messaggio inviato dalla funzione e convalida l'output.

Un approccio di test con mock implicherebbe la simulazione della coda con un oggetto fittizio in corso di elaborazione e la creazione di un'istanza durante il processo della classe o del modulo

che contiene il codice della funzione Lambda. Durante la fase di asserzione, il messaggio in coda verrebbe recuperato dall'oggetto simulato.

In un approccio basato sul cloud, il test creerebbe una coda Amazon SQS ai fini del test e implementerebbe la funzione Lambda con variabili di ambiente configurate per utilizzare la coda Amazon SQS isolata come destinazione di output. Dopo aver eseguito la funzione Lambda, il test recupererebbe il messaggio dalla coda di Amazon SQS.

Il test basato sul cloud eseguirebbe lo stesso codice, affermerebbe lo stesso comportamento e convaliderebbe la correttezza funzionale dell'applicazione. Tuttavia, avrebbe l'ulteriore vantaggio di poter convalidare le seguenti impostazioni della funzione Lambda: AWS Identity and Access Management il ruolo (IAM), le politiche IAM e le impostazioni di timeout e memoria della funzione.

## Risorse e passaggi successivi

Utilizza le seguenti risorse per ulteriori letture e altri esempi concreti.

## Implementazioni di esempio

Il [repository Serverless Test Samples](#) GitHub contiene esempi concreti di test che seguono i modelli e le best practice descritti in questa guida. Il repository contiene codice di esempio e procedure dettagliate dei processi di test con mock, emulazione e cloud descritti nelle sezioni precedenti. Usa questo repository per aggiornarti sulle ultime linee guida per i test serverless di AWS.

## Approfondimenti

Visita [Serverless Land](#) per accedere ai blog, ai video e ai corsi di formazione più recenti sulle tecnologie serverless. AWS

## Riferimenti

- [Accelerazione dello sviluppo serverless con AWS SAM](#) Accelerate (post sul blog)AWS
- [Aumentare la velocità di sviluppo con CDK Watch](#) (AWS post sul blog)
- [Simulazione delle integrazioni di servizi con AWS Step Functions Local](#) (post sul blog)AWS
- [Guida introduttiva al test di applicazioni serverless](#) (post sul blog)AWS
- [Accelera i test serverless con LocalStack l'integrazione in VS Code IDE](#) (AWS post sul blog)
- Esegui il [debug locale delle funzioni con AWS SAM\(documentazione\)](#)AWS

## Tools (Strumenti)

- AWS SAM — [Test e debug](#) di applicazioni serverless
- AWS SAM — [Integrazione con test automatici](#)
- AWS Lambda — [Test delle funzioni Lambda nella console](#)
- LocalStack Cloud Emulator: [migliora l'esperienza di test locale per le applicazioni serverless](#) con LocalStack

# Collaboratori

## Creazione

- Dan Fox, AWS
- Leslie Raj, AWS
- Rohan Mehta, AWS
- Rob Hill, AWS

## Revisione

- Brian Krygsman, AWS

## Scrittura tecnica

- Lilly AbouHarb, AWS

# Cronologia dei documenti

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

Modifica	Descrizione	Data
<a href="#">Aggiornamenti</a>	Abbiamo aggiunto linee guida sui test ai limiti dell'architettura e del codice, sui test harness per i flussi di lavoro asincroni e sull'isolamento dell'ambiente di sviluppo. Abbiamo anche aggiornato i consigli sui test di emulazione.	18 marzo 2026
<a href="#">Pubblicazione iniziale</a>	—	9 dicembre 2022

# 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 riproduzione 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 epopee 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 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.

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

## metadati di migrazione

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

## modello di migrazione

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

## Valutazione del portfolio di migrazione (MPA)

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

## valutazione della preparazione alla migrazione (MRA)

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

## strategia di migrazione

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

## ML

[Vedi machine learning.](#)

## modernizzazione

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

## valutazione della preparazione alla modernizzazione

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

## applicazioni monolitiche (monoliti)

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

## MAPPA

Vedi [Migration Portfolio Assessment](#).

## MQTT

Vedi [Message Queuing Telemetry Transport](#).

## classificazione multiclasse

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

## infrastruttura mutabile

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

## O

### OAC

Vedi [Origin Access Control](#).

### QUERCIA

Vedi [Origin Access Identity](#).

### OCM

Vedi [gestione delle modifiche organizzative](#).

## migrazione offline

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

## OI

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

## OLA

Vedi accordo a [livello operativo](#).

## migrazione online

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

## OPC-UA

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

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

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

## accordo a livello operativo (OLA)

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

## revisione della prontezza operativa (ORR)

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

## tecnologia operativa (OT)

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

## integrazione delle operazioni (OI)

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

## trail organizzativo

Un percorso creato da noi AWS CloudTrail che registra tutti gli eventi di un'organizzazione per tutti Account AWS . AWS Organizations Questo percorso viene creato in ogni Account AWS che

fa parte dell'organizzazione e tiene traccia dell'attività in ogni account. Per ulteriori informazioni, consulta [Creazione di un percorso per un'organizzazione](#) nella CloudTrail documentazione.

#### gestione del cambiamento organizzativo (OCM)

Un framework per la gestione di trasformazioni aziendali importanti e che comportano l'interruzione delle attività dal punto di vista delle persone, della cultura e della leadership. OCM aiuta le organizzazioni a prepararsi e passare a nuovi sistemi e strategie accelerando l'adozione del cambiamento, affrontando i problemi di transizione e promuovendo cambiamenti culturali e organizzativi. Nella strategia di AWS migrazione, questo framework si chiama accelerazione delle persone, a causa della velocità di cambiamento richiesta nei progetti di adozione del cloud. Per ulteriori informazioni, consultare la [Guida OCM](#).

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

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

#### identità di accesso origine (OAI)

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

#### ORR

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

#### NON

Vedi la [tecnologia operativa](#).

#### VPC in uscita (egress)

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

# P

## limite delle autorizzazioni

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

## informazioni di identificazione personale (PII)

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

## Informazioni che consentono l'identificazione personale degli utenti

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

## playbook

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

## PLC

Vedi [controllore logico programmabile](#).

## PLM

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

## policy

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

## persistenza poliglotta

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

## valutazione del portfolio

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

## predicate

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

## predicato pushdown

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

## controllo preventivo

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

## principale

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

## privacy fin dalla progettazione

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

## zone ospitate private

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

## controllo proattivo

Un [controllo di sicurezza](#) progettato per impedire l'implementazione di risorse non conformi. Questi controlli analizzano le risorse prima del loro provisioning. Se la risorsa non è conforme al

controllo, non viene fornita. Per ulteriori informazioni, consulta la [guida di riferimento sui controlli](#) nella AWS Control Tower documentazione e consulta Controlli [proattivi in Implementazione dei controlli](#) di sicurezza su. AWS

gestione del ciclo di vita del prodotto (PLM)

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

Ambiente di produzione

[Vedi ambiente.](#)

controllore logico programmabile (PLC)

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

concatenamento rapido

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

pseudonimizzazione

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

publish/subscribe (pub/sub)

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

## Q

Piano di query

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

## regressione del piano di query

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

## R

### Matrice RACI

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

### RAG

Vedi [Retrieval](#) Augmented Generation.

### ransomware

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

### Matrice RASCI

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

### RCAC

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

### replica di lettura

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

### riprogettare

Vedi [7 Rs](#).

### obiettivo del punto di ripristino (RPO)

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

## obiettivo del tempo di ripristino (RTO)

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

## rifattorizzare

Vedi [7 R.](#)

## Region

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

## regressione

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

## riospitare

Vedi [7 R.](#)

## rilascio

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

## trasferisco

Vedi [7 Rs.](#)

## ripiattaforma

Vedi [7 Rs.](#)

## riacquisto

Vedi [7 Rs.](#)

## resilienza

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

## policy basata su risorse

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

## matrice di assegnazione di responsabilità (RACI)

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

## controllo reattivo

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

## retain

Vedi [7 R](#).

## andare in pensione

Vedi [7 Rs](#).

## Retrieval Augmented Generation (RAG)

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

## rotazione

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

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

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

## RPO

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

## VERSO

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

## runbook

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

## S

### SAML 2.0

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

### SCADA

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

### SCP

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

### Secret

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

### sicurezza fin dalla progettazione

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

## controllo di sicurezza

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

## rafforzamento della sicurezza

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

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

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

## automazione della risposta alla sicurezza

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

## Crittografia lato server

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

## Policy di controllo dei servizi (SCP)

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

## endpoint del servizio

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

## accordo sul livello di servizio (SLA)

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

## indicatore del livello di servizio (SLI)

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

## obiettivo a livello di servizio (SLO)

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

## Modello di responsabilità condivisa

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

## SIEM

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

## punto di errore singolo (SPOF)

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

## SLAM

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

## SLI

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

## LENTA

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

## split-and-seed modello

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

## SPOF

Vedi [punto di errore singolo](#).

## schema a stella

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

## modello del fico strangolatore

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

## sottorete

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

## controllo di supervisione e acquisizione dati (SCADA)

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

## crittografia simmetrica

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

## test sintetici

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

## prompt di sistema

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

# T

## tag

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

## variabile di destinazione

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

## elenco di attività

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

## ambiente di test

[Vedi ambiente.](#)

## training

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

## Transit Gateway

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

### flusso di lavoro basato su trunk

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

### Accesso attendibile

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

### regolazione

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

### team da due pizze

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

## U

### incertezza

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

## 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.