



Implementazione di PostgreSQL gestito per applicazioni SaaS multi-tenant su AWS

AWS Guida prescrittiva



AWS Guida prescrittiva: Implementazione di PostgreSQL gestito per applicazioni SaaS multi-tenant su AWS

Copyright © 2025 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 delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Introduzione	1
Obiettivi aziendali specifici	1
Selezione di un database per un'applicazione SaaS	3
Scelta tra Amazon RDS e Aurora	5
Modelli di partizionamento SaaS multi-tenant per PostgreSQL	7
Modello di silo PostgreSQL	8
Modello di pool PostgreSQL	9
Modello bridge PostgreSQL	11
Matrice decisionale	13
Raccomandazioni di sicurezza a livello di riga	29
Disponibilità di PostgreSQL per il modello pool	31
Best practice	33
Confronta AWS le opzioni per PostgreSQL gestito	33
Seleziona un modello di partizionamento SaaS multi-tenant	33
Usa la sicurezza a livello di riga per i modelli di partizionamento SaaS del pool	33
Domande frequenti	34
Quali opzioni PostgreSQL gestite offre? AWS	34
Qual è il servizio ottimale per le applicazioni SaaS?	34
Quali requisiti unici devo prendere in considerazione se decido di utilizzare un database PostgreSQL con un'applicazione SaaS multi-tenant?	34
Quali modelli posso usare per mantenere l'isolamento dei dati dei tenant con PostgreSQL?	34
Come posso mantenere l'isolamento dei dati dei tenant con un singolo database PostgreSQL condiviso tra più tenant?	35
Passaggi successivi	36
Risorse	37
Riferimenti	37
Partner	37
Cronologia dei documenti	38
Glossario	39
#	39
A	40
B	43
C	45
D	48

E	52
F	54
G	56
H	57
I	59
L	61
M	62
O	67
P	69
Q	72
R	73
S	76
T	80
U	81
V	82
W	82
Z	84
.....	lxxxv

Implementazione di PostgreSQL gestito per applicazioni SaaS multi-tenant su AWS

Tabby Ward e Thomas Davis, Amazon Web Services (AWS)

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

Quando si seleziona un database per archiviare i dati operativi, è fondamentale considerare come devono essere strutturati i dati, a quali domande risponderanno, quanto velocemente forniranno le risposte e la resilienza della piattaforma dati stessa. Oltre a queste considerazioni generali, vi sono le implicazioni del software as a service (SaaS) per i dati operativi, come l'isolamento delle prestazioni, la sicurezza dei tenant e le caratteristiche uniche e i modelli di progettazione tipici dei dati per le applicazioni SaaS multi-tenant. Questa guida illustra come questi fattori si applicano all'utilizzo di un database PostgreSQL su Amazon Web Services (AWS) come archivio dati operativo principale per un'applicazione SaaS multi-tenant. In particolare, la guida si concentra su due opzioni PostgreSQL AWS gestite: Amazon Aurora PostgreSQL Compatible Edition e Amazon Relational Database Service (Amazon RDS) per PostgreSQL.

Obiettivi aziendali specifici

Questa guida fornisce un'analisi dettagliata delle best practice per le applicazioni SaaS multi-tenant che utilizzano Aurora, compatibile con PostgreSQL e Amazon RDS for PostgreSQL. Ti consigliamo di utilizzare i modelli e i concetti di progettazione forniti in questa guida per informare e standardizzare l'implementazione di Aurora compatibile con PostgreSQL o Amazon RDS for PostgreSQL per le tue applicazioni SaaS multi-tenant.

Questa guida prescrittiva aiuta a raggiungere i seguenti risultati aziendali:

- Scelta dell'opzione PostgreSQL AWS gestita più ottimale per il tuo caso d'uso: questa guida confronta le opzioni relazionali e non relazionali per l'utilizzo del database con le applicazioni SaaS. Descrive inoltre quali sono i casi d'uso più ottimali per Aurora, compatibile con PostgreSQL e Amazon RDS for PostgreSQL. Queste informazioni ti aiuteranno a selezionare l'opzione migliore per la tua applicazione SaaS.
- Applicazione delle migliori pratiche SaaS attraverso l'adozione di un modello di partizionamento SaaS: questa guida illustra e confronta tre ampi modelli di partizionamento SaaS applicabili a un sistema di gestione di database PostgreSQL (DBMS): modelli pool, bridged e silo e le relative

varianti. Questi approcci acquisiscono le migliori pratiche SaaS e offrono flessibilità durante la progettazione di un'applicazione SaaS. L'applicazione di un modello di partizionamento SaaS è una parte cruciale per preservare le migliori pratiche.

- Uso efficace di RLS nei modelli di partizionamento SaaS dei pool: la sicurezza a livello di riga (RLS) supporta l'applicazione dell'isolamento dei dati dei tenant all'interno di una singola tabella PostgreSQL limitando le righe che possono essere visualizzate in base all'utente o a una variabile di contesto. Quando si utilizza il modello di partizionamento del pool, RLS è necessario per impedire l'accesso tra tenant.

Selezione di un database per un'applicazione SaaS

Per molte applicazioni SaaS multi-tenant, la selezione di un database operativo può essere sintetizzata in una scelta tra database relazionali e non relazionali o una combinazione dei due. Per prendere una decisione, prendete in considerazione questi requisiti e caratteristiche di alto livello relativi ai dati delle applicazioni:

- Modello di dati dell'applicazione
- Schemi di accesso per i dati
- Requisiti di latenza del database
- Requisiti di integrità dei dati e integrità transazionale (atomicità, coerenza, isolamento e durabilità o ACID)
- Requisiti di disponibilità e ripristino in più regioni

La tabella seguente elenca i requisiti e le caratteristiche dei dati delle applicazioni e li descrive nel contesto delle offerte di AWS database: compatibile con Aurora PostgreSQL e Amazon RDS for PostgreSQL (relazionale) e Amazon DynamoDB (non relazionale). Puoi fare riferimento a questa matrice quando cerchi di decidere tra offerte di database operativi relazionali e non relazionali.

Database	Requisiti e caratteristiche dei dati delle applicazioni SaaS				
	Modello di dati	Modelli di accesso	Requisiti di latenza	Integrità dei dati e delle transazioni	Disponibilità e ripristino in più regioni
Relazionale (Compatibile con Aurora PostgreSQL e Amazon RDS per PostgreSQL)	Relazionale o altamente normalizzato.	Non deve essere pianificato a fondo in anticipo.	Preferibilmente una tolleranza di latenza più elevata; può ottenere latenze inferiori per impostazione predefinita	Elevata integrità transazionale e dei dati mantenuta per impostazione predefinita.	In Amazon RDS, puoi creare una replica di lettura per il ridimensionamento e il failover tra regioni. Aurora

con Aurora e implementando repliche di lettura, memorizzazione nella cache e funzionalità simili.

[automatizza principalmente](#) questo processo. [Per configurazioni attive-attive su più piattaforme Regioni AWS, puoi utilizzare l'inoltro di scrittura insieme ai database globali Aurora.](#)

Non relazionale (Amazon DynamoDB)	Di solito denormalizzato. Questi database sfruttano i modelli per modellare many-to-many relazioni, elementi di grandi dimensioni e dati di serie temporali.	Tutti i modelli di accesso (query) per i dati devono essere compresi a fondo prima di creare un modello di dati.	Latenza molto bassa con opzioni come Amazon DynamoDB Accelerator (DAX) in grado di migliorare ulteriormente le prestazioni.	Integrità transazionale opzionale a scapito delle prestazioni. Le preoccupazioni relative all'integrità dei dati vengono trasferite e all'applicazione.	Semplice ripristino tra regioni e configurazione attiva-attiva con tabelle globali. (La conformità ACID è ottenibile solo in una singola regione). AWS
--	--	--	---	---	---

Alcune applicazioni SaaS multi-tenant potrebbero avere modelli di dati unici o circostanze speciali che sono meglio servite da database non inclusi nella tabella precedente. Ad esempio, set di dati di serie temporali, set di dati altamente connessi o la gestione di un registro delle transazioni centralizzato potrebbero richiedere l'utilizzo di un diverso tipo di database. L'analisi di tutte le

possibilità esula dallo scopo di questa guida. Per un elenco completo delle offerte di AWS database e di come possono soddisfare diversi casi d'uso ad alto livello, consulta la sezione [Database](#) del white paper Overview of Amazon Web Services.

Il resto di questa guida si concentra sui servizi di database AWS relazionali che supportano PostgreSQL: compatibile con Amazon RDS e Aurora PostgreSQL. DynamoDB richiede un approccio diverso per l'ottimizzazione delle applicazioni SaaS, che esula dallo scopo di questa guida. Per ulteriori informazioni su DynamoDB, consulta AWS il [post di blog Partizionare dati SaaS multi-tenant in pool](#) con Amazon DynamoDB.

Scelta tra Amazon RDS e Aurora

Nella maggior parte dei casi, consigliamo di utilizzare Aurora PostgreSQL compatibile con Amazon RDS for PostgreSQL. La tabella seguente mostra i fattori da considerare al momento di decidere tra queste due opzioni.

Componente DBMS	Amazon RDS per PostgreSQL	Compatibile con Aurora PostgreSQL
Scalabilità	Ritardo di replica di minuti, massimo 5 repliche di lettura	Ritardo di replica inferiore a un minuto (in genere meno di 1 secondo con database globali), massimo 15 repliche di lettura
Ripristino da un crash	I checkpoint a distanza di 5 minuti (per impostazione predefinita) possono rallentare le prestazioni del database	Ripristino asincrono con thread paralleli per un ripristino rapido
Failover	60-120 secondi in aggiunta al tempo di ripristino in caso di incidente	Di solito circa 30 secondi (incluso il ripristino in caso di incidente)
Storage	IOPS massimo di 256.000	IOPS vincolato solo dalle dimensioni e dalla capacità dell'istanza Aurora

Componente DBMS	Amazon RDS per PostgreSQL	Compatibile con Aurora PostgreSQL
Alta disponibilità e disaster recovery	Due zone di disponibilità con un'istanza in standby, failover interregionale per leggere le repliche o i backup copiati	Tre zone di disponibilità per impostazione predefinita, failover tra regioni con database globali Aurora, inoltre di scrittura per configurazioni attive-attive Regioni AWS
Backup	Durante la finestra di backup, può influire sulle prestazioni	Backup incrementali automatici, nessun impatto sulle prestazioni
Classi di istanze di database	Visualizza l'elenco delle classi di istanze Amazon RDS	Vedi l'elenco delle classi di istanze Aurora

In tutte le categorie descritte nella tabella precedente, Aurora PostgreSQL Compatible è in genere l'opzione migliore. Tuttavia, Amazon RDS for PostgreSQL potrebbe ancora essere utile per carichi di lavoro di piccole e medie dimensioni, perché offre una selezione più ampia di classi di istanze che potrebbero fornire un'opzione più conveniente a scapito del set di funzionalità più robusto di Aurora.

Modelli di partizionamento SaaS multi-tenant per PostgreSQL

Il metodo migliore per realizzare la multi-tenancy dipende dai requisiti dell'applicazione SaaS. Le sezioni seguenti illustrano i modelli di partizionamento per implementare con successo la multi-tenancy in PostgreSQL.

Note

I modelli discussi in questa sezione sono applicabili sia ad Amazon RDS for PostgreSQL che a Aurora PostgreSQL. I riferimenti a PostgreSQL in questa sezione si applicano a entrambi i servizi.

Esistono tre modelli di alto livello che è possibile utilizzare in PostgreSQL per il partizionamento SaaS: silo, bridge e pool. L'immagine seguente riassume i compromessi tra i modelli silo e pool. Il modello a ponte è un ibrido tra i modelli silo e pool.

Modello di partizionamento	Vantaggi	Svantaggi
Silo	<ul style="list-style-type: none"> • Allineamento alla conformità • Nessun impatto tra i tenant • Ottimizzazione a livello di tenant • Disponibilità a livello di tenant 	<ul style="list-style-type: none"> • Agilità compromessa • Nessuna gestione centralizzata • Complessità di implementazione • Costo
Pool	<ul style="list-style-type: none"> • Agilità • Ottimizzazione dei costi • Gestione centralizzata • Implementazione semplificata 	<ul style="list-style-type: none"> • Impatto tra i tenant • Sfide di conformità • Disponibilità «tutto o niente»
Ponte	<ul style="list-style-type: none"> • Qualche allineamento alla conformità 	<ul style="list-style-type: none"> • Alcune sfide di conformità

Modello di partizionamento	Vantaggi	Svantaggi
	<ul style="list-style-type: none">• Agilità• Ottimizzazione dei costi• Gestione centralizzata	<ul style="list-style-type: none">• Disponibilità totale o nulla (principalmente)• Impatto tra i tenant• Complessità di implementazione

Le sezioni seguenti illustrano ciascun modello in modo più dettagliato.

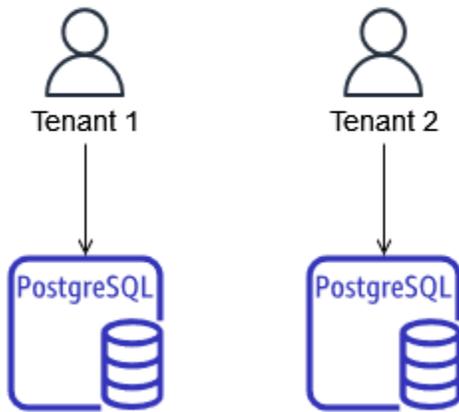
Modelli di partizionamento:

- [Modello di silo PostgreSQL](#)
- [Modello di pool PostgreSQL](#)
- [Modello bridge PostgreSQL](#)
- [Matrice decisionale](#)

Modello di silo PostgreSQL

Il modello silo viene implementato fornendo un'istanza PostgreSQL per ogni tenant di un'applicazione. Il modello silo eccelle in termini di prestazioni dei tenant e isolamento della sicurezza ed elimina completamente il fenomeno dei rumorosi vicini. Il fenomeno Noisy Neighbor si verifica quando l'utilizzo di un sistema da parte di un inquilino influisce sulle prestazioni di un altro inquilino. Il modello a silo consente di personalizzare le prestazioni in modo specifico per ciascun tenant e di limitare potenzialmente le interruzioni al silo di un tenant specifico. Tuttavia, ciò che generalmente guida l'adozione di un modello a silo sono i rigorosi vincoli normativi e di sicurezza. Questi vincoli possono essere motivati dai clienti SaaS. Ad esempio, i clienti SaaS potrebbero richiedere l'isolamento dei propri dati a causa di vincoli interni e i provider SaaS potrebbero offrire tale servizio a un costo aggiuntivo.

Silo model (separate PostgreSQL instances or clusters for each tenant)

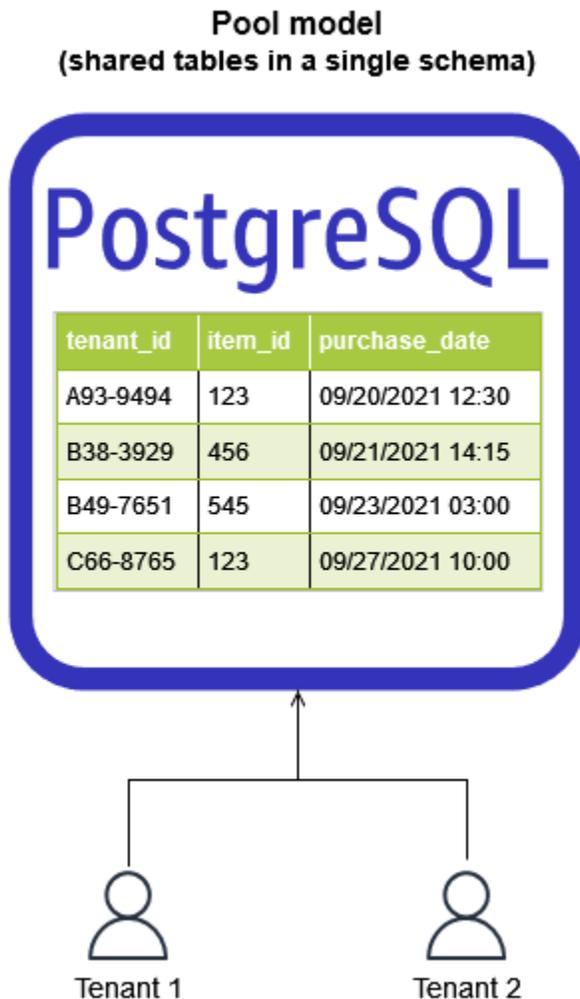


Sebbene il modello a silo possa essere necessario in alcuni casi, presenta molti inconvenienti. Spesso è difficile utilizzare il modello a silo in modo conveniente, perché la gestione del consumo di risorse su più istanze PostgreSQL può essere complicata. Inoltre, la natura distribuita dei carichi di lavoro dei database in questo modello rende più difficile mantenere una visione centralizzata dell'attività dei tenant. La gestione di così tanti carichi di lavoro gestiti in modo indipendente aumenta il sovraccarico operativo e amministrativo. Il modello a silo rende inoltre l'onboarding dei tenant più complicato e dispendioso in termini di tempo, poiché è necessario fornire risorse specifiche per il tenant. Inoltre, l'intero sistema SaaS può essere più difficile da scalare, perché il numero sempre crescente di istanze PostgreSQL specifiche per tenant richiederà più tempo operativo per l'amministrazione. Un'ultima considerazione è che un'applicazione o un livello di accesso ai dati dovrà mantenere una mappatura dei tenant alle istanze PostgreSQL associate, il che aumenta la complessità dell'implementazione di questo modello.

Modello di pool PostgreSQL

Il modello pool viene implementato effettuando il provisioning di una singola istanza PostgreSQL (Amazon RDS o Aurora) [e utilizzando la sicurezza a livello di riga](#) (RLS) per mantenere l'isolamento dei dati dei tenant. Le politiche RLS limitano quali righe di una tabella vengono restituite dalle SELECT query o quali righe sono interessate da e comandi. INSERT UPDATE DELETE Il modello di pool centralizza tutti i dati dei tenant in un unico schema PostgreSQL, quindi è significativamente più conveniente e richiede meno sovraccarico operativo per la manutenzione. Il monitoraggio di questa soluzione è inoltre notevolmente più semplice grazie alla sua centralizzazione. Tuttavia, il monitoraggio degli impatti specifici dei tenant nel modello di pool richiede in genere una

strumentazione aggiuntiva nell'applicazione. Questo perché PostgreSQL per impostazione predefinita non è a conoscenza di quale tenant stia consumando risorse. L'onboarding dei tenant è semplificato perché non è richiesta alcuna nuova infrastruttura. Questa agilità semplifica la realizzazione di flussi di lavoro di onboarding dei tenant rapidi e automatizzati.



Sebbene il modello di pool sia generalmente più economico e più semplice da amministrare, presenta alcuni svantaggi. Il fenomeno dei vicini rumorosi non può essere completamente eliminato in un modello di piscina. Tuttavia, può essere mitigato assicurando che siano disponibili risorse appropriate sull'istanza PostgreSQL e utilizzando strategie per ridurre il carico in PostgreSQL, come l'offloading delle query su repliche di lettura o su Amazon. ElastiCache Un monitoraggio efficace svolge anche un ruolo nel rispondere ai problemi di isolamento delle prestazioni dei tenant, poiché la strumentazione applicativa può registrare e monitorare le attività specifiche del tenant. Infine, alcuni clienti SaaS

potrebbero non ritenere sufficiente la separazione logica fornita da RLS e potrebbero richiedere ulteriori misure di isolamento.

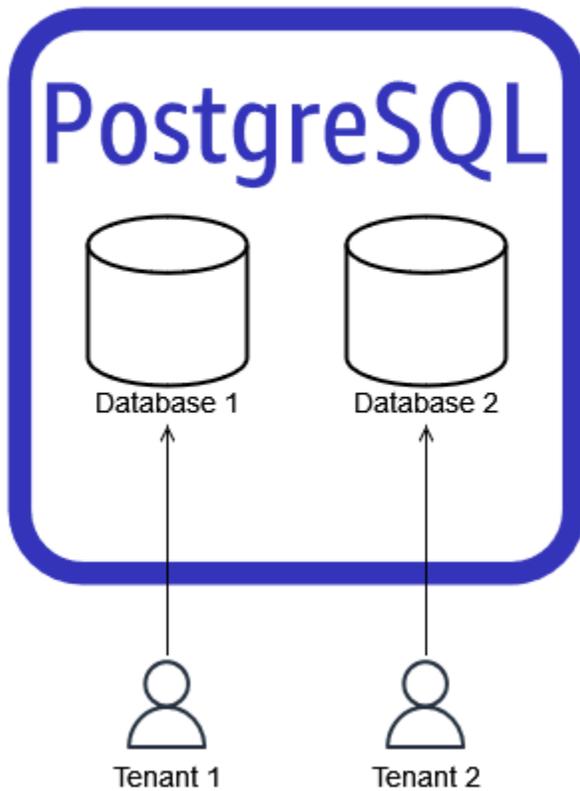
Modello bridge PostgreSQL

Il modello bridge PostgreSQL è una combinazione di approcci in pool e in silos. Analogamente al modello in pool, esegui il provisioning di una singola istanza PostgreSQL per ogni tenant. Per mantenere l'isolamento dei dati dei tenant, si utilizzano costrutti logici PostgreSQL. Nel diagramma seguente, i database PostgreSQL vengono utilizzati per separare logicamente i dati.

Note

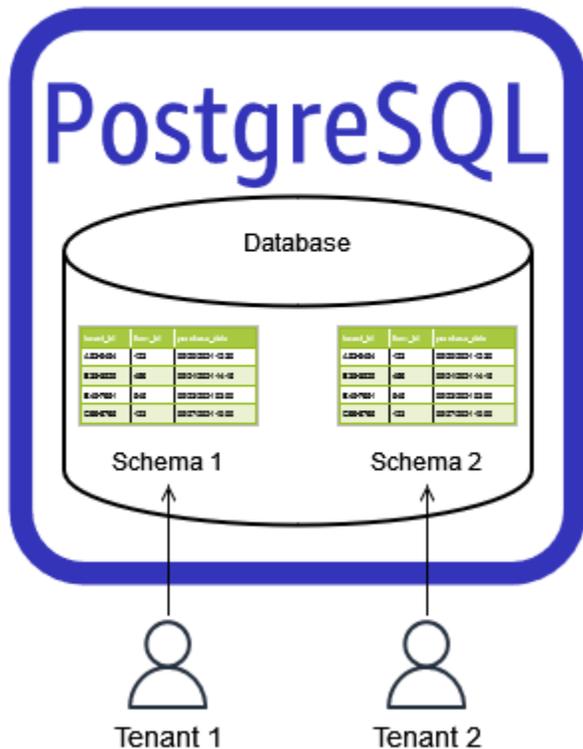
Un database PostgreSQL non fa riferimento a un'istanza DB separata compatibile con Amazon RDS for PostgreSQL o Aurora PostgreSQL. Si riferisce invece a un costrutto logico del sistema di gestione del database PostgreSQL per separare i dati.

Bridge model with separate databases (separate databases in a single instance)



È inoltre possibile implementare il modello bridge utilizzando un singolo database PostgreSQL, con schemi specifici del tenant in ogni database, come illustrato nel diagramma seguente.

Bridge model with separate schemas (separate schemas in a single database)



Il modello bridge presenta gli stessi problemi di isolamento delle prestazioni dei rumorosi vicini e dei tenant del modello pool. Inoltre, comporta un sovraccarico operativo e di provisioning aggiuntivo in quanto richiede il provisioning di database o schemi separati su base per-tenant. Richiede un monitoraggio efficace per rispondere rapidamente ai problemi relativi alle prestazioni degli inquilini. Richiede inoltre una strumentazione applicativa per monitorare l'utilizzo specifico del tenant. Nel complesso, il modello bridge può essere visto come un'alternativa a RLS che aumenta leggermente lo sforzo di onboarding dei tenant richiedendo nuovi database o schemi PostgreSQL. Come per il modello a silo, un'applicazione o un livello di accesso ai dati dovrà mantenere una mappatura dei tenant ai database o agli schemi PostgreSQL associati.

Matrice decisionale

Per decidere quale modello di partizionamento SaaS multi-tenant utilizzare con PostgreSQL, consulta la seguente matrice decisionale. La matrice analizza queste quattro opzioni di partizionamento:

- Silo: un'istanza o un cluster PostgreSQL separato per ogni tenant.

- Bridge con database separati: un database separato per ogni tenant in una singola istanza o cluster PostgreSQL.
- Bridge con schemi separati: uno schema separato per ogni tenant in un singolo database PostgreSQL, in una singola istanza o cluster PostgreSQL.
- Pool: tabelle condivise per i tenant in un'unica istanza e schema.

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Caso d'uso	L'isolamento dei dati con il pieno controllo dell'utilizzo delle risorse è un requisito fondamentale, altrimenti si hanno inquilini molto grandi e molto sensibili alle prestazioni.	L'isolamento dei dati è un requisito fondamentale e sono necessari riferimenti incrociati limitati o nulli ai dati degli inquilini.	Numero moderato di inquilini con una quantità moderata di dati. Questo è il modello preferito se devi fare riferimenti incrociati ai dati degli inquilini.	Grande numero di inquilini con meno dati per inquilino.
Agilità di onboarding dei nuovi inquilini	Molto lento. (È richiesta una nuova istanza o cluster per ogni tenant.)	Moderatamente lento. (Richiede la creazione di un nuovo database per ogni tenant per archiviare gli oggetti dello schema.)	Moderatamente lento. (Richiede la creazione di un nuovo schema per ogni tenant per archiviare gli oggetti.)	L'opzione più veloce. (È richiesta una configurazione minima).
Impegno ed efficienza di configurazione del pool di	È richiesto uno sforzo significativo. (Un pool di	È richiesto uno sforzo significativo. (Una configura	È richiesto un minore sforzo. (Una configura	Minimo sforzo richiesto.

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
connessioni al database	<p>connessioni per tenant.)</p> <p>Meno efficiente. (Nessuna condivisione della connessione al database tra i tenant.)</p>	<p>zione di pool di connessioni per tenant, a meno che non si utilizzi Amazon RDS Proxy).</p> <p>Meno efficiente. (Nessuna condivisione della connessione al database tra tenant e numero totale di connessioni. L'utilizzo tra tutti i tenant è limitato in base alla classe dell'istanza DB.)</p>	<p>connessioni per tutti i tenant.)</p> <p>Moderatamente efficiente. (Riutilizzo della connessione tramite il SET SCHEMA comando SET ROLE or solo in modalità pool di sessioni. SETi comandi causano anche il blocco della sessione quando si utilizza Amazon RDS Proxy, ma i pool di connessioni client possono essere eliminati ed è possibile effettuare connessioni dirette per ogni richiesta di efficienza.)</p>	<p>Il più efficiente. (Un pool di connessioni per tutti gli inquilini e riutilizzo efficiente delle connessioni tra tutti i tenant. I limiti di connessione al database si basano sulla classe dell'istanza DB.)</p>

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Manutenzione del database (gestione del vuoto) e utilizzo delle risorse	Gestione più semplice.	Complessità media. (Potrebbe comportare un elevato consumo di risorse, poiché successivamente è necessario avviare un aspirapolvere per ogni database <code>vacuum_timeout</code> , il che comporta un elevato utilizzo della CPU dell'Autovacuum Launcher. Potrebbe inoltre esercitare un sovraccarico aggiuntivo o associato all'eliminazione delle tabelle del catalogo del sistema PostgreSQL per ogni database.)	Tabelle di catalogo del sistema PostgreSQL di grandi dimensioni. (<code>pg_catalog</code> Dimensione totale in decine di GBs, a seconda del numero di inquilini e delle relazioni. Probabilmente richiederà modifiche ai parametri relativi all'aspirazione per controllare l'ingombro del tavolo.)	Le tabelle potrebbero essere di grandi dimensioni, a seconda del numero di inquilini e dei dati per inquilino. (È probabile che richieda modifiche ai parametri relativi all'aspirazione per controllare il gonfiore della tabella.)

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Impegno di gestione delle estensioni	Impegno significativo (per ogni database in istanze separate)	Impegno significativo (a ogni livello di database).	Sforzo minimo (una volta nel database comune).	Sforzo minimo (una volta nel database comune).
Modifica lo sforzo di implementazione	Impegno significativo. (Connect a ciascuna istanza separata e implementa le modifiche).	Sforzo significativo. (Connect a ogni database e schema e implementa le modifiche).	Sforzo moderato. (Connect al database comune e implementa le modifiche per ogni schema).	Sforzo minimo. (Connect a un database comune e implementa le modifiche.)
Implementazione delle modifiche: ambito di impatto	Minimo. (Singolo inquilino interessato.)	Minimo. (Singolo inquilino interessato.)	Minimo. (Singolo inquilino interessato.)	Molto grande. (Tutti gli inquilini interessati.)

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Gestione e gestione delle prestazioni delle query	Prestazioni gestibili delle query.	Prestazioni gestibili delle query.	Prestazioni gestibili delle query.	Probabilmente è necessari o uno sforzo significativo per mantenere le prestazioni delle query. (Nel tempo, le query potrebbero essere eseguite più lentamente e a causa delle maggiori dimensioni delle tabelle. È possibile utilizzare il partizionamento delle tabelle e la suddivisione del database per mantenere le prestazioni.)
Impatto sulle risorse tra i tenant	Nessun impatto. (Nessuna condivisione delle risorse tra gli inquilini.)	Impatto moderato. (I tenant condividono risorse comuni come la CPU e la memoria dell'istanza).	Impatto moderato. (I tenant condividono risorse comuni come la CPU e la memoria dell'istanza).	Impatto pesante. (Gli inquilini si influenzano a vicenda in termini di risorse, conflitti di blocco e così via.)

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Ottimizzazione a livello di tenant (ad esempio, creazione di indici aggiuntivi per tenant o modifica dei parametri del DB per un determinato tenant)	Possibile.	Piuttosto possibile. (È possibile apportare modifiche a livello di schema per ogni tenant, ma i parametri del database sono globali per tutti i tenant.)	In qualche modo possibile. (È possibile apportare modifiche a livello di schema per ogni tenant, ma i parametri del database sono globali per tutti i tenant.)	Non è possibile. (Le tabelle sono condivise da tutti gli inquilini.)

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Riequilibra gli sforzi per gli inquilini sensibili alle prestazioni	Minimo. (Non è necessario riequilibrare. Ridimensiona le risorse di server e I/O per gestire questo scenario.)	Moderato. (Utilizzate la replica logica o pg_dump per esportare il database, ma i tempi di inattività potrebbero essere lunghi a seconda delle dimensioni dei dati. Puoi utilizzare la funzionalità di database trasportabile di Amazon RDS for PostgreSQL per copiare i database tra istanze più velocemente.)	Moderato ma probabile che comporti tempi di inattività prolungati. (Utilizza la replica logica o pg_dump per esportare lo schema, ma i tempi di inattività potrebbero essere lunghi a seconda delle dimensioni dei dati).	Significativo, perché tutti i tenant condividono le stesse tabelle. (La suddivisione del database richiede la copia di tutto su un'altra istanza e un passaggio aggiuntivo per ripulire i dati dei tenant.) Molto probabilmente richiede una modifica della logica dell'applicazione.

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Tempo di inattività del database per gli aggiornamenti delle versioni principali	Tempo di inattività standard. (Dipende dalla dimensione del catalogo del sistema PostgreSQL.)	Probabilmente tempi di inattività più lunghi. (A seconda della dimensione del catalogo di sistema, il tempo può variare. Le tabelle del catalogo del sistema PostgreSQL vengono duplicate anche tra i database)	Probabilmente tempi di inattività più lunghi. (A seconda della dimensione del catalogo del sistema PostgreSQL, il tempo può variare.)	Tempo di inattività standard. (Dipende dalla dimensione del catalogo del sistema PostgreSQL.)
Sovraccarico di amministrazione (ad esempio, per l'analisi dei log del database o il monitoraggio dei processi di backup)	Impegno significativo	Sforzo minimo	Sforzo minimo.	Sforzo minimo.

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Disponibilità a livello di tenant	Massima. (Ogni inquilino fallisce e si riprende in modo indipendente.)	Ambito di impatto più elevato. (Tutti gli inquilini falliscono e si ripristinano insieme in caso di problemi hardware o di risorse.)	Ambito di impatto più elevato. (Tutti gli inquilini falliscono e si ripristinano insieme in caso di problemi hardware o di risorse.)	Ambito di impatto più elevato. (Tutti gli inquilini falliscono e si ripristinano insieme in caso di problemi hardware o di risorse.)
Attività di backup e ripristino a livello di tenant	Minimo sforzo. (È possibile eseguire il backup e il ripristino di ciascun inquilino in modo indipendente.)	Sforzo moderato. (Utilizza l'esportazione e l'importazione logiche per ogni tenant. Sono necessarie alcune operazioni di codifica e automazione.)	Sforzo moderato. (Utilizza l'esportazione e l'importazione logiche per ogni tenant. Sono necessarie alcune operazioni di codifica e automazione.)	Sforzo significativo. (Tutti gli inquilini condividono gli stessi tavoli.)
Sforzo di ripristino a livello di inquilino point-in-time	Sforzo minimo. (Utilizza il ripristino point-in-time utilizzando istantanee e o utilizza il backtracking in Amazon Aurora.)	Sforzo moderato. (Usa il ripristino delle istantanee, seguito da esportazione/importazione. Tuttavia, questa sarà un'operazione lenta.)	Sforzo moderato. (Usa il ripristino delle istantanee, seguito da esportazione/importazione. Tuttavia, questa sarà un'operazione lenta.)	Sforzo e complessità significativi.
Nome dello schema uniforme	Stesso nome dello schema per ogni inquilino.	Stesso nome dello schema per ogni tenant.	Schema diverso per ogni inquilino.	Schema comune.

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Personalizzazione per tenant (ad esempio, colonne di tabella aggiuntive per un tenant specifico)	Possibile.	Possibile.	Possibile.	Complicato (perché tutti gli inquilini condividono gli stessi tavoli).
Efficienza della gestione del catalogo a livello di mappatura relazionale degli oggetti (ORM) (ad esempio, Ruby)	Efficiente (perché la connessione client è specifica per un tenant).	Efficiente (perché la connessione client è specifica per un database).	Moderatamente efficiente e. (A seconda dell'ORM utilizzato, del modello di sicurezza utente/ruolo e della <code>search_path</code> configurazione, il client a volte memorizza nella cache i metadati per tutti i tenant, con conseguente utilizzo elevato della memoria di connessione DB.)	Efficiente (perché tutti i tenant condividono le stesse tabelle).

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Attività consolidata di rendicontazione degli inquilini	Sforzo significativo. (È necessario utilizzare data wrapper esterni [FDWs] per consolidare i dati in tutti i tenant o estrarre, trasformare e caricare [ETL] su un altro database di reporting.)	Sforzo significativo. (È necessario utilizzarli per FDWs consolidare i dati di tutti gli inquilini o ETL in un altro database di reporting.)	Sforzo moderato. (È possibile aggregare i dati in tutti gli schemi utilizzando i sindacati.)	Sforzo minimo. (Tutti i dati degli inquilini si trovano nelle stesse tabelle, quindi la creazione di report è semplice.)
Istanza di sola lettura specifica per il tenant per la generazione di report (ad esempio, basata sull'abbonamento)	Minimo sforzo. (Crea una replica di lettura).	Sforzo moderato. (È possibile utilizzare la replica logica o il AWS Database Migration Service [AWS DMS] per la configurazione.)	Sforzo moderato. (È possibile utilizzare la replica logica o AWS DMS la configurazione.)	Complicato (perché tutti i tenant condividono le stesse tabelle).

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Isolamento dei dati	Migliore.	Meglio. (È possibile gestire le autorizzazioni a livello di database utilizzando i ruoli PostgreSQL.)	Meglio. (È possibile gestire le autorizzazioni a livello di schema utilizzando i ruoli PostgreSQL.)	Peggioro. (Poiché tutti i tenant condividono le stesse tabelle, è necessario implementare funzionalità come la sicurezza a livello di riga [RLS] per l'isolamento dei tenant.)
Chiave di crittografia dello storage specifica per il tenant	Possibile (Ogni cluster PostgreSQL può avere la AWS Key Management Service propria chiave AWS KMS] per la crittografia dello storage.)	Non è possibile. (Tutti i tenant condividono la stessa chiave KMS per la crittografia dello storage).	Non è possibile. (Tutti i tenant condividono la stessa chiave KMS per la crittografia dello storage).	Non è possibile. (Tutti i tenant condividono la stessa chiave KMS per la crittografia dello storage).

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Utilizzo di AWS Identity and Access Management (IAM) per l'autenticazione del database per ogni tenant	Possibile.	Possibile.	Possibile (avendo utenti PostgreSQL separati per ogni schema).	Non possibile (perché le tabelle sono condivise da tutti i tenant).
Costo dell'infrastruttura	Il più alto (perché nulla è condiviso).	Moderato.	Moderato.	Il più basso.
Duplicazione dei dati e utilizzo dello storage	L'aggregato più elevato tra tutti gli inquilini. (Le tabelle del catalogo del sistema PostgreSQL e i dati statici e comuni dell'applicazione vengono duplicati su tutti i tenant.)	L'aggregato più alto tra tutti gli inquilini. (Le tabelle del catalogo del sistema PostgreSQL e i dati statici e comuni dell'applicazione vengono duplicati su tutti i tenant.)	Moderato. (I dati statici e comuni dell'applicazione possono trovarsi in uno schema comune e possono accedervi altri tenant.)	Minimo. (Nessuna duplicazione dei dati. I dati statici e comuni dell'applicazione possono trovarsi nello stesso schema.)

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Monitoraggio incentrato sul tenant (scopri rapidamente quale tenant sta causando problemi)	Minimo sforzo. (Poiché ogni inquilino viene monitorato separatamente, è facile controllare l'attività di un inquilino specifico.)	Sforzo moderato. (Poiché tutti gli inquilini condividono la stessa risorsa fisica, è necessario applicare filtri aggiuntivi per controllare l'attività di un inquilino specifico.)	Sforzo moderato. (Poiché tutti gli inquilini condividono la stessa risorsa fisica, è necessario applicare filtri aggiuntivi per controllare l'attività di un inquilino specifico.)	Sforzo significativo. (Poiché tutti i tenant condividono tutte le risorse, incluse le tabelle, è necessario utilizzare bind variable capture per verificare a quale tenant appartiene una specifica query SQL.)
Gestione centralizzata e monitoraggio dello stato di salute/attività	Impegno significativo (per configurare il monitoraggio centrale e un centro di comando centrale).	Sforzo moderato (perché tutti gli inquilini condividono la stessa istanza).	Sforzo moderato (perché tutti gli inquilini condividono la stessa istanza).	Impegno minimo (perché tutti i tenant condividono le stesse risorse, incluso lo schema).

	Silo	Bridge con database separati	Bridge con schemi separati	Pool
Possibilità di incrociare l'identificatore dell'oggetto (OID) e l'ID della transazione (XID)	Minimo.	Elevato. (Poiché OID, XID è un singolo contatore a livello di cluster PostgreSQL e possono verificarsi problemi di evacuazione efficace tra database fisici).	Moderato. (Poiché OID, XID è un singolo contatore PostgreSQL a livello di cluster).	Elevato. (Ad esempio, una singola tabella può raggiungere il limite TOAST OID di 4 miliardi, a seconda del numero di colonne.) out-of-line

Raccomandazioni di sicurezza a livello di riga

La sicurezza a livello di riga (RLS) è necessaria per mantenere l'isolamento dei dati dei tenant in un modello in pool con PostgreSQL. RLS centralizza l'applicazione delle politiche di isolamento a livello di database e rimuove l'onere di mantenere tale isolamento dagli sviluppatori di software. Il modo più comune per implementare RLS è abilitare questa funzionalità nel DBMS PostgreSQL. RLS prevede il filtraggio dell'accesso alle righe di dati in base a un valore in una colonna specificata. È possibile utilizzare due metodi per filtrare l'accesso ai dati:

- Una colonna di dati specificata in una tabella viene confrontata con il valore dell'utente PostgreSQL corrente. I valori nella colonna che sono equivalenti all'utente PostgreSQL che ha effettuato l'accesso sono accessibili a quell'utente.
- Una colonna di dati specificata in una tabella viene confrontata con il valore di una variabile di runtime impostata dall'applicazione. I valori nella colonna che sono equivalenti alla variabile di runtime sono accessibili durante quella sessione.

La seconda opzione è preferita, perché la prima richiede la creazione di un nuovo utente PostgreSQL per ogni tenant. Invece, un'applicazione SaaS che utilizza PostgreSQL dovrebbe essere responsabile dell'impostazione di un contesto specifico del tenant in fase di esecuzione quando interroga PostgreSQL. Ciò avrà l'effetto di applicare RLS. È inoltre possibile abilitare RLS su base individuale. table-by-table. Come procedura consigliata, è consigliabile abilitare RLS su tutte le tabelle che contengono dati dei tenant.

L'esempio seguente crea due tabelle e abilita RLS. Questo esempio confronta una colonna di dati con il valore della variabile runtime. `app.current_tenant`

```
-- Create a table for our tenants with indexes on the primary key and the tenant's name
CREATE TABLE tenant (
    tenant_id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,
    name VARCHAR(255) UNIQUE,
    status VARCHAR(64) CHECK (status IN ('active', 'suspended', 'disabled')),
    tier VARCHAR(64) CHECK (tier IN ('gold', 'silver', 'bronze'))
);

-- Create a table for users of a tenant
CREATE TABLE tenant_user (
    user_id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,
    tenant_id UUID NOT NULL REFERENCES tenant (tenant_id) ON DELETE RESTRICT,
```

```
email VARCHAR(255) NOT NULL UNIQUE,  
given_name VARCHAR(255) NOT NULL CHECK (given_name <> ''),  
family_name VARCHAR(255) NOT NULL CHECK (family_name <> '')  
);  
  
-- Turn on RLS  
ALTER TABLE tenant ENABLE ROW LEVEL SECURITY;  
  
-- Restrict read and write actions so tenants can only see their rows  
-- Cast the UUID value in tenant_id to match the type current_setting  
-- This policy implies a WITH CHECK that matches the USING clause  
CREATE POLICY tenant_isolation_policy ON tenant  
USING (tenant_id = current_setting('app.current_tenant')::UUID);  
  
-- And do the same for the tenant users  
ALTER TABLE tenant_user ENABLE ROW LEVEL SECURITY;  
  
CREATE POLICY tenant_user_isolation_policy ON tenant_user  
USING (tenant_id = current_setting('app.current_tenant')::UUID);
```

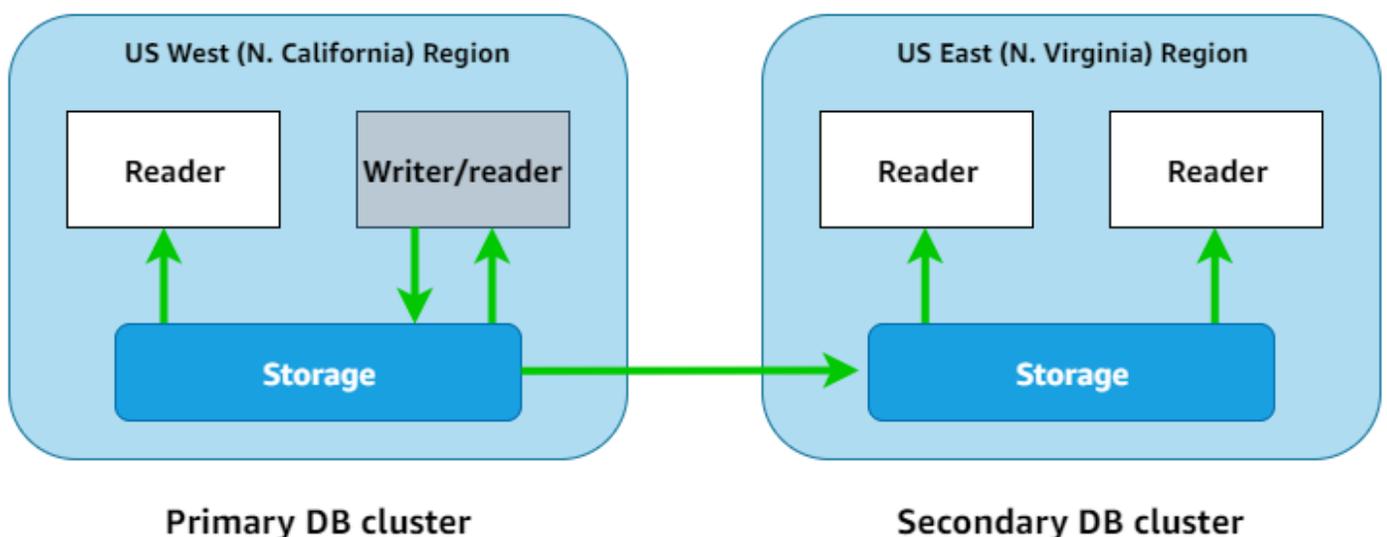
Per ulteriori informazioni, consulta il post sul blog [Isolamento dei dati multi-tenant con PostgreSQL Row Level Security](#). Il team AWS SaaS Factory ha anche [alcuni esempi GitHub per aiutare nell'implementazione di RLS](#).

Disponibilità di PostgreSQL per il modello pool

I modelli di pool per loro natura hanno una sola istanza PostgreSQL. Pertanto, la progettazione dell'applicazione per un'elevata disponibilità è fondamentale. Un guasto o un'interruzione di un database condiviso comporta il deterioramento dell'applicazione o l'inaccessibilità per tutti i tenant.

Le istanze DB di Amazon RDS for PostgreSQL possono essere rese ridondanti su due zone di disponibilità abilitando la funzionalità di alta disponibilità. Per ulteriori informazioni, consulta [High availability \(Multi-AZ\) per Amazon RDS nella documentazione](#) di Amazon RDS. Per il failover tra regioni, puoi creare una replica di lettura in una regione diversa. AWS (Questa replica di lettura deve essere promossa come parte di un processo di failover). Inoltre, è possibile replicare i backup replicati tra le regioni per il ripristino. AWS Per ulteriori informazioni, consulta [Replicazione dei backup automatici in un'altra AWS regione nella documentazione](#) di Amazon RDS.

Aurora, compatibile con PostgreSQL, esegue automaticamente il backup dei dati in modo da sostenere il fallimento di più zone di disponibilità. (Vedi [Alta disponibilità per Amazon Aurora nella documentazione](#) di Aurora.) Per rendere Aurora più resiliente e ripristinare più rapidamente, puoi creare repliche di lettura Aurora in altre zone di disponibilità. È possibile utilizzare i database globali Aurora per replicare i dati in cinque AWS regioni aggiuntive per il ripristino interregionale e il failover automatico. (Vedi [Utilizzo dei database globali di Amazon Aurora nella documentazione](#) di Aurora.) Inoltre, puoi abilitare l'[inoltro delle scritture](#) con i database globali Aurora per ottenere un'elevata disponibilità su più piattaforme. Regioni AWS



Indipendentemente dal fatto che tu stia utilizzando Amazon RDS for PostgreSQL o Aurora PostgreSQL, ti consigliamo di implementare funzionalità di alta disponibilità per mitigare l'impatto di eventuali interruzioni per tutte le applicazioni SaaS multi-tenant che utilizzano un modello di pool.

Best practice

Questa sezione elenca alcuni dei punti salienti di questa guida. Per discussioni dettagliate su ciascun punto, segui i collegamenti alle sezioni corrispondenti.

Confronta AWS le opzioni per PostgreSQL gestito

AWS offre due modi principali per eseguire PostgreSQL in un ambiente gestito. (In questo contesto, gestito significa che l'infrastruttura PostgreSQL e il DBMS sono parzialmente o completamente supportati da un servizio.) AWS Le opzioni AWS PostgreSQL gestite offrono il vantaggio di automatizzare i backup, il failover, l'ottimizzazione e una parte dell'amministrazione di PostgreSQL. Come opzioni gestite, AWS offre l'edizione compatibile con Amazon Aurora PostgreSQL e Amazon Relational Database Service (Amazon RDS) per PostgreSQL. Puoi selezionare la scelta migliore tra questi due modelli analizzando il tuo caso d'uso di PostgreSQL. Per ulteriori informazioni, consulta la sezione [Scelta tra Amazon RDS e Aurora](#) in questa guida.

Seleziona un modello di partizionamento SaaS multi-tenant

Puoi scegliere tra tre modelli di partizionamento SaaS applicabili a PostgreSQL: silo, bridge e pool. Ogni modello presenta vantaggi e svantaggi e dovresti scegliere il modello più ottimale in base al tuo caso d'uso. La compatibilità con Amazon RDS for PostgreSQL e Aurora PostgreSQL supporta tutti e tre i modelli. La scelta di un modello è fondamentale per mantenere l'isolamento dei dati dei tenant nelle applicazioni SaaS. Per una discussione dettagliata di questi modelli, consulta la sezione [Modelli di partizionamento SaaS multi-tenant per PostgreSQL in questa guida](#).

Usa la sicurezza a livello di riga per i modelli di partizionamento SaaS del pool

La sicurezza a livello di riga (RLS) è necessaria per mantenere l'isolamento dei dati dei tenant in un modello di pool con PostgreSQL. Questo perché non esiste una separazione logica tra infrastruttura, database PostgreSQL o schemi su base per-tenant in un modello di pool. RLS centralizza l'applicazione delle politiche di isolamento a livello di database e rimuove l'onere di mantenere tale isolamento dagli sviluppatori di software. È possibile utilizzare RLS per limitare le operazioni del database a un tenant specifico. Per ulteriori informazioni e un esempio, consultate la sezione [Raccomandazioni di sicurezza a livello di riga](#) di questa guida.

Domande frequenti

Questa sezione fornisce le risposte alle domande più frequenti sull'implementazione di PostgreSQL gestito in applicazioni SaaS multi-tenant.

Quali opzioni PostgreSQL gestite offre? AWS

AWS offre compatibilità con [Amazon Aurora PostgreSQL e Amazon Relational Database Service \(Amazon RDS\) per PostgreSQL](#). AWS dispone inoltre [di un ampio catalogo di offerte di database gestiti](#).

Qual è il servizio ottimale per le applicazioni SaaS?

Puoi utilizzare sia la compatibilità con Aurora PostgreSQL che Amazon RDS for PostgreSQL per applicazioni SaaS e tutti i modelli di partizionamento SaaS descritti in questa guida. Questi due servizi presentano differenze in termini di scalabilità, crash recovery, failover, opzioni di storage, alta disponibilità, disaster recovery, backup e classi di istanze disponibili per ciascuna opzione. La scelta ottimale dipenderà dal caso d'uso specifico. Utilizza la [matrice decisionale](#) contenuta in questa guida per scegliere l'opzione migliore per il tuo caso d'uso.

Quali requisiti unici devo prendere in considerazione se decido di utilizzare un database PostgreSQL con un'applicazione SaaS multi-tenant?

Come per qualsiasi archivio dati utilizzato con un'applicazione SaaS, la considerazione più importante è il metodo per mantenere l'isolamento dei dati dei tenant. Come illustrato in questa guida, esistono diversi modi per ottenere l'isolamento dei dati dei tenant con le offerte AWS PostgreSQL gestite. Inoltre, dovresti considerare l'isolamento delle prestazioni su base per-tenant per qualsiasi implementazione di PostgreSQL.

Quali modelli posso usare per mantenere l'isolamento dei dati dei tenant con PostgreSQL?

È possibile utilizzare i modelli silo, bridge e pool come strategie di partizionamento SaaS per mantenere l'isolamento dei dati dei tenant. Per una discussione su questi modelli e su come possono

essere applicati a PostgreSQL, consulta la sezione Modelli di partizionamento [SaaS multi-tenant per PostgreSQL in questa guida](#).

Come posso mantenere l'isolamento dei dati dei tenant con un singolo database PostgreSQL condiviso tra più tenant?

PostgreSQL supporta una funzionalità di sicurezza a livello di riga (RLS) che puoi utilizzare per imporre l'isolamento dei dati dei tenant in un singolo database o istanza PostgreSQL. Inoltre, puoi fornire database PostgreSQL separati per tenant in una singola istanza o creare schemi per tenant per raggiungere questo obiettivo. [Per i vantaggi e gli svantaggi di questi approcci, consulta la sezione Raccomandazioni di sicurezza a livello di riga di questa guida](#).

Passaggi successivi

AWS offre due opzioni per il funzionamento di PostgreSQL gestito: Aurora PostgreSQL compatibile e Amazon RDS for PostgreSQL. Ti consigliamo di valutare i due servizi e scegliere l'opzione che meglio supporta il tuo caso d'uso specifico per le tue applicazioni SaaS multi-tenant. La conformità a un modello di partizionamento SaaS può garantire che un'applicazione SaaS che utilizza PostgreSQL aderisca rigorosamente alle migliori pratiche per mantenere la tenancy. I modelli di partizionamento SaaS a silo, bridge e pool supportano molti casi d'uso SaaS. Questi modelli offrono diversi vantaggi tra fattori quali l'isolamento delle prestazioni, il sovraccarico operativo e la sicurezza degli inquilini.

Passaggi successivi:

- [Valuta la compatibilità con Aurora PostgreSQL e Amazon RDS for PostgreSQL e scegli l'opzione migliore per la tua applicazione SaaS.](#)
- [Seleziona un modello di partizionamento SaaS](#) che soddisfi i requisiti per la tua applicazione: silo, bridge o pool.
- Implementa PostgreSQL in base al modello di partizionamento SaaS selezionato.

Risorse

Riferimenti

- [Strategie di storage SaaS: creazione di un modello di storage multitenant](#) su (white paper) AWSAWS
- [Disaster recovery tra regioni con Amazon Aurora Global Database per Amazon Aurora PostgreSQL](#) (post di blog)AWS
- Isolamento [dei dati multitenant con PostgreSQL Row Level Security](#) (post sul blog)AWS
- [Lavorare con Amazon Aurora PostgreSQL](#) (documentazione Aurora)
- [PostgreSQL su Amazon RDS](#) (documentazione Amazon RDS)

Partner

- [Partner Amazon Aurora per PostgreSQL](#)
- [Partner Amazon RDS per PostgreSQL](#)

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
Aggiorna	Aggiornamenti che riflettono la disponibilità dell'inoltro di scrittura in Aurora.	29 aprile 2024
Aggiorna	È stata aggiornata la tabella di confronto tra Amazon RDS e Aurora .	21 ottobre 2022
=	Pubblicazione iniziale	30 settembre 2021

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 database Oracle locale su Oracle su un'istanza in EC2 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. Si esegue 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 di 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 aggregata

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

Intelligenza artificiale

Vedi [intelligenza artificiale](#).

AIOps

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

anonimizzazione

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

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

anti-modello

Una soluzione utilizzata di frequente 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 utilizzato 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.

distribuzioni blu/verdi

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 [AWS Cloud Adoption Framework](#).

implementazione canaria

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

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 o Bitbucket Cloud. Ogni versione del codice è denominata ramo. In una

struttura a microservizi, ogni repository è dedicato a una singola funzionalità. Una singola 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 is commonly described as a 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, puoi utilizzarlo AWS CloudFormation per [rilevare la deriva nelle risorse di sistema](#) oppure puoi usarlo AWS Control Tower per [rilevare cambiamenti nella tua landing zone](#) che potrebbero influire sulla conformità ai requisiti di governance.

DVSM

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

E

EDA

Vedi [analisi esplorativa dei dati](#).

MODIFICA

Vedi [scambio elettronico di dati](#).

edge computing

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

scambio elettronico di dati (EDI)

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

crittografia

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

chiave crittografica

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

endianità

L'ordine in cui i byte vengono archiviati nella memoria del computer. 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 pipeline CI/CD, l'ambiente di produzione è l'ultimo ambiente di implementazione.
- ambienti superiori: tutti gli ambienti a cui possono accedere utenti diversi dal team di sviluppo principale. Si può trattare di un ambiente di produzione, ambienti di preproduzione e ambienti per i test di accettazione da parte degli utenti.

epica

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

ERP

Vedi [pianificazione delle risorse aziendali](#).

analisi esplorativa dei dati (EDA)

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

F

tabella dei fatti

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

fallire velocemente

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

limite di isolamento dei guasti

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

ramo di funzionalità

Vedi [filiale](#).

caratteristiche

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

importanza delle caratteristiche

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

trasformazione delle funzionalità

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

prompt con pochi scatti

Fornire a un [LLM](#) un numero limitato di esempi che dimostrino l'attività e il risultato desiderato prima di chiedergli di eseguire un'attività simile. Questa tecnica è un'applicazione dell'apprendimento contestuale, in cui i modelli imparano da esempi (immagini) incorporati nei prompt. 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 il modello di base.](#)

modello di fondazione (FM)

Una grande rete neurale di deep learning che si è addestrata su enormi set di dati generalizzati e non etichettati. 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

AI generativa

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

blocco geografico

Vedi [restrizioni geografiche](#).

limitazioni geografiche (blocco geografico)

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

Flusso di lavoro di GitFlow

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

immagine dorata

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

strategia greenfield

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

guardrail

Una regola di 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 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 esclusione

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

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

Considera [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.

IloT

Vedi [Industrial Internet of Things](#).

infrastruttura immutabile

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

migrazione incrementale

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

I

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

Industria 4.0

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

infrastruttura

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

infrastruttura come codice (IaC)

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

IIoInternet delle cose industriale (T)

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

VPC di ispezione

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

Internet of Things (IoT)

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

interpretabilità

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

IoT

Vedi [Internet of Things](#).

libreria di informazioni IT (ITIL)

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

gestione dei servizi IT (ITSM)

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

ITIL

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

ITSM

Vedi [Gestione dei servizi IT](#).

L

controllo degli accessi basato su etichette (LBAC)

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

zona di destinazione

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

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

modello linguistico di grandi dimensioni (LLM)

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

migrazione su larga scala

Una migrazione di 300 o più server.

BIANCO

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

Privilegio minimo

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

eseguire il rehosting (lift and shift)

Vedi [7 R](#).

sistema little-endian

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

LLM

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

ambienti inferiori

Vedi [ambiente](#).

M

machine learning (ML)

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

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

ramo principale

Vedi [filiale](#).

malware

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

servizi gestiti

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

sistema di esecuzione della produzione (MES)

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

MAP

Vedi [Migration Acceleration Program](#).

meccanismo

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

account membro

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

MEH.

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

Message Queuing Telemetry Transport (MQTT)

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

microservizio

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

architettura di microservizi

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

Programma di accelerazione della migrazione (MAP)

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

migrazione su larga scala

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

fabbrica di migrazione

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

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

metadati di migrazione

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

modello di migrazione

Un'attività di migrazione ripetibile che descrive in dettaglio la strategia di migrazione, la destinazione della migrazione e l'applicazione o il servizio di migrazione utilizzati. Esempio: riorganizza la migrazione su Amazon EC2 con AWS 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. Per ulteriori informazioni, consulta la sezione [Abilitazione della persistenza dei dati nei microservizi](#).

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 \(RACI\)](#).

STRACCIO

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.](#)

Regione

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

regressione

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

riospitare

Vedi [7 R.](#)

rilascio

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

trasferisco

Vedi [7 Rs.](#)

ripiattaforma

Vedi [7 Rs.](#)

riacquisto

Vedi [7 Rs.](#)

resilienza

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

policy basata su risorse

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

matrice di assegnazione di responsabilità (RACI)

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

controllo reattivo

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

retain

Vedi [7 R](#).

andare in pensione

Vedi [7 Rs](#).

Retrieval Augmented Generation (RAG)

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

rotazione

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

controllo dell'accesso a righe e colonne (RCAC)

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

RPO

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

RTO

Vedi l'[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 AWS Management Console 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 AWS Secrets Manager, 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 EC2 Amazon 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

tags

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

variabile di destinazione

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

elenco di attività

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

Ambiente di test

[Vedi ambiente.](#)

training

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

Transit Gateway

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

flusso di lavoro basato su trunk

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

Accesso attendibile

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

regolazione

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

team da due pizze

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

U

incertezza

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

compiti indifferenziati

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

ambienti superiori

[Vedi ambiente.](#)

V

vacuum

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

controllo delle versioni

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

Peering VPC

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

vulnerabilità

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

W

cache calda

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

dati caldi

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

funzione finestra

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

Carico di lavoro

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

flusso di lavoro

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

VERME

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

WQF

Vedi [AWS Workload Qualification Framework](#).

scrivi una volta, leggi molte (WORM)

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

Z

exploit zero-day

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

vulnerabilità zero-day

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

prompt zero-shot

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

applicazione zombie

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

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