Creazione di architetture multi-tenant per l'intelligenza artificiale agentica su AWS

AWS Guida prescrittiva

AWS Guida prescrittiva: Creazione di architetture multi-tenant per l'intelligenza artificiale agentica su AWS

Table of Contents

Creazione di architetture multi-tenant per l'intelligenza artificiale agentica su AWS

Aaron Sempf e Tod Golding, Amazon Web Services

Luglio 2025 (storia del documento)

L'intelligenza artificiale agentica rappresenta un cambiamento di paradigma dirompente che richiede alle organizzazioni di ripensare a come costruire, fornire e gestire i propri sistemi. Il modello agentico prevede che i team esplorino nuovi modi per scomporre i sistemi in uno o più agenti che creano nuovi percorsi, possibilità e valori.

Gran parte della discussione agentica è incentrata sugli strumenti, i framework e i modelli utilizzati per creare e implementare gli agenti. Non dobbiamo solo adottare buoni strumenti per creare agenti, ma anche nuovi protocolli di integrazione, strategie di autenticazione e meccanismi di scoperta che possano fungere da base per le architetture agentiche.

Mentre il numero di strumenti agentici cresce, i team devono anche considerare in che modo i loro agenti affrontano le sfide architettoniche più tradizionali. Scalabilità, rumorosità, resilienza, costi ed efficienza operativa sono argomenti fondamentali che devono essere valutati durante la progettazione, la creazione e l'implementazione degli agenti. A prescindere da quanto possano essere autonomi e intelligenti gli agenti, dobbiamo anche garantire che raggiungano economie di scala, efficienza e agilità in linea con le esigenze aziendali.

L'obiettivo di questa guida è esplorare varie dimensioni dell'impronta degli agenti. Ciò include la revisione dei vari modelli di implementazione e consumo degli agenti e l'evidenziazione delle diverse strategie per la creazione di agenti che soddisfino gli obiettivi architetturali. Significa anche esaminare come gli agenti potrebbero essere utilizzati in un ambiente multi-tenant introducendo costrutti interni che sono generalmente richiesti in un ambiente multi-tenant.

Destinatari principali

Questa guida è rivolta ad architetti, sviluppatori e leader tecnologici che desiderano creare sistemi multi-tenant basati sull'intelligenza artificiale.

Obiettivi

Questa guida ti consente di:

- Comprendi le implementazioni di agenti multi-tenant, esplora i modelli suddivisi in silos e in pool e come il contesto dei tenant influisce sull'implementazione degli agenti
- Esplora la gestione degli agenti, tra cui onboarding, isolamento dei tenant e gestione delle risorse in ambienti con uno o più provider
- Valuta gli aspetti degli agenti multi-tenant, tra cui la proprietà dei dati, il monitoraggio e i test

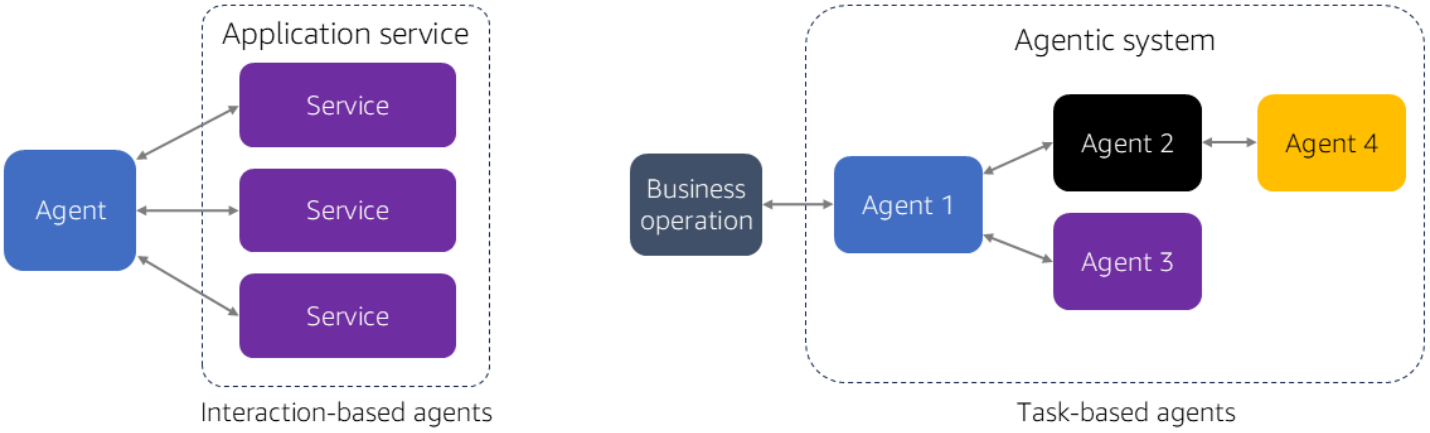
Informazioni su questa serie di contenuti

Questa guida fa parte di una serie sull'intelligenza artificiale agentica su. AWS Per ulteriori informazioni e per visualizzare le altre guide di questa serie, consulta Agentic AI sul sito Web Prescriptive Guidance. AWS

Nozioni di base sugli agenti

Prima di discutere dei dettagli architettonici, dovremmo delineare i diversi ruoli svolti dagli agenti, perché «agente» è un termine sovraccarico che può essere applicato a molti casi d'uso. Cominciamo con alcuni termini generici che possono aiutare a classificarli.

Al livello più esterno, dobbiamo iniziare classificando il ruolo e la natura degli agenti. Questo è difficile perché esiste un'ampia gamma di scenari in cui gli agenti possono essere applicati a qualsiasi numero di problemi. In questa discussione, tuttavia, ci concentriamo su cosa significa introdurre un agente in un'applicazione o in un sistema. In questo modello, sottolineiamo come e dove gli agenti possono arricchire al meglio l'esperienza del sistema. Le opzioni scelte influiscono sul modo in cui gli agenti vengono creati, integrati e applicati a diversi domini e casi d'uso. Il diagramma seguente mostra due modelli agentici utilizzati dai costruttori.

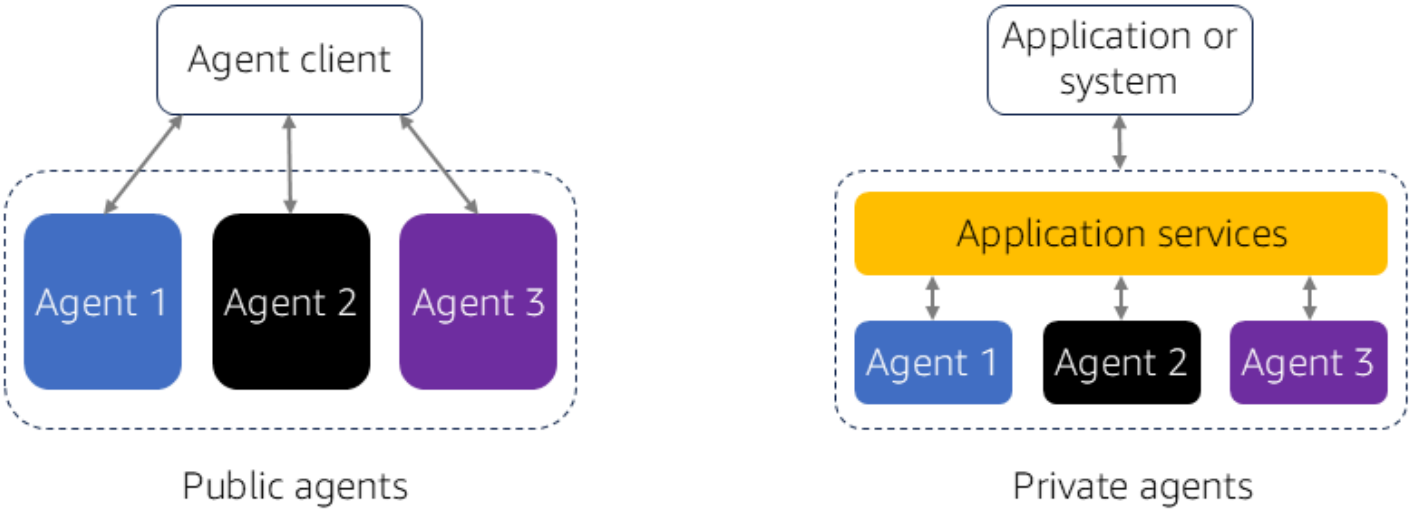


Sul lato sinistro del diagramma c'è un agente basato sull'interazione. In questa modalità, un agente crea una vista di un sistema esistente per orchestrare le interazioni con i servizi sottostanti per raggiungere un obiettivo o un risultato. La chiave è che l'agente venga aggiunto a un sistema come approccio alternativo per migliorare le caratteristiche e le capacità del sistema. Immagina, ad esempio, che un fornitore di software indipendente (ISV) disponga di un sistema di contabilità con una UX utilizzata per eseguire operazioni. L'agente basato sull'interazione semplifica l'interazione con queste funzionalità esistenti. Si tratta meno di imparare a raggiungere un obiettivo vagamente definito e più di fornire un modo per orchestrare percorsi noti.

Al contrario, il sistema basato sulle attività sul lato destro del diagramma rappresenta un approccio diverso. Gli agenti di quel sistema utilizzano le proprie conoscenze e abilità per imparare a completare le attività e ottenere risultati aziendali. Si potrebbe sostenere che entrambi i modelli ottengono risultati aziendali, ma un modello basato sulle attività si affida agli agenti stessi per

determinare come raggiungere un risultato. Tali agenti sono meno deterministi e si affidano invece alla loro capacità di apprendere ed evolversi. Al contrario, gli agenti basati sull'interazione sono progettati principalmente per orchestrare una serie di funzionalità note. Queste differenze influiscono sul modo in cui crei, definisci e integri gli agenti a supporto del tuo business.

Abbiamo anche bisogno di termini che descrivano come e dove impieghiamo gli agenti. La posizione in cui un agente vive all'interno del sistema può influire sul modo in cui questo viene costruito, definito e protetto. Il diagramma seguente delinea due modelli distinti che possono essere applicati agli agenti.

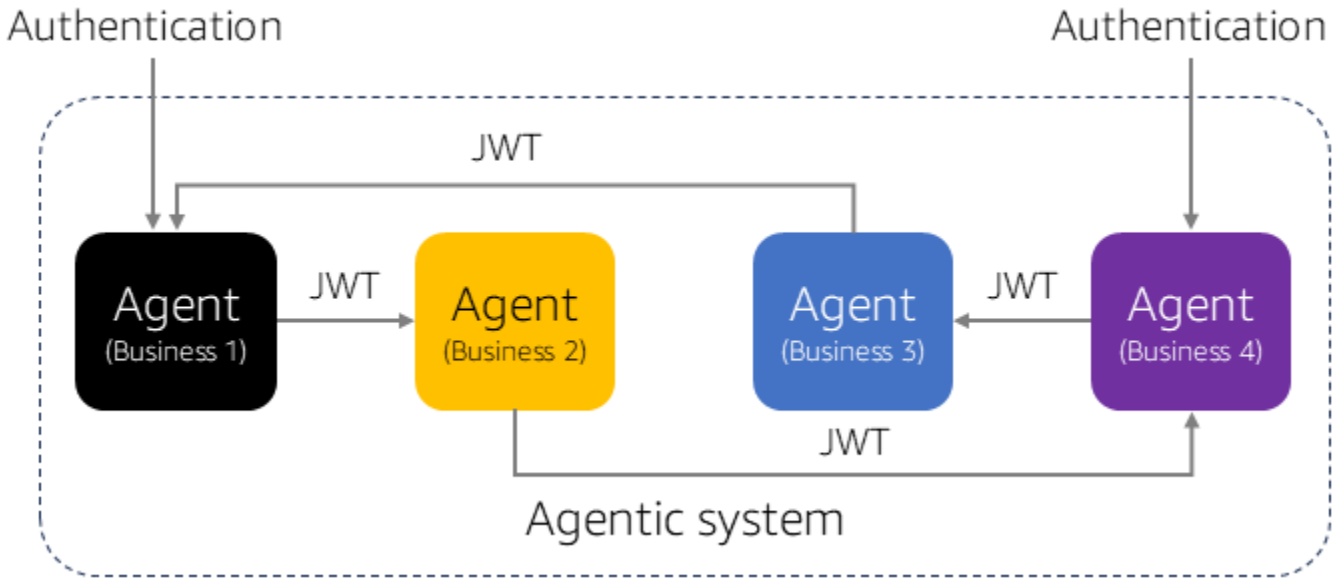


Sul lato sinistro del diagramma c'è un sistema di distribuzione con tre agenti diversi. Gli agenti sono esposti a client esterni che possono essere altri agenti o applicazioni. Per questo modello, gli agenti vengono definiti agenti pubblici.

Al contrario, il diagramma a destra mostra gli agenti all'interno dell'implementazione della soluzione. In questo caso, esistono una serie di servizi applicativi che vengono utilizzati dagli utenti o dai sistemi. Questi utenti interagiscono con l'applicazione senza rendersi conto del fatto che gli agenti fanno parte dell'esperienza. Gli agenti vengono quindi richiamati e orchestrati dai servizi del sistema sottostante. Gli agenti distribuiti in questo modo vengono definiti agenti privati.

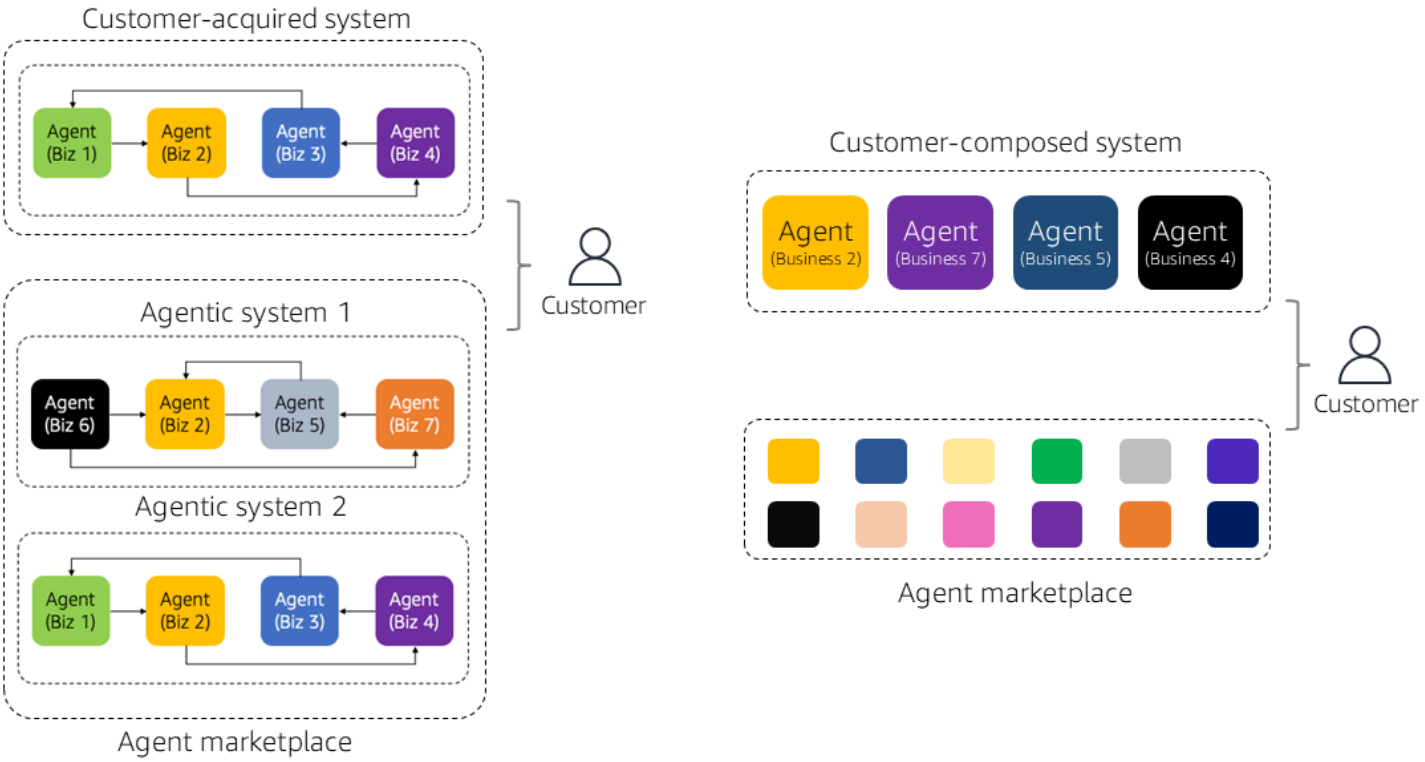
Gran parte del valore di un agente si concentra sul modello pubblico in cui i fornitori possono pubblicare i propri agenti con l'intenzione di integrarli con altri agenti di terze parti. Gli agenti farebbero quindi parte di una rete o rete di servizi interconnessi che, collettivamente, sono in grado di soddisfare molti casi d'uso. Sebbene questi agenti possano essere utilizzati in molti domini, il caso business-to-business d'uso è una soluzione naturale. Il diagramma seguente fornisce una visione

concettualizzata di come potrebbe essere assemblare un agente di raccolta che risolva un problema specifico.



Il diagramma mostra quattro agenti commerciali che collaborano per raggiungere una serie di obiettivi. Quando gli agenti sono composti in questo modo, rappresentano un sistema agentico e esistono molti tipi di tali sistemi. Potrebbero essere un set preconfezionato di agenti collaboratori che vengono comunemente consumati come singola unità. Oppure il sistema potrebbe essere assemblato dinamicamente dai clienti che desiderano scegliere una combinazione di agenti che meglio soddisfi le loro esigenze.

Entrambi gli approcci offrono percorsi praticabili per l'integrazione degli agenti. Alcuni agenti sono progettati con l'aspettativa di essere integrati in sistemi specifici in cui possono massimizzare il loro valore, la loro portata e il loro impatto. Questa nozione di sistemi agentici solleva anche interrogativi su come vengono acquisiti gli agenti e potrebbero esserci molti modi per risolvere questo problema. Il diagramma seguente fornisce esempi di come questi agenti e sistemi possono essere creati attraverso esperienze transazionali.

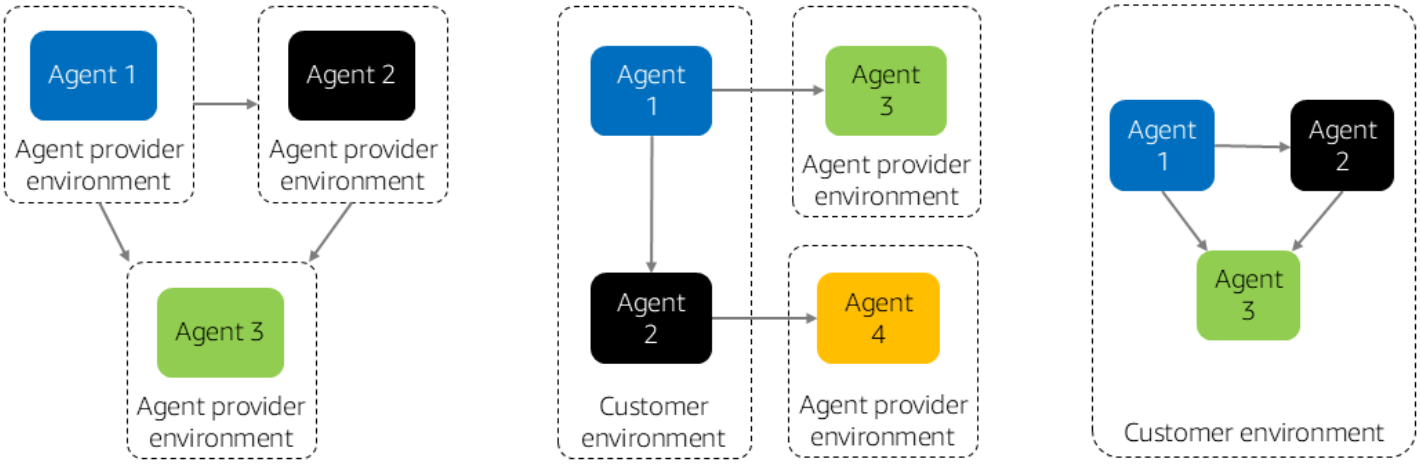


Vengono mostrati due esempi di esperienze di mercato. Sul lato sinistro, viene utilizzato un marketplace per acquisire sistemi preconfezionati. In questo scenario, il mercato scopre e integra sistemi che rispondono a obiettivi più ampi che richiedono l'integrazione e l'orchestrazione di più agenti.

L'esempio a destra mostra un marketplace in cui gli agenti vengono scoperti e composti in sistemi agentici. In questo scenario, i clienti possono creare qualsiasi sistema di agenti compatibili e integrati per soddisfare le loro esigenze. La capacità di assemblare gli agenti in questo modo dipende dal modello di compatibilità e dai requisiti di integrazione dei singoli agenti.

Considerazioni sull'hosting degli agenti

Ora che hai un'idea dei concetti più ampi relativi agli agenti, esaminiamo cosa significa ospitare ed eseguire questi agenti. Dobbiamo pensare a come e dove vengono eseguiti i calcoli, come si scalano, come funzionano e come vengono gestiti. Allo stesso tempo, alcuni modelli che ci aspettiamo di vedere come agenti sono applicati e adottati più ampiamente. Il diagramma seguente mostra un esempio di possibili permutazioni.



Qui sono rappresentate tre strategie distinte. Sul lato sinistro del diagramma, viene visualizzato un modello in cui i nostri agenti sono ospitati, scalati e gestiti all'interno degli ambienti di ciascun fornitore di agenti. Questi agenti vengono pubblicati e utilizzati come servizi e operano secondo il modello denominato Agent as a Service (AaaS). Sul lato destro c'è un modello in cui gli agenti di un provider sono tutti ospitati in un ambiente dedicato al cliente.

Al centro del diagramma c'è un modello di implementazione misto che combina queste due strategie, ospitando alcuni agenti localmente nell'ambiente del cliente e interagendo con alcuni agenti ospitati in remoto nell'ambiente di un provider.

Una quarta opzione (non illustrata) potrebbe consistere nella creazione di agenti sotto forma di servizi a basso contenuto di codice o privi di codice, scalabili e gestiti dai servizi di infrastruttura degli agenti. Non li tratteremo in dettaglio perché l'architettura e l'hosting degli agenti gestiti sono dettati principalmente dall'organizzazione proprietaria dei servizi.

Potete immaginare la gamma di fattori che potrebbero influenzare l'adozione di uno di questi modelli. I vincoli di conformità, normativi e di sicurezza, ad esempio, potrebbero spingere qualcuno a preferire

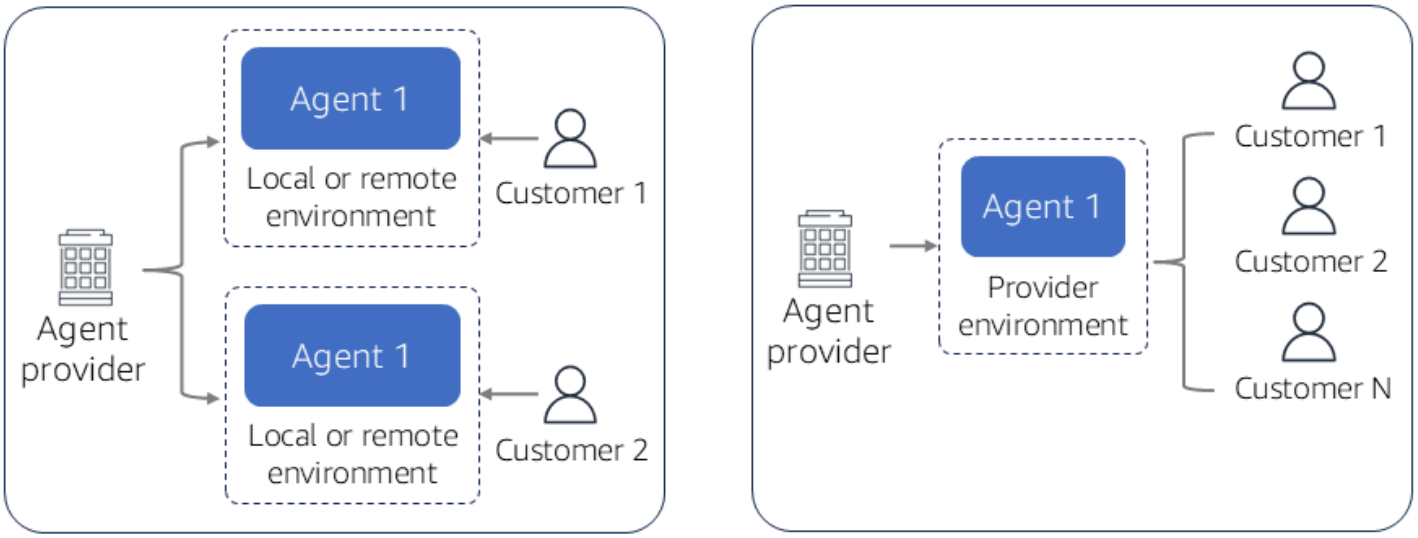
agenti ospitati presso i clienti. Scalabilità, agilità ed efficienza potrebbero spingere le organizzazioni verso il modello AaaS.

Il concetto chiave in questo caso è che gli agenti possono e vengono distribuiti e ospitati in molti modi. Il tuo compito è determinare in che modo gli agenti possono essere utilizzati al meglio. L'impronta, la sicurezza e l'implementazione, tra gli altri fattori, influiscono in modo significativo sul modo in cui ci si avvicina agli agenti edili e operativi. Gli agenti privati e pubblici, ad esempio, possono avere design e cicli di vita di rilascio diversi.

Gli agenti incontrano la multi-tenancy

È facile pensare agli agenti come elementi costitutivi in cui gli agenti sono visti come una serie di componenti autonomi assemblati per supportare le esigenze di un dominio o di un problema aziendale specifico. La cosa diventa più interessante quando iniziamo a pensare a come questi agenti vengono impacchettati e utilizzati dai provider. Per molti aspetti, un agente diventa una fonte di costi e ricavi per un'azienda. I fornitori di agenti devono considerare le diverse persone che utilizzano i loro servizi, il profilo di consumo delle persone e le strategie di monetizzazione che consentono ai fornitori di agenti di creare modelli di prezzi e livelli in linea con i consumatori.

I fornitori di agenti potrebbero supportare diversi modelli per l'implementazione dei propri agenti per soddisfare le esigenze dei clienti. Il diagramma seguente mostra una visione concettuale dei due principali modelli di distribuzione degli agenti.



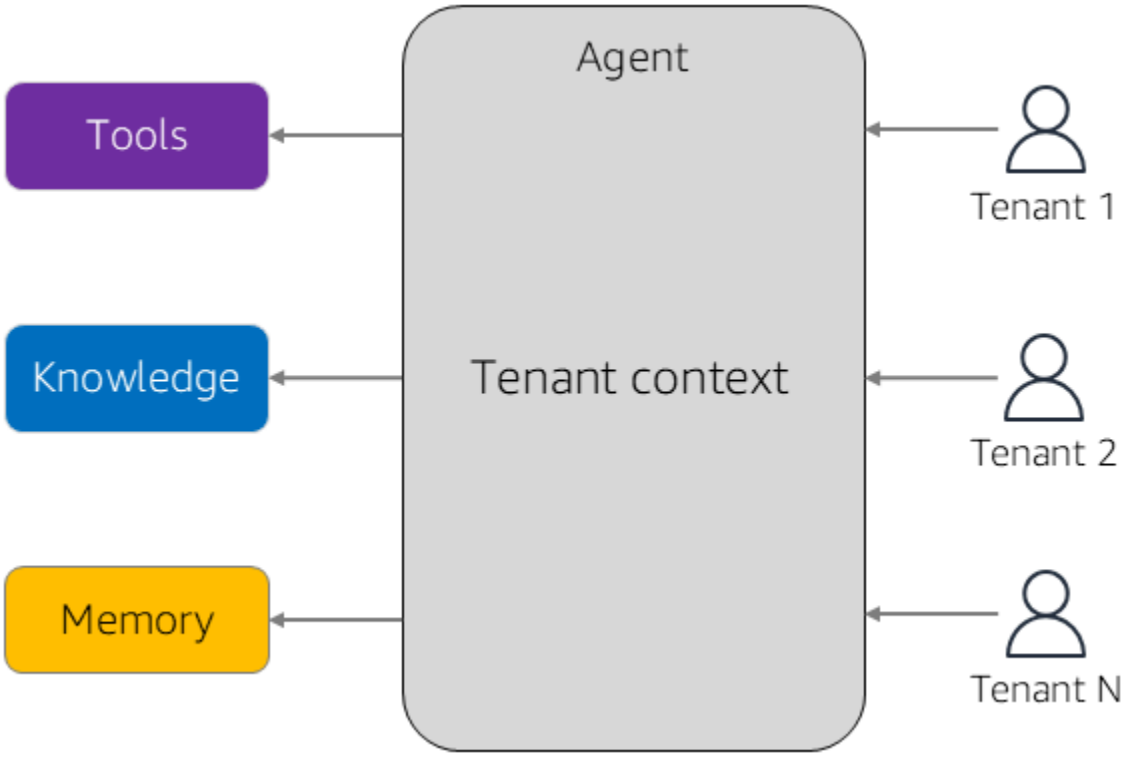
Il lato sinistro del diagramma mostra il modello di agente dedicato al cliente. Un fornitore di agenti crea un agente implementando un'istanza di agente separata per ogni cliente registrato. Con questo approccio, le capacità dell'agente e la sua capacità di acquisire conoscenze sarebbero limitate all'ambito dell'ambiente di un determinato cliente. Ciò finisce per rappresentare un'esperienza per cliente che eredita alcune delle complessità e dei vantaggi del supporto di ambienti dedicati ai clienti.

Al contrario, il diagramma sul lato destro del diagramma presenta un singolo agente che viene distribuito nell'ambiente del provider. L'agente elabora le richieste di più clienti, evolvendole e apprende sulla base dell'esperienza collettiva di tutti i clienti. Ogni nuovo cliente aggiunto

rappresenterebbe semplicemente un altro cliente valido dell'agente. L'agente funziona come un modello Agent as a Service (AaaS), utilizzando costrutti condivisi per supportare le esigenze del cliente. In entrambi i casi, gli agenti consumatori possono essere applicazioni, sistemi o anche altri agenti.

Esistono due modi per considerare il modello AaaS. Il modello sopra riportato offre la stessa esperienza a tutti i clienti. Ciò significa che gli interni dell'agente non includeranno alcun livello di specializzazione che consideri il contesto del cliente richiedente. In genere, per questa modalità, si presuppone che la natura dell'ambito, degli obiettivi e del valore di un agente sia incentrata su un insieme condiviso di risorse, conoscenze e risultati applicati universalmente a tutti i clienti.

L'approccio alternativo all'AaaS prevede che il contesto dei clienti influenzi l'esperienza e l'implementazione dell'agente. Il diagramma seguente fornisce una visione concettuale dell'impronta di un agente AaaS in questo contesto.



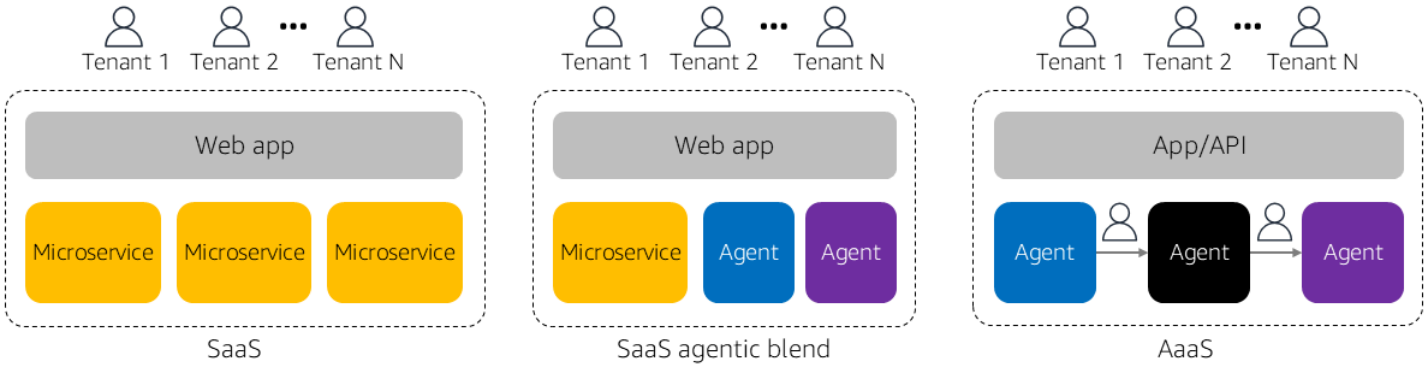
In questa visualizzazione AaaS, l'origine e il contesto delle richieste in arrivo influiscono in modo significativo sull'impronta dell'agente. Le risorse, le azioni e gli strumenti che fanno parte dell'implementazione sottostante dell'agente possono variare per ogni richiesta del tenant in entrata. Il valore di un agente è legato alla sua capacità di utilizzare il contesto del tenant per ottenere azioni e risultati influenzati dallo stato del tenant, dalle conoscenze e da altri fattori. Alcune richieste possono produrre un risultato unico per il tenant, mentre altre possono portare a risultati più personalizzati per

ogni tenant. Ciò aggiunge una nuova dimensione alla capacità di apprendimento dell'agente, che potrebbe includere una maggiore contestualità e l'acquisizione e l'applicazione delle conoscenze che migliorano i risultati mirati.

Per i provider, il modello AaaS offre molti vantaggi. Poiché più clienti utilizzano un solo agente, il provider ha maggiori opportunità di realizzare economie di scala, promuovere l'efficienza operativa, controllare i costi e creare un'esperienza di gestione unificata. Ciò ha il potenziale per una maggiore agilità, innovazione e crescita per il business degli agenti.

Queste qualità si sovrappongono agli stessi principi che guidano l'adozione del modello SaaS (Software as a Service). In sostanza, il modello AaaS è costruito come un servizio multi-tenant che eredita molti degli stessi attributi di scala, resilienza, isolamento, onboarding e operativi presenti in un ambiente SaaS. Per molti aspetti, l'esperienza AaaS si ispira molto alle strategie e alle pratiche utilizzate dai provider SaaS, ma è ragionevole separare questi termini. Per i nostri scopi, l'accento è posto principalmente sulle implicazioni derivanti dagli agenti edili e operativi che richiedono un supporto multi-tenant.

Per un sistema in grado di trattare tutti gli utenti allo stesso modo e che non richieda la gestione di dati persistenti, sensibili o specifici del cliente, il concetto di locazione avrebbe un impatto minimo sugli agenti. Per i sistemi che dovrebbero servire più clienti preservando l'isolamento dei dati, la personalizzazione e la consapevolezza del contesto, il supporto di più tenant potrebbe essere un elemento essenziale della progettazione, della strategia e dell'obiettivo di un agente. Il diagramma seguente mostra come la multi-tenancy può essere utilizzata in ambienti agentici.



Sul lato sinistro di questo diagramma c'è una classica architettura multi-tenant. Include un'applicazione Web e una serie di microservizi che implementano la logica aziendale. Più tenant utilizzano l'infrastruttura condivisa di questo ambiente, scalabile per soddisfare i mutevoli carichi di lavoro di una popolazione di inquilini in continua evoluzione. L'ambiente è gestito e gestito da un unico pannello di controllo per tutti gli inquilini.

Immagina come questo modello mentale corrisponda all'agente sul lato destro di questo diagramma. Un agente esegue un modello AaaS utilizzato da uno o più tenant. Gli agenti potrebbero provenire da più provider e il contesto dei tenant fluisce tra di loro, poiché una singola istanza di un agente deve elaborare le richieste di più tenant.

L'esempio al centro di questo diagramma è un modello ibrido in cui gli agenti fanno parte dell'esperienza SaaS complessiva. Alcune parti del sistema sono implementate in un modello più tradizionale e altre parti del sistema si basano su agenti. È probabile che questo modello sia comune a molte offerte SaaS, in particolare per le organizzazioni che stanno passando a un'esperienza agentica. È normale che questo modello persista perché non tutti i sistemi vengono forniti come puro modello AaaS. Si noti inoltre che la multi-tenancy si applica ancora agli agenti del modello. Sebbene gli agenti possano essere incorporati in un sistema, possono comunque elaborare le richieste di più tenant.

È naturale chiedersi se la multi-tenancy sia davvero importante. Si potrebbe sostenere che un agente elabora le richieste, quindi supportare la locazione potrebbe avere scarso effetto. Tuttavia, man mano che approfondiamo le implicazioni degli agenti multi-tenant, la locazione può influire direttamente sul modo in cui gli agenti influenzano il modo in cui gli agenti influenzano il modo in cui gli strumenti, la memoria, i dati e altre parti degli agenti vengono accessibili, distribuiti e configurati per supportare i singoli tenant. La locazione influenza anche il modo in cui la scalabilità, la limitazione, la determinazione dei prezzi, la suddivisione in più livelli e altri aspetti aziendali si applicano all'architettura dell'agente.

Uno dei punti salienti di questa situazione è che esistono casi d'uso agentici che richiedono un supporto multi-tenancy. La sfida consiste nel determinare in che modo la multi-tenancy influenzi il design e l'architettura complessivi della vostra esperienza agentica. Per alcuni agenti, il supporto multi-tenant rappresenta una capacità di differenziazione, che consente agli agenti di applicare un contesto specifico del tenant agli agenti che forniscono risultati mirati.

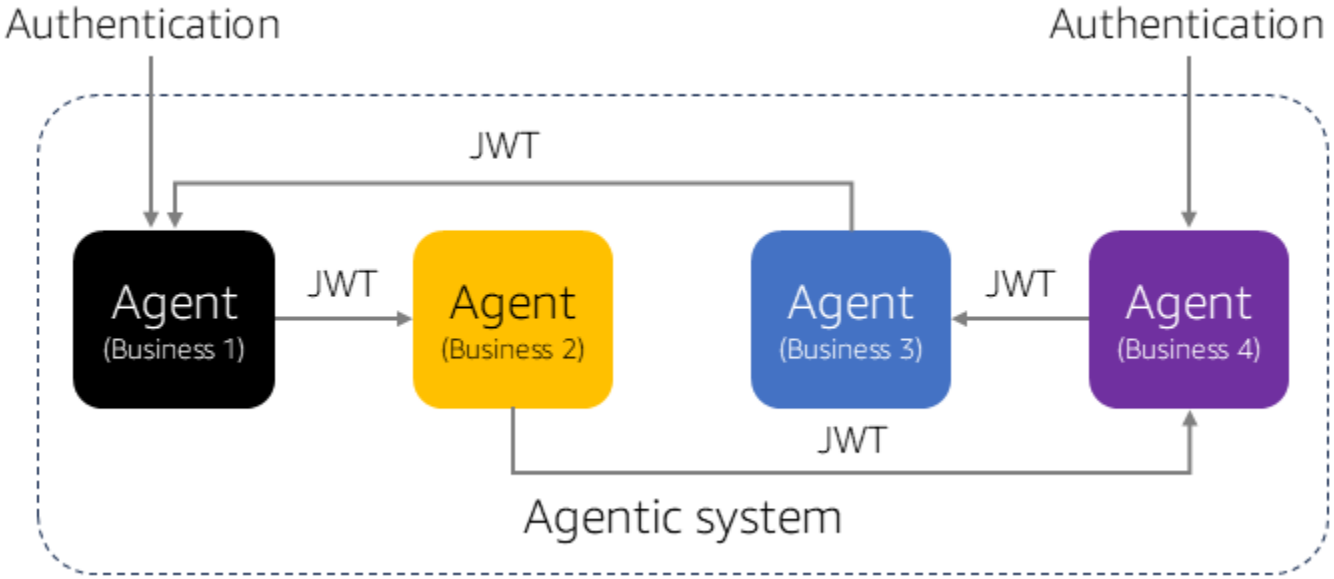
Nelle sezioni successive, vedrete come saranno utili la terminologia e i modelli di progettazione che creiamo per descrivere le architetture SaaS multi-tenant. Questi concetti possono essere adottati dal modello AaaS prendendo in prestito aspetti utili, che introduce nuovi concetti specifici per agente laddove necessari.

Identità, contesto degli inquilini e sistemi agentici

L'aggiunta del contesto del tenant ai singoli agenti non è particolarmente impegnativa. In molti casi, i team possono fare affidamento su meccanismi tipici che vincolano utenti e sistemi agli inquilini e

trasferiscono token che riconoscono i tenant agli agenti. Questo è importante se consideriamo in che modo il contesto e l'identità dei tenant supportano più agenti. In questo modello, gli inquilini devono essere legati a un'identità che comprenda tutti gli agenti che collaborano.

In generale, il dominio agentico richiede un modello di identità più trasversale, in linea con le esigenze attuali ed emergenti dei sistemi agentici. I provider di agenti richiedono meccanismi di identità che supportino modelli di sicurezza, conformità e autorizzazione esclusivi forniti con i sistemi operativi agentic. Ciò è particolarmente difficile in ambienti in cui i sistemi sono composti da clienti o altri agenti. Ogni agente integrato deve collegare la propria identità e il contesto del tenant alle interazioni con gli agenti. Il diagramma seguente evidenzia le potenziali sfide relative all'identità e al contesto del tenant che fanno parte delle agent-to-agent interazioni (a2a).



Questo diagramma mostra una serie di agenti creati dal provider che interagiscono come parte del sistema agentico trattato. Ora è stato aggiornato con l'identità e il contesto degli inquilini. Questo scenario è un esempio di sistema agentico che supporta più punti di ingresso. Partiamo dal presupposto che ogni agente di questo sistema richieda il proprio meccanismo di autenticazione per assegnare il sistema o l'utente a un determinato tenant. Man mano che questi agenti interagiscono, il contesto del tenant viene passato a un token web JSON (JWT) che verrà utilizzato per autorizzare l'accesso e inserire il contesto del tenant nell'agente.

Concettualmente, la differenza principale rispetto a questo scenario è che gli agenti si implementano e operano in modo indipendente, il che significa che ogni agente deve essere in grado di determinare la propria identità e autorizzare l'accesso. La chiave è che la sua identità deve avere una certa capacità distribuita di gestire le esigenze del più ampio sistema agentico. È inoltre necessario un allineamento sul modo in cui gli agenti condividono il contesto degli inquilini.

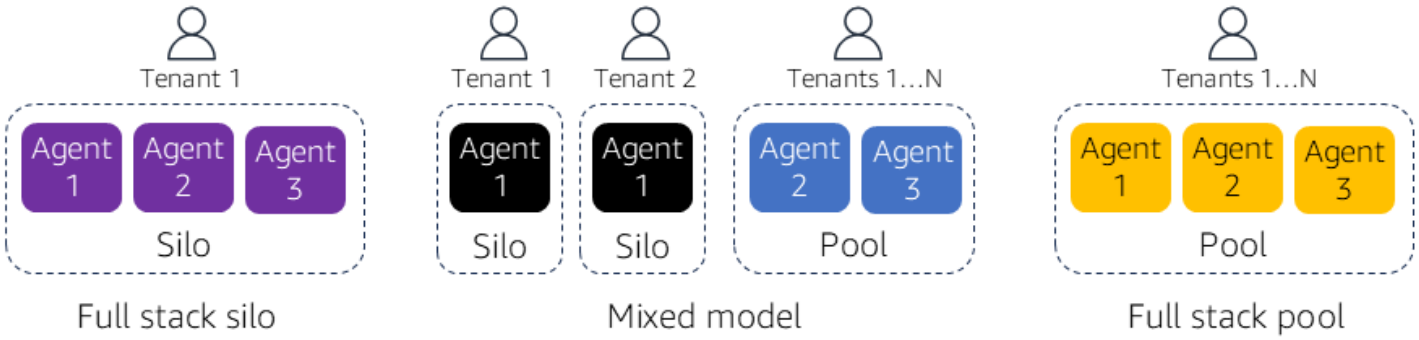
Applicare il valore aziendale SaaS al modello AaaS

In generale, quando consideriamo l'utilizzo di qualsiasi sistema all'interno di un as-a-service modello, consideriamo la natura dell'esperienza e il modo in cui la sua impronta tecnica e operativa determina i risultati aziendali. Quando adottano il SaaS, ad esempio, le organizzazioni utilizzano economie di scala, efficienze operative, profili di costo e agilità per promuovere crescita, margini e innovazione.

È probabile che gli agenti forniti come AAAs perseguano risultati aziendali simili. Supportando più inquilini, un agente può allineare il consumo di risorse alle attività degli inquilini. In questo modo si ottengono economie di scala tipiche degli ambienti SaaS tradizionali. Il modello AaaS consente inoltre alle organizzazioni di gestire, utilizzare e distribuire gli agenti in modo da consentire rilasci frequenti e favorire l'agilità dei fornitori di agenti. La chiave è che il modello AaaS non dipende dalla tecnologia. Crea e guida strategie aziendali che promuovono la crescita, semplificano l'adozione e semplificano le operazioni.

Modelli di distribuzione degli agenti

In un'esperienza AAaS di base, un provider può implementare agenti utilizzando vari modelli. Esistono una miriade di fattori che influenzano il modo in cui gli agenti vengono implementati per soddisfare le esigenze dei clienti, in termini di prestazioni, conformità, geografia e sicurezza. Diverse strategie di implementazione influiscono sul modo in cui un agente viene progettato, implementato e utilizzato. È qui che possiamo introdurre termini multi-tenant classici per etichettare diverse strategie di implementazione. Il diagramma seguente mostra diverse permutazioni per la distribuzione degli agenti in un ambiente AaaS.



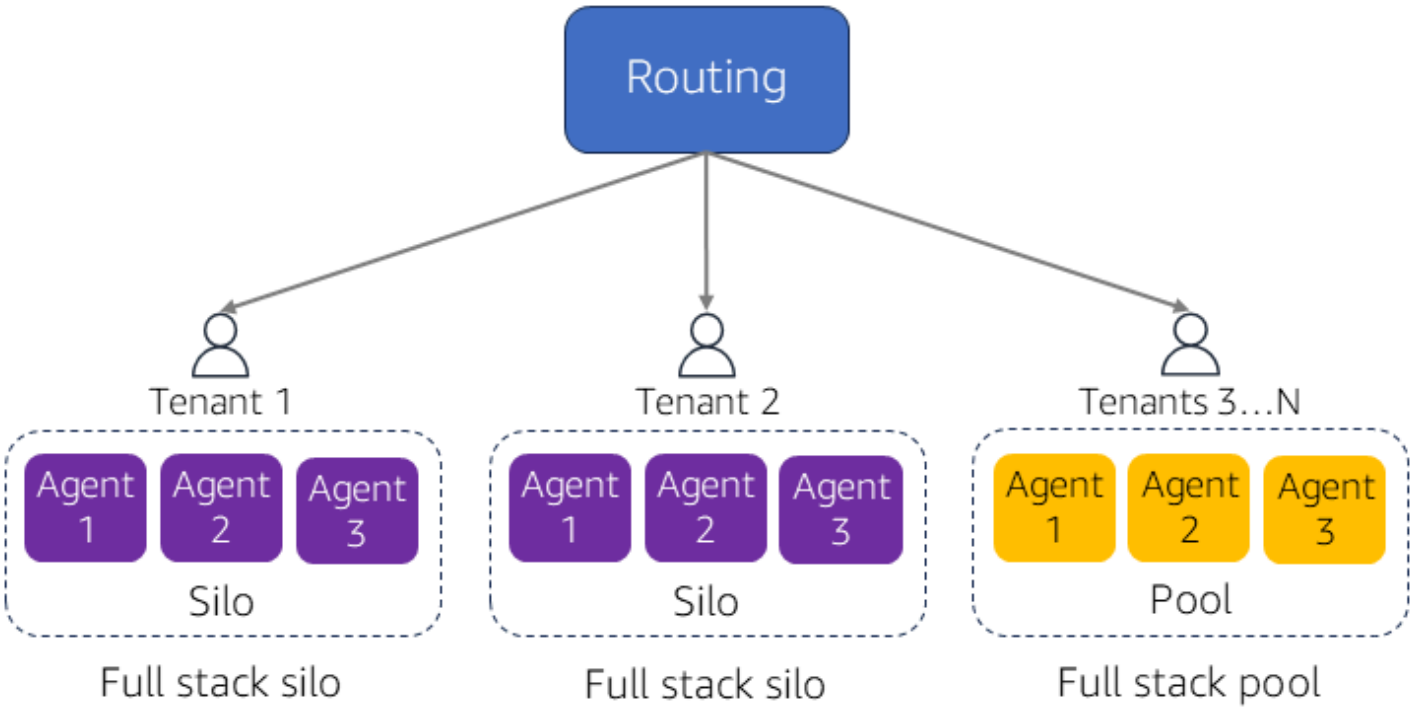
Questo diagramma rappresenta tre modalità di distribuzione degli agenti. Sul lato sinistro c'è un modello suddiviso in silos, in cui a ciascun tenant viene fornita un'esperienza completamente isolata e un set di agenti dedicato. In questo scenario, gli agenti non condividono l'elaborazione, le risorse o gli ambienti di esecuzione tra i tenant.

L'esempio centrale illustra un modello ibrido, in cui i tenant utilizzano una combinazione di agenti suddivisi in silos e in pool. Ad esempio, l'Agente 1 viene distribuito in modalità silos: ogni tenant riceve un'istanza dedicata, mentre gli agenti 2 e 3 operano in un modello condiviso, condividendo le risorse tra i tenant.

Sul lato destro c'è un modello completamente raggruppato, in cui tutti gli agenti sono condivisi tra i tenant, che offre una classica implementazione multi-tenant. In questo scenario, i tenant sfruttano un'infrastruttura di elaborazione, memoria e servizio comune per l'esecuzione degli agenti.

L'idea è che gli agenti possano operare in diversi modelli di implementazione, con risorse di calcolo e dipendenti dedicate (suddivise in silos) o condivise (raggruppate) tra i tenant. Queste strategie di implementazione non si escludono a vicenda. I servizi offerti agli agenti spesso soddisfano una vasta gamma di esigenze dei clienti, combinando entrambi i modelli per bilanciare prestazioni, isolamento,

costi e scalabilità. Il diagramma seguente mostra un sistema agentic che supporta più configurazioni di implementazione all'interno dello stesso ambiente operativo.



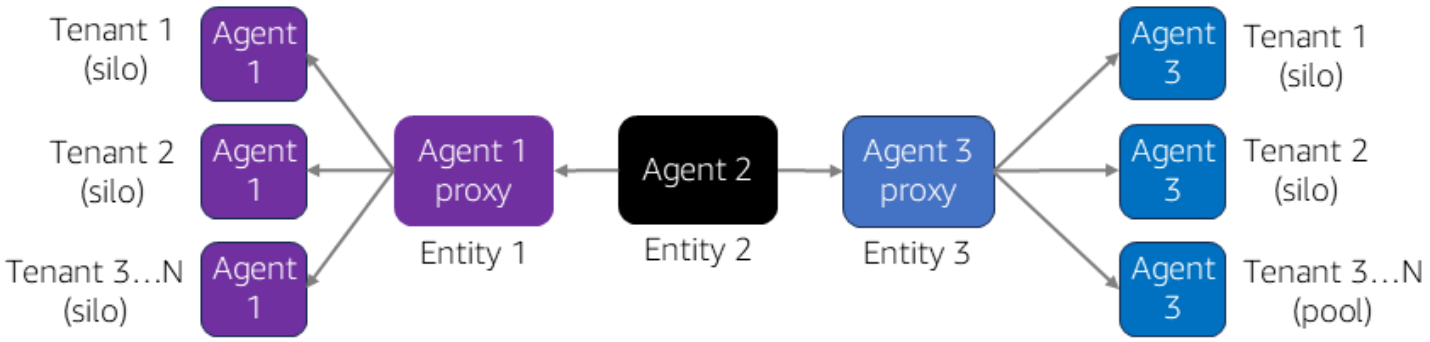
In questo diagramma, un provider di agenti dispone di tre agenti che vengono distribuiti tramite Agent as a Service (AaaS). Supportano due tipi di inquilini. Sul lato sinistro, due inquilini hanno requisiti di conformità e prestazioni che soddisfano attraverso un modello di silo completo. Il tenant rimanente sul lato destro funziona secondo un modello condiviso in cui i tenant condividono le risorse.

Se l'obiettivo è l'agilità e l'efficienza operativa, prova a limitare gli effetti associati al supporto di modelli di implementazione per-tenant. Ciò significa mettere in atto meccanismi di routing e altri meccanismi di esperienza che consentano la gestione, il funzionamento e l'implementazione degli agenti attraverso un unico pannello di controllo.

Se si crea un agente in un ambiente a basso o senza codice, non esisterà la nozione di agenti suddivisi in silos o in pool. Invece, gli agenti possono essere gestiti completamente da un altro agente. I modelli suddivisi in silos e in pool si applicano maggiormente agli ambienti in cui un'organizzazione controlla la struttura e l'impronta dell'agente. In questo caso, i team dovrebbero valutare quale modello di implementazione supportare.

In apparenza, questi modelli di implementazione non influiscono direttamente sul funzionamento di un agente in un sistema più ampio. Un agente può non avere alcuna conoscenza diretta degli altri agenti che vengono implementati in un modello a silo o in pool. Queste strategie di implementazione

possono invece essere implementate come parte di un costrutto di routing all'interno di un ambiente. Il diagramma seguente mostra un esempio di come è possibile implementare modelli suddivisi in silos e in pool utilizzando una strategia di routing.



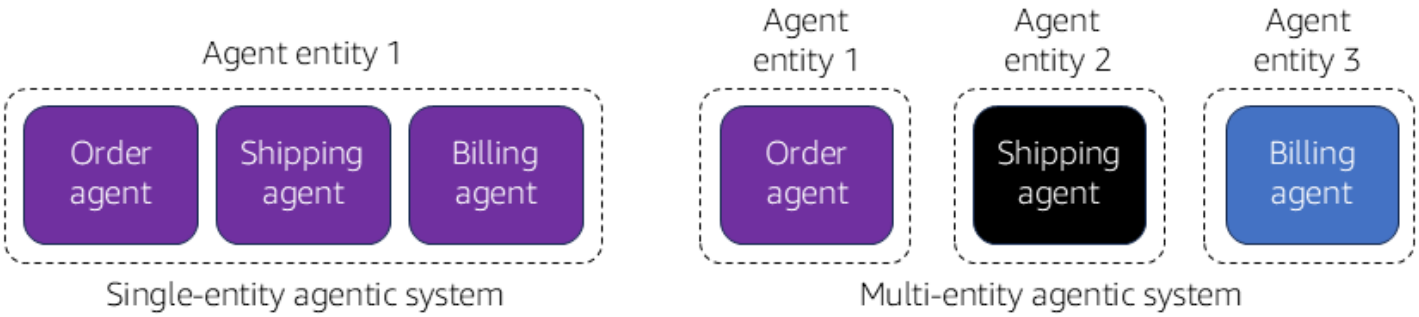
Questo esempio include tre agenti di tre provider diversi. Ogni fornitore di agenti ha la possibilità di implementare la propria strategia di implementazione. Ad esempio, l'agente 1 utilizza un proxy per distribuire le richieste in entrata a un set di tenant agent suddivisi in silos. L'Agente 2 non richiede alcun routing e supporta tutte le richieste dei tenant tramite un agente in pool. L'Agente 3 è una distribuzione di modello ibrido in cui alcuni tenant sono suddivisi in silos e altri in pool.

Se e come scegliete di supportare questi modelli di implementazione dipende dalla natura della soluzione. Potrebbe non essere necessario supportare nessuno dei due modelli. Tuttavia, potrebbero verificarsi casi in cui è necessario prendere in considerazione l'idea di supportare questa strategia, ad esempio per quanto riguarda la conformità, la rumorosità, le prestazioni o la suddivisione in più livelli.

Introduzione e applicazione del contesto del locatario

Se creiamo agenti che supportano il multi-tenancy, dobbiamo iniziare a considerare come impostare il contesto del tenant, che verrà utilizzato per applicare politiche, strategie e meccanismi specifici del tenant all'interno dell'implementazione dell'agente.

Al livello più elementare, è possibile introdurre il contesto dei tenant negli agenti tramite gli strumenti e i meccanismi comuni che utilizziamo nelle classiche architetture multi-tenant. Ciò potrebbe avvenire tramite una chiave API o vari altri meccanismi di OAuth convalida. Molti esempi di ciò si concentrano sulla risoluzione di un sistema o di un utente autenticato in una chiave JSON web token (JWT) che contiene il contesto del tenant. Il JWT viene quindi propagato attraverso il sistema. Ciò diventa più interessante se consideriamo come comporre sistemi agentici. Il diagramma seguente mostra un esempio di due varietà di ambienti agentici.



In questo diagramma, il modello sul lato sinistro rappresenta un sistema agentico in cui tutti gli agenti sono posseduti, gestiti e ospitati da una singola entità. Quando hai il pieno controllo dell'intera esperienza, puoi utilizzare le strategie tipiche per far passare gli inquilini a ciascun agente.

Il modello sul lato destro, che potrebbe essere più comune, rappresenta un sistema di agenti che si estendono su più entità. Gli agenti sono creati, gestiti e gestiti in modo indipendente, quindi ognuno di essi dispone dei propri schemi di autenticazione e autorizzazione. La sfida in questo caso è che abbiamo bisogno di un modo universale per risolvere e condividere il contesto degli inquilini tra questi agenti. Ciò si basa su un modello più distribuito in cui ogni agente deve essere in grado di autenticare i sistemi o gli utenti e trasferirli a un tenant in base ai meccanismi applicati.

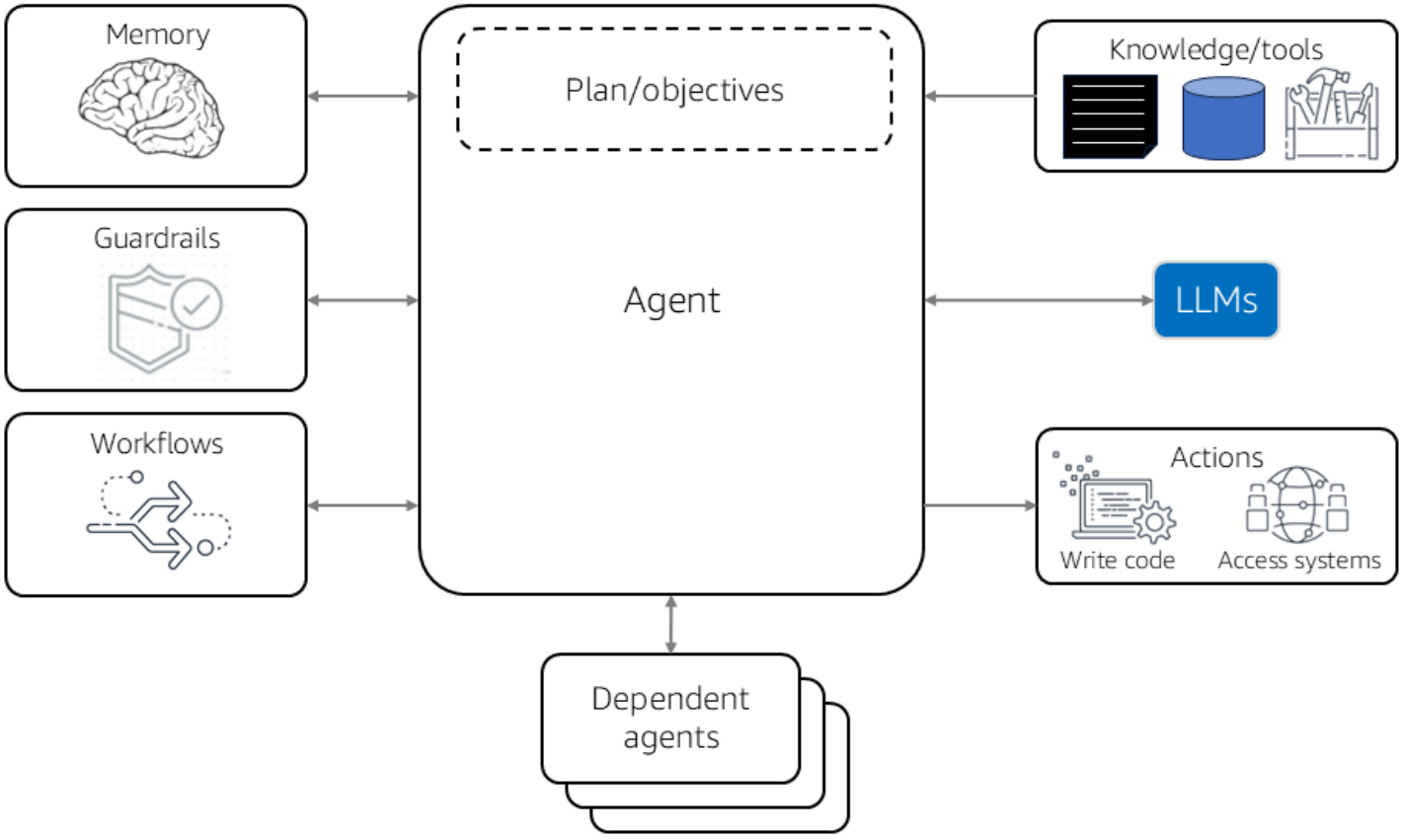
Creazione di agenti che riconoscono i tenant

La multi-tenancy influenza il modo in cui implementiamo i singoli agenti. Mentre un agente elabora le richieste, considera in che modo il contesto del tenant influisce sul modo in cui un agente accede ai

dati, prende decisioni e invoca azioni. Per comprendere meglio come e dove la multi-tenancy influisce sul profilo dell'agente, stabilite innanzitutto in che modo i costrutti possono far parte di qualsiasi agente.

La sfida è che l'ambito, la natura e la progettazione degli agenti sono tutt'altro che concreti, perché i fornitori fanno le proprie scelte in merito alla progettazione dell'esperienza degli agenti. In definitiva, il punto di forza di un agente è che si tratta di un servizio di apprendimento autonomo che può accedere a una serie di strumenti, fonti di dati e memoria per determinare il modo migliore per risolvere un compito.

È meno importante sapere esattamente quali strategie e modelli utilizza un agente. In un modello multi-tenant, è più importante identificare in che modo le varie parti di un agente sono configurate, accessibili e applicate. Prendi in considerazione un potenziale ambiente di agenti che si affida a una serie di risorse e meccanismi per raggiungere i propri obiettivi. Il diagramma seguente mostra un esempio di tale agente.



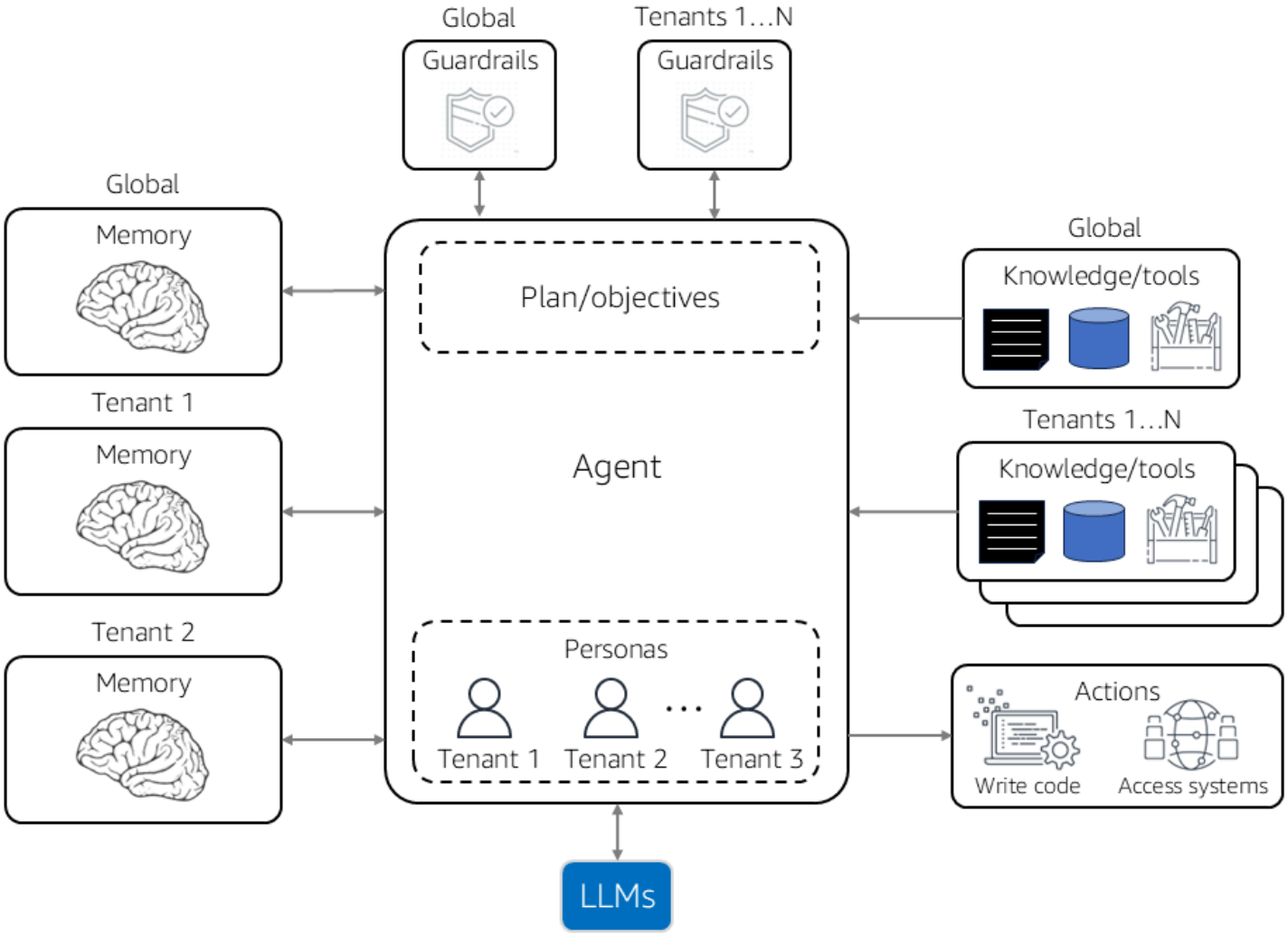
Questo diagramma rappresenta una gamma completa di possibilità agentiche e mostra vari strumenti e meccanismi che potrebbero essere combinati per raggiungere un obiettivo. Sul lato sinistro del diagramma, notate come un agente dipenda dalla memoria come parte del suo contesto, le barriere

per la definizione delle politiche che guidano le sue attività e i flussi di lavoro indirizzati a compiti specifici. Alcuni potrebbero obiettare che i flussi di lavoro non dovrebbero essere inclusi in questo contesto, ma potrebbero esserci scenari in cui i flussi di lavoro sono parte integrante di un'esperienza agentica.

La parte destra del diagramma mostra come input come conoscenze e strumenti possano fornire informazioni e contesto aggiuntivi che migliorano le capacità dell'agente. L'agente emette quindi azioni, come la scrittura di codice o l'accesso ai sistemi. La parte inferiore del diagramma mostra come gli agenti dipendono da uno o più agenti interni o di terze parti che possono essere orchestrati come parte di un sistema più ampio.

Ora possiamo pensare a cosa significhi introdurre la multi-tenancy. La locazione ci obbliga a considerare come e dove un agente introduce strategie e meccanismi che determinano comportamenti e azioni. Ciò aggiunge un'altra dimensione al modo in cui pensiamo agli agenti in termini di conoscenza, apprendimento, strumenti e memoria.

Vediamo ora come modificare questo modello per supportare la multi-tenancy. Il diagramma seguente mostra un esempio di modello multiagente.



In questo diagramma, introduciamo i tenant personas che hanno lo scopo di modellare il modo in cui un agente integra il contesto del tenant. Ad esempio, sul lato sinistro del diagramma, la memoria dell'agente viene modificata per supportare la memoria specifica del tenant. Lo stesso vale sul lato destro del diagramma, in cui l'agente supporta conoscenze e strumenti specifici del tenant. Lo stesso supporto viene applicato anche ai guardrail.

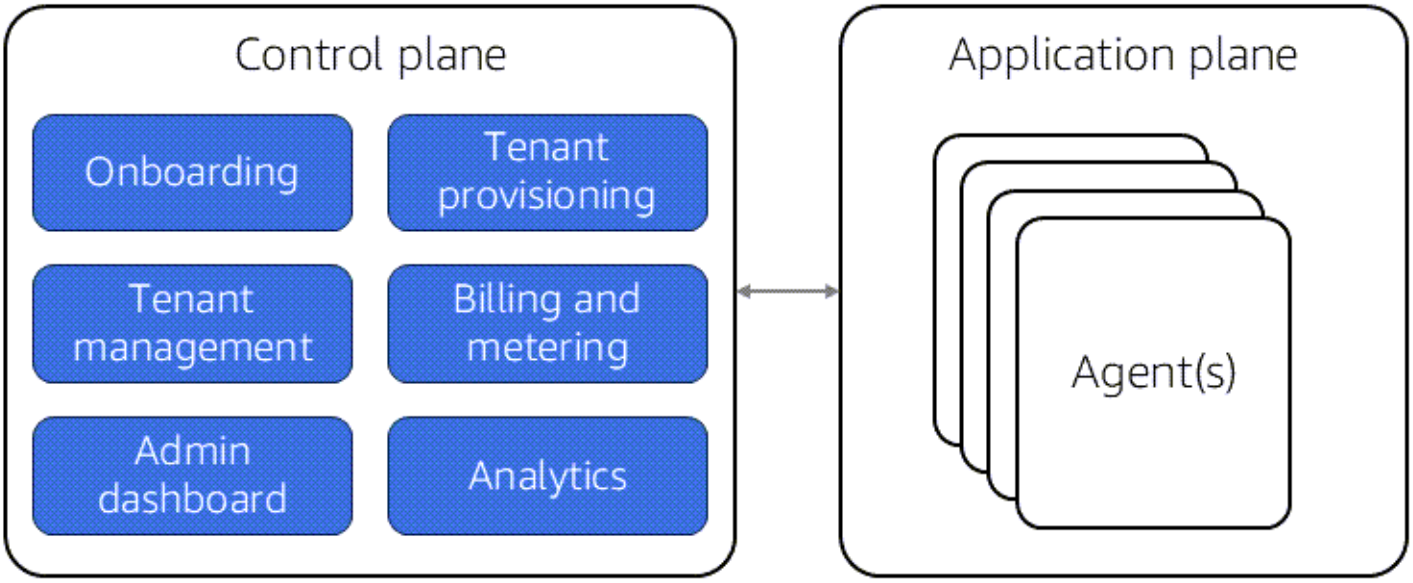
Questo può essere un esempio estremo, perché non tutti gli aspetti di un agente multi-tenant richiedono risorse specifiche per tenant. Il punto è che dovreste considerare in che modo personalizzare il vostro agente per inquilini specifici può migliorarne l'efficacia. Questo approccio consente all'agente di aumentarne l'impatto e il valore, fornire un contesto più pertinente nelle sue risposte e sviluppare capacità specializzate. L'agente sarà quindi in grado di apprendere, adattarsi ed eseguire attività che si adattano in modo esclusivo a diverse persone.

L'idea principale è che il contesto dei tenant influenzi direttamente il modo in cui si creano gli agenti. Può anche modellare le interazioni dei tenant con entità esterne, inclusi altri agenti. La creazione di un agente multi-tenant introduce sfide tradizionali come la rumorosità dei vicini, l'isolamento degli inquilini, la suddivisione in più livelli, la limitazione e la gestione dei costi. Il design e l'architettura del tuo agente devono rispondere a questi concetti fondamentali del multi-tenant, che esploreremo nella prossima sezione.

Utilizzo di piani di controllo in ambienti agentici

Le best practice multi-tenant spesso dividono le implementazioni in due parti distinte: un piano di controllo e un piano applicativo. Il piano di controllo fornisce un unico pannello di controllo per accedere ai meccanismi operativi, di gestione e di orchestrazione che coprono tutti i tenant dell'ambiente. Il piano applicativo è il luogo in cui risiedono la logica aziendale, le caratteristiche e le capacità funzionali.

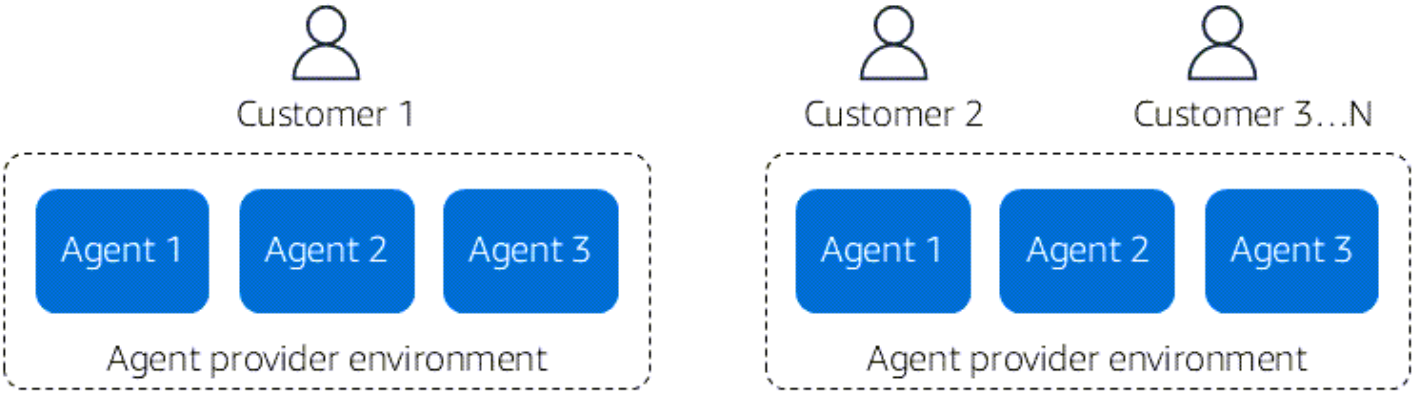
Questa divisione di responsabilità si applica anche ai modelli agentici. Un agente multi-tenant richiede un certo grado di gestione, funzionamento e approfondimenti centralizzati ed è opportuno soddisfare continuamente queste esigenze attraverso un piano di controllo. Il diagramma seguente mostra una visione concettuale di come questi piani sono suddivisi all'interno di un ambiente Agent as a Service (AaaS).



Questo diagramma mostra la tradizionale separazione dei piani di controllo e di applicazione. La novità è che il piano di controllo ora gestisce gli agenti che compongono un ambiente AaaS. Il piano di controllo interagisce con tutti gli agenti perché si presume che gli agenti siano creati, gestiti e distribuiti da un unico provider.

Questo modello introduce ulteriori livelli di complessità, in particolare nel ciclo di vita degli agenti e nel coordinamento con terze parti, ma mantiene la separazione fondamentale delle preoccupazioni. Il piano di controllo offre ancora le stesse funzionalità di base orchestrando la configurazione degli agenti, garantendo l'osservabilità di tenant e agent, raccogliendo dati di consumo e misurazione per la fatturazione e gestendo le politiche degli inquilini.

Questo scenario diventa più complesso se si considera un sistema multiagente che incorpora agenti di vari provider. Il diagramma seguente mostra un esempio di tale modello.



Questo diagramma mostra quattro agenti di diversi fornitori che fanno parte di un sistema multiagente. I provider di terze parti continuano a utilizzare e implementare ciascun agente, configurato per consentire l'accesso autorizzato da uno o più provider. Gli agenti, tuttavia, rimangono sotto il controllo del provider, quindi ogni agente mantiene il proprio piano di controllo.

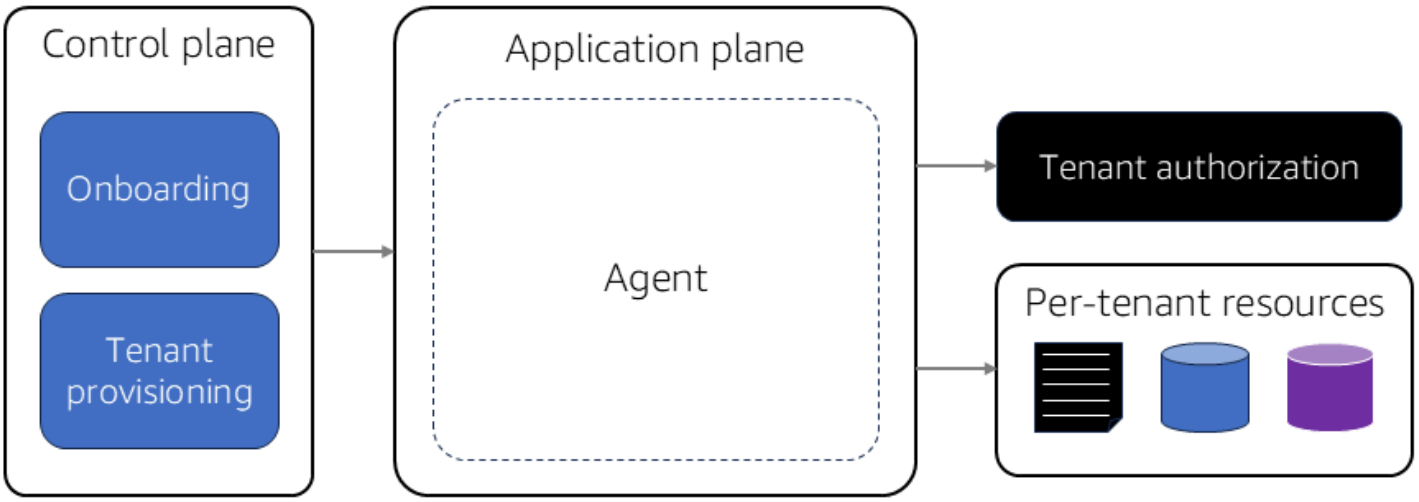
In sostanza, questi agenti multi-tenant si comportano come servizi di terze parti che si integrano con altri agenti. Pertanto, devono disporre di un proprio piano di controllo per fornire il funzionamento, la configurazione e la gestione centralizzati delle funzionalità di un agente.

Partiamo dal presupposto che gli agenti siano servizi indipendenti eseguiti in un'esperienza ospitata dal provider. Ma ciò potrebbe non essere chiaro in uno scenario in cui un consumatore agente impone maggiori vincoli su come e dove ospitare un agente.

Assegnazione degli inquilini agli agenti

L'onboarding è in genere una parte vitale di qualsiasi ambiente AaaS. Il modo in cui si creano, configurano e forniscono i tenant spesso coinvolgono molte parti mobili, integrazioni e strumenti. L'esperienza di onboarding degli agenti può richiedere gli stessi servizi disponibili in un piano di controllo AaaS, che include l'identità del tenant, il tiering, il provisioning delle risorse per tenant e la configurazione delle politiche dei tenant.

Il vostro approccio all'onboarding degli agenti è influenzato dall'impronta e dal modello di locazione del vostro ambiente agentico. Gli agenti suddivisi in silos e raggruppati hanno ciascuno le proprie sfumature e la scelta di utilizzare un solo agente o più agenti influisce anche sul processo di onboarding. Il diagramma seguente mostra una panoramica concettuale di come l'onboarding influenzi la configurazione di un agente.



Ogni volta che si aggiunge un agente, il piano di controllo deve adottare le misure necessarie per consentire all'inquilino di accedere all'agente. La modalità di introduzione dei tenant varia in base al modello di autorizzazione dell'agente, ma si supponga di creare un'identità del tenant che associ le richieste degli agenti ai singoli tenant. Questo contesto del tenant determina l'esperienza dell'agente applicandola a percorsi, ambiti e controllo degli accessi.

L'onboarding può anche richiedere la configurazione di tutte le risorse per-tenant utilizzate da un agente. È qui che il servizio di fornitura dei tenant del piano di controllo collega l'agente ai dati e alle risorse specifici del tenant consultati dall'agente.

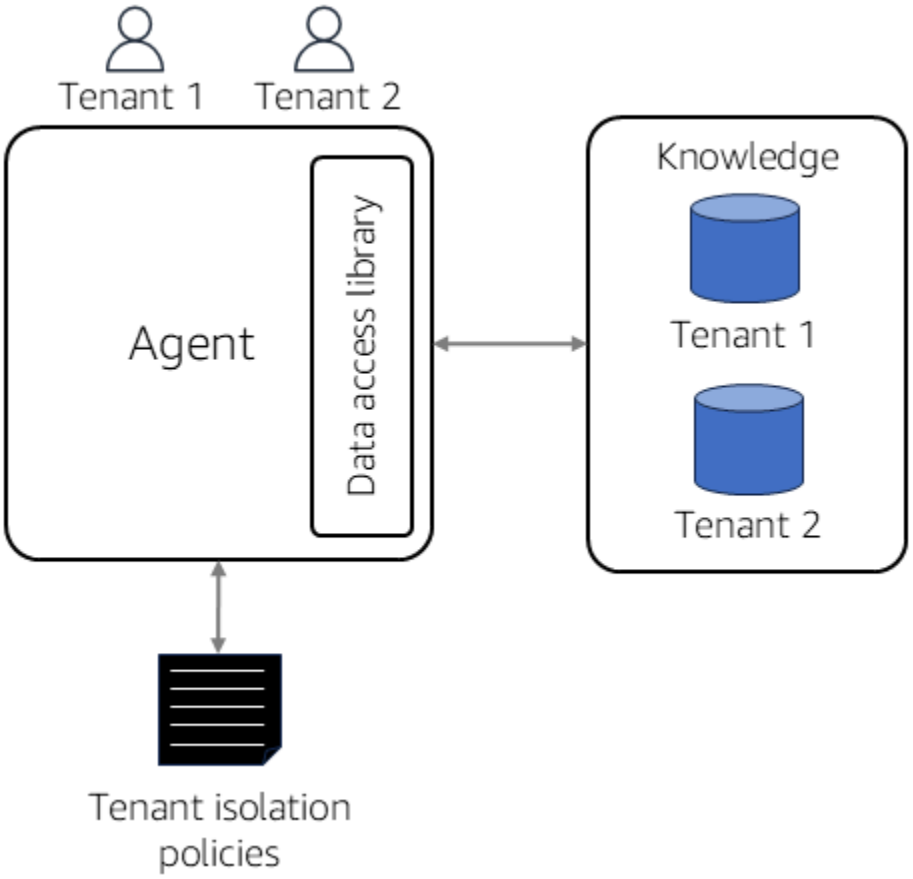
Se il sistema si basa sull'integrazione di agenti di terze parti, è necessario soddisfare anche le esigenze di tali agenti durante il processo di onboarding. Il modo in cui funziona dipende dai meccanismi di sicurezza e integrazione utilizzati per autorizzare l'accesso tra agenti. Idealmente, i passaggi necessari per orchestrare e configurare agent-to-agent l'autenticazione e l'autorizzazione vengono risolti tramite l'onboarding automatizzato.

Rafforzare l'isolamento degli inquilini

L'isolamento dei tenant è un concetto che si applica a tutte le impostazioni multi-tenant. Significa che le tue politiche e strategie garantiscono che un tenant non possa accedere alle risorse degli altri tenant. Per gli agenti multi-tenant, potrebbe essere necessario introdurre costrutti e meccanismi che aiutino a far rispettare i requisiti di isolamento dei tenant degli agenti.

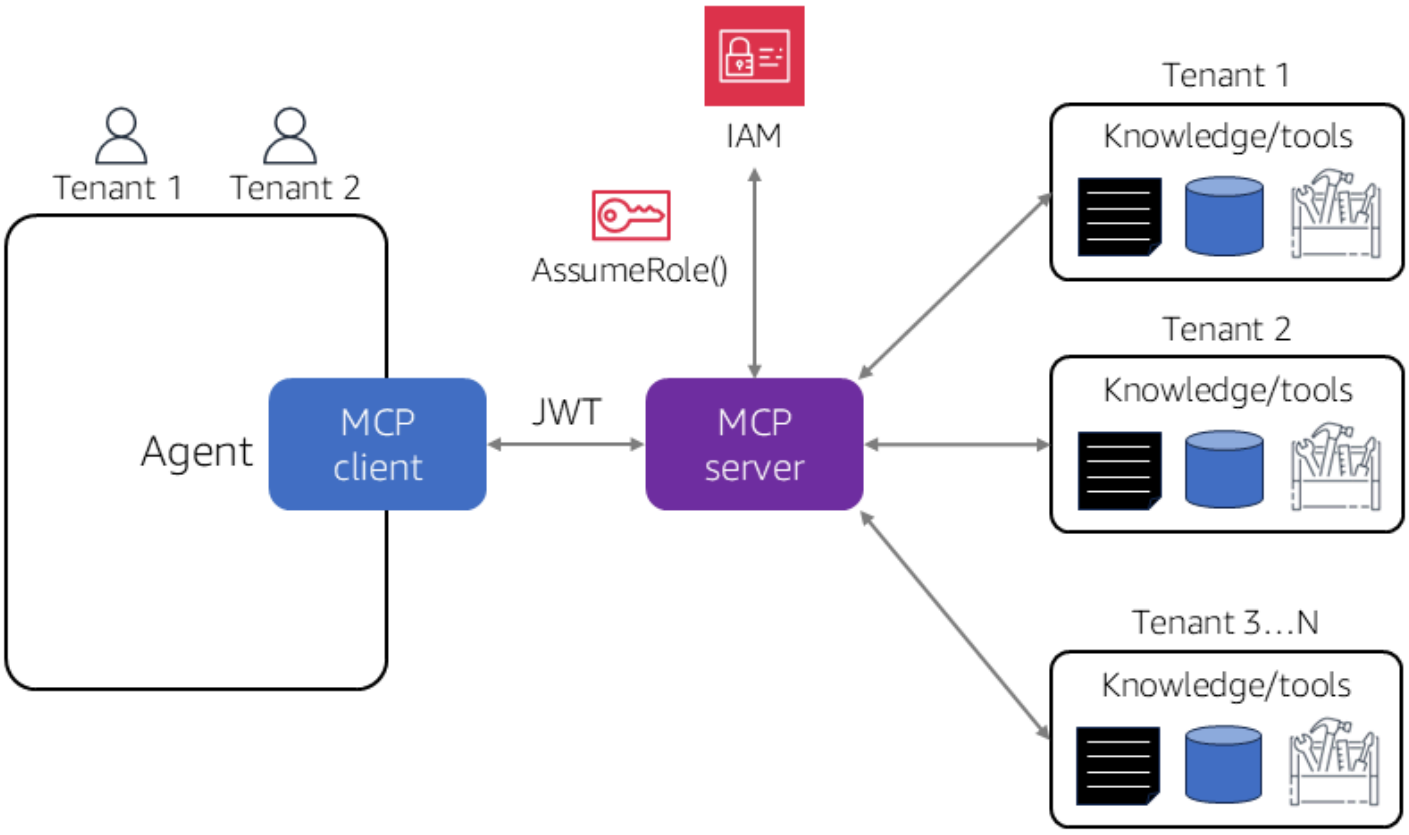
L'applicazione dell'isolamento dei tenant è come altre strategie che utilizzano sistemi multi-tenant tradizionali. In genere, quando costruite un'architettura AaaS, identificate qualsiasi area del sistema in cui una richiesta o un'azione può accedere alle risorse per determinare se la richiesta oltrepassa i limiti dei tenant. Ad esempio, i microservizi possono avere dipendenze da tabelle Amazon DynamoDB dedicate per tenant. Ciò richiede l'introduzione di politiche che garantiscano che la tabella di un tenant non sia accessibile a un altro tenant.

In questo caso, prendete in considerazione l'isolamento del tenant attraverso la lente di un agente e le sue interazioni con una qualsiasi delle sue risorse relative al tenant. Il diagramma seguente mostra un esempio concettuale di come gli agenti applicano le politiche di isolamento dei tenant per controllare l'accesso alle risorse dei tenant.



Sul lato destro di questo diagramma, l'agente dispone di informazioni relative al tenant archiviate in database vettoriali separati. Quando l'agente elabora una richiesta, esamina il contesto del tenant che la effettua. Sulla base di ciò, l'agente applica una politica di isolamento appropriata per garantire che agli inquilini sia impedito l'accesso ai dati o alle risorse al di fuori dei confini designati.

Se l'agente utilizza un Model Context Protocol (MCP), può anche implementare il modello di isolamento dei tenant. Il diagramma seguente mostra un esempio di come introdurre MCP e applicare politiche di isolamento.



MCP è un protocollo standardizzato che un agente utilizza per l'integrazione con qualsiasi strumento, dato e risorsa. In questo esempio, un client MCP e un server MCP interagiscono con le conoscenze e gli strumenti specifici del tenant mostrati sul lato destro del diagramma. Il contesto del tenant scorre dal client al server e il server utilizza questo contesto per acquisire credenziali con ambito tenant dal servizio (IAM). AWS Identity and Access Management Le credenziali controllano l'accesso alle risorse di ciascun tenant, garantendo che un tenant possa accedere alle risorse di un altro tenant.

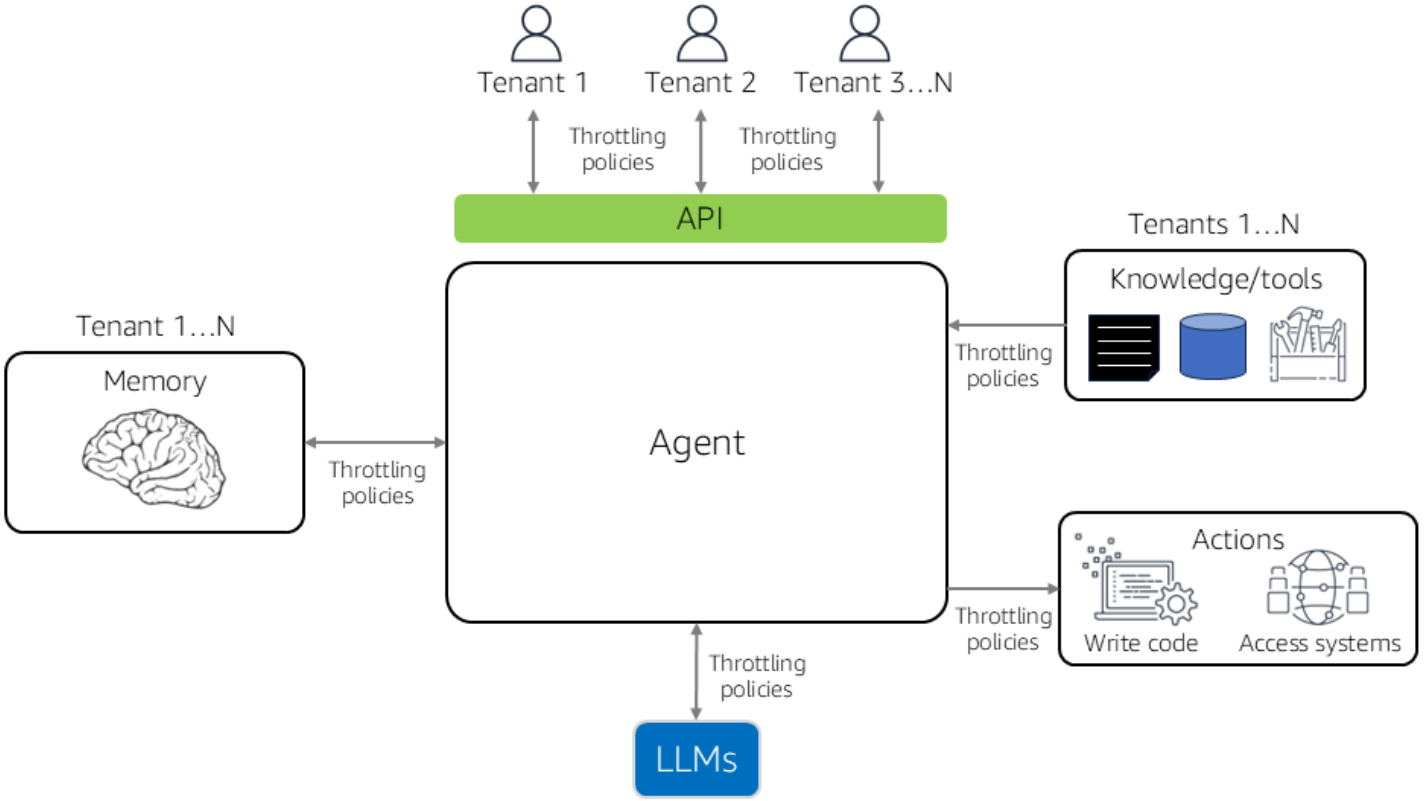
Poiché gli agenti incorporano la multi-tenancy, devono introdurre meccanismi che applichino le politiche di isolamento dei tenant durante l'elaborazione delle richieste. In alcuni casi, IAM può aiutare a limitare l'accesso alle risorse dei tenant. In altri casi, potrebbe essere necessario introdurre altri strumenti o framework per applicare le politiche di isolamento dei tenant.

Vicini e agenti rumorosi

In un ambiente AaaS multi-tenant in cui più inquilini condividono un agente, pensate a dove e come introdurre politiche che prevengano condizioni rumorose nei vicini. Le politiche possono introdurre una limitazione generica che si applica a tutti i consumi, oppure puoi avere politiche basate sugli

inquilini o sui livelli che applicano la limitazione in base a una determinata persona. Potresti imporre restrizioni di consumo maggiori agli inquilini di livello base rispetto agli inquilini di livello premium.

Questa nozione di limitazione può essere applicata a più punti dell'architettura. Il diagramma seguente mostra un esempio di alcune aree in cui è possibile introdurre politiche relative ai rumorosi vicini.



Nella nostra precedente analisi dell'implementazione multiagente, abbiamo esaminato diverse risorse che l'agente può utilizzare, evidenziando il potenziale di risorse pertenant all'interno di un agente. Ogni punto di contatto è una potenziale area in cui introdurre politiche di limitazione, che aiutano a garantire che gli inquilini non superino i limiti di consumo del sistema o le politiche di suddivisione in più livelli del tenant.

I luoghi migliori per introdurre protezioni contro i rumorosi vicini sono i punti dell'architettura in cui gli inquilini condividono le risorse. Questi componenti condivisi o raggruppati, come elaborazione, memoria e modelli linguistici di grandi dimensioni, sono i più suscettibili al degrado delle prestazioni se un singolo tenant consuma in modo sproporzionato. APIs

Un luogo naturale in cui applicare la limitazione è il punto di ingresso dell'agente, a volte chiamato «bordo esterno». Qui puoi introdurre limiti globali o di tenant-tier-based velocità prima che l'agente inizi a elaborare la richiesta. Il throttling può anche essere applicato più in profondità nel percorso di

esecuzione, ad esempio quando l'agente chiama un LLM, accede alla memoria o richiama strumenti condivisi.

Queste politiche aiutano a imporre un utilizzo equo, a mantenere la resilienza degli agenti sotto carico e a preservare un'esperienza coerente tra i tenant. A seconda dei tuoi obiettivi, potresti concentrarti sulla protezione generale del sistema (resilienza) o sulla gestione granulare dell'esperienza degli inquilini (ad esempio, con diritti basati su più livelli).

Dati, operazioni e test

Agenti e proprietà dei dati

Una revisione dell'implementazione degli agenti evidenzia gli scenari in cui un agente si affida ai dati di un determinato tenant. In questo caso, considera il ciclo di vita dei dati e, soprattutto, dove sono archiviati. Ciò è particolarmente importante per i settori e i casi d'uso in cui la natura dei dati influenza il modo in cui un agente vi accede.

I fornitori di servizi AaaS devono valutare come risolvere i problemi relativi ai dati in un ambiente multi-tenant, il che può influire sull'onboarding, sull'isolamento e sulle operazioni di un agente. Le sfumature e le strategie applicabili variano a seconda degli strumenti, delle tecnologie e dei dati utilizzati. È possibile affrontare questo problema in molti modi, il che è importante tenere presente quando si crea un'offerta AaaS.

Operazioni con agenti multi-tenant

Mentre crei ambienti con agenti, pensa a come far funzionare e gestire i tuoi agenti. In qualità di provider, hai bisogno di metriche, dati, approfondimenti e registri che ti consentano di monitorare lo stato, la scalabilità e l'attività di un agente. Ciò è più evidente in un ambiente agentico multi-tenant in cui è necessario comprendere in che modo i singoli inquilini consumano le risorse degli agenti.

Ciò è ancora più importante nelle impostazioni multi-agente, quando sono necessarie informazioni dettagliate sulle interazioni tra agenti. La possibilità di profilare e tracciare le attività tra gli agenti può essere essenziale per la risoluzione di problemi che influiscono sulla scalabilità, la precisione e l'efficacia del sistema.

I team operativi possono anche profilare le interazioni LLM per avere un'idea migliore del carico a cui sono sottoposti gli agenti. LLMs Questi dati sono essenziali per perfezionare l'implementazione degli agenti. Può inoltre fornire ai team operativi una panoramica di come gli agenti e la locazione influiscono sul profilo di costo complessivo di un sistema.

Formazione e test di agenti multi-tenant

Una sfida associata agli agenti edili è che ci si aspetta che imparino ed evolvano. Significa anche che dobbiamo testare il nostro agente, perfezionarlo e migliorarne la precisione prima di metterlo in

produzione. Esistono molte aree in cui è possibile ispezionare e valutare se l'agente sta valutando e classificando correttamente le intenzioni o scegliendo e invocando strumenti e azioni appropriati. L'elenco delle variabili è ampio, ma in ultima analisi si tratta di garantire che l'agente trovi risultati che consentano di raggiungere gli obiettivi prefissati.

L'esame di tutte le parti mobili e i principi associati agli agenti di test esula dallo scopo di questo documento, ma tenete presente che le strategie di test aggiungono complessità agli ambienti AAaS multi-tenant. Ad esempio, se un agente dispone di dati, memoria e altri costrutti che vengono applicati contestualmente a ciascun tenant, i risultati di un agente possono essere modellati dalle risorse del tenant.

Se si utilizza un agente per simulare uno scenario, potrebbe essere necessario espandere la simulazione per casi d'uso specifici del tenant. Di conseguenza, è necessario perfezionare le procedure di convalida per consentire i casi in cui i criteri di convalida differiscono per ogni tenant.

Considerazioni e discussioni

Dove si colloca il SaaS?

Gli esperti del settore discutono attivamente su come gli agenti influenzano il panorama del software as a service (SaaS). Se è vero che gli agenti stanno cambiando il software per molti sistemi, è esagerato suggerire che gli agenti rendano obsoleti i modelli di distribuzione. Alcuni fornitori SaaS saranno probabilmente sconvolti dall'adozione di agenti, mentre altri potrebbero ripensare completamente la loro proposta di valore, affidandosi a un modello Agent as a Service (AaaS). Altri potrebbero trovare un equilibrio introducendo selettivamente agenti per soddisfare esigenze specifiche.

Questo argomento è interessante perché l'adozione dei migliori principi SaaS può rappresentare la prossima evoluzione del SaaS. Ciò potrebbe significare che il SaaS sta prendendo piede o potrebbe significare che i principi fondamentali del SaaS vengono impacchettati e realizzati in un modello basato su agenti. Probabilmente è meno importante decidere dove finirà la terminologia, ma sembra improbabile che il concetto SaaS scompaia. È più probabile che gli agenti influenzino l'impronta SaaS.

In definitiva, dobbiamo decidere quali strategie possono essere applicate all'AaaS, il che significa consentire alle organizzazioni di adottare architetture agentiche e strategie aziendali in modo che i fornitori possano massimizzare l'efficienza, il valore e l'impatto dei loro sistemi agentici. Gli agenti non sono scatole nere. Gli agenti consumano risorse, scalano le operazioni, dipendono dai dati e generano costi, tutti fattori che i provider devono affrontare. I fornitori di agenti devono valutare in che modo i principi multi-tenant possono modellare l'offerta di servizi e ottimizzare i modelli operativi.

Discussione

Il panorama degli agenti continua a evolversi con progetti che variano in base ai domini, ai casi d'uso previsti e ai settori di destinazione. Parte di questa evoluzione include l'ulteriore perfezionamento della nostra visione delle strategie, dei modelli e dei compromessi che gli architetti considerano quando progettano e costruiscono agenti.

Una strategia completa per gli agenti deve essere in linea con gli obiettivi aziendali e tecnici. Ciò include la definizione dei mercati e dei personaggi target, la definizione di strategie di prezzi e gestione delle risorse e la determinazione del modo in cui gli agenti si adattano a sistemi più grandi.

Queste considerazioni sono particolarmente importanti quando si fornisce un servizio AaaS, dove la scalabilità, l'efficienza dei costi e l'innovazione sono obiettivi primari.

Le capacità operative sono altrettanto importanti. L'ambiente deve supportare il monitoraggio dell'attività degli agenti, delle metriche sullo stato di salute e dei modelli di utilizzo. Ciò diventa più complesso nei sistemi multiagente, in cui le operazioni devono essere coordinate tra agenti indipendenti.

Nel complesso, questa discussione sugli agenti non fa che somigliare alle varie considerazioni architettoniche che potrebbero far parte dei sistemi agentici. Oltre alla selezione degli strumenti e LLMs dei framework appropriati, il successo dipende dalla creazione di un'architettura che soddisfi i requisiti aziendali di scalabilità, efficienza, implementazione e multi-tenancy.

Cronologia dei documenti

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

Modifica	Descrizione	Data
Pubblicazione iniziale	—	14 luglio 2025

AWS Glossario delle linee guida prescrittive

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

Numeri

7 R

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

- Refactor/re-architect — Sposta un'applicazione e modificane l'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 Amazon PostgreSQL-Compatible Aurora.
- Ridefinire la piattaforma (lift and reshape): trasferisci un'applicazione nel cloud e introduci un certo livello di ottimizzazione per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale ad Amazon Relational Database Service (Amazon RDS) per Oracle in. Cloud AWS
- Riacquistare (drop and shop): passa a un prodotto diverso, in genere effettuando la transizione da una licenza tradizionale a un modello SaaS. Esempio: migra il tuo sistema di gestione delle relazioni con i clienti (CRM) su. Salesforce.com
- Eseguire il rehosting (lift and shift): trasferisci un'applicazione sul cloud senza apportare modifiche per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale su Oracle su un'istanza EC2 in. Cloud AWS
- Trasferire (eseguire il rehosting a livello hypervisor): trasferisci l'infrastruttura sul cloud senza acquistare nuovo hardware, riscrivere le applicazioni o modificare le operazioni esistenti. Esegui la migrazione dei server da una piattaforma locale a un servizio cloud per la stessa piattaforma. Esempio: migra un'applicazione suMicrosoft 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

A2A () Agent-to-Agent

Un protocollo statico per la collaborazione tra agenti che supporta la delega delle attività e il trasferimento dello stato.

ABAC

Vedi controllo degli accessi basato sugli attributi.

servizi astratti

Vedi servizi gestiti.

ACIDO

Vedi atomicità, consistenza, isolamento, durata.

migrazione attiva-attiva

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

migrazione attiva-passiva

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

Agente

Un sistema di intelligenza artificiale in grado di ragionare, pianificare e intraprendere azioni in modo autonomo utilizzando strumenti per raggiungere gli obiettivi.

Agente Ops

Pratiche operative per la creazione, il test, l'implementazione e l'esecuzione di agenti di intelligenza artificiale in produzione su larga scala.

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

controllo delle applicazioni

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

portfolio di applicazioni

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

intelligenza artificiale (IA)

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

operazioni di intelligenza artificiale (AIOps)

Il processo di utilizzo delle tecniche di machine learning per risolvere problemi operativi, ridurre gli incidenti operativi e l'intervento umano e aumentare la qualità del servizio. Per ulteriori

informazioni su come viene utilizzato AIOps nella strategia di migrazione AWS , 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 for 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 disturbare 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.

blue/green dispiegamento

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, consulta l'indicatore Implementare le procedure break-glass nella guida. AWS Well-Architected

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, distribuisci la nuova versione e sostituisci la versione corrente nella sua interezza.

CoE

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.

Sviluppatore cittadino

Un utente aziendale che crea applicazioni di intelligenza artificiale utilizzando piattaforme senza code/low codice senza competenze tecniche specializzate.

crittografia lato client

Crittografia dei dati localmente, prima che il bersaglio li Servizio AWS riceva.

centro di eccellenza del 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 i post di CCoE 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 dimensionare l'adozione del cloud (ad esempio, creazione di una zona di destinazione, definizione di un CCoE, definizione di un modello operativo)
- Migrazione: migrazione di singole applicazioni
- Re-invention — Ottimizzazione di prodotti e servizi e innovazione nel cloud

Queste fasi sono state definite da Stephen Orban nel post del blog The Journey Toward Cloud-First & the Stages of Adoption sul blog Enterprise Strategy. Cloud AWS Per informazioni sulla loro relazione con la strategia di AWS migrazione, consulta la guida alla preparazione alla migrazione.

CMDB

Vedi database di gestione della configurazione.

repository di codice

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

Il processo di automazione delle fasi di origine, compilazione, test, gestione temporanea e produzione del processo di rilascio del software. CI/CD viene comunemente descritto come una pipeline. CI/CD può aiutarvi ad automatizzare i processi, migliorare la produttività, migliorare la qualità del codice e velocizzare le consegne. Per ulteriori informazioni, consulta Vantaggi della distribuzione continua. CD può anche significare continuous deployment (implementazione continua). Per ulteriori informazioni, consulta Distribuzione continua e implementazione continua a confronto.

CV

Vedi visione artificiale.

D

dati a riposo

Dati stazionari nella rete, ad esempio i dati archiviati.

classificazione dei dati

Un processo per identificare e classificare i dati nella rete in base alla loro criticità e sensibilità. È un componente fondamentale di qualsiasi strategia di gestione dei rischi di sicurezza informatica

perché consente di determinare i controlli di protezione e conservazione appropriati per i dati. La classificazione dei dati è un componente del pilastro della sicurezza nel AWS Well-Architected Framework. Per ulteriori informazioni, consulta Classificazione dei dati.

deriva dei dati

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

dati in transito

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

rete di dati

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

riduzione al minimo dei dati

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

perimetro dei dati

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

pre-elaborazione dei dati

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

provenienza dei dati

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

soggetto dei dati

Un individuo i cui dati vengono raccolti ed elaborati.

data warehouse

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

linguaggio di definizione del database (DDL)

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

linguaggio di manipolazione del database (DML)

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

DDL

Vedi linguaggio di definizione del database.

deep ensemble

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

deep learning

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

difesa in profondità

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 approccio di difesa approfondita potrebbe combinare autenticazione a più fattori, segmentazione della rete e 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 Workload 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 lo strangler fig pattern, consulta Modernizzare i servizi Web Microsoft ASP.NET (ASMX) legacy in modo incrementale utilizzando contenitori e Amazon API Gateway.

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. Big-endian i sistemi memorizzano per primi il byte più importante. Little-endian i sistemi memorizzano per primi il byte meno importante.

endpoint

Vedi service endpoint.

servizio endpoint

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

pianificazione delle risorse aziendali (ERP)

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

crittografia envelope

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

ambiente

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

- ambiente di sviluppo: un'istanza di un'applicazione in esecuzione disponibile solo per il team principale responsabile della manutenzione dell'applicazione. Gli ambienti di sviluppo vengono utilizzati per testare le modifiche prima di promuoverle negli ambienti superiori. Questo tipo di ambiente viene talvolta definito ambiente di test.
- ambienti inferiori: tutti gli ambienti di sviluppo di un'applicazione, ad esempio quelli utilizzati per le build e i test iniziali.
- ambiente di produzione: un'istanza di un'applicazione in esecuzione a cui gli utenti finali possono accedere. In una CI/CD pipeline, l'ambiente di produzione è l'ultimo ambiente di distribuzione.
- ambienti superiori: tutti gli ambienti a cui possono accedere utenti diversi dal team di sviluppo principale. Si può trattare di un ambiente di produzione, ambienti di preproduzione e ambienti per i test di accettazione da parte degli utenti.

epica

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

ERP

Vedi pianificazione delle risorse aziendali.

analisi esplorativa dei dati (EDA)

Il processo di analisi di un set di dati per comprenderne le caratteristiche principali. Si raccolgono o si aggregano dati e quindi si eseguono indagini iniziali per trovare modelli, rilevare anomalie

e verificare ipotesi. L'EDA viene eseguita calcolando statistiche di riepilogo e creando visualizzazioni di dati.

F

tabella dei fatti

Il tavolo centrale 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. Few-shot i suggerimenti 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. Le FM sono in grado di eseguire 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.

Gateway FM

Un intermediario centralizzato che controlla e normalizza l'accesso ai modelli di base. Conosciuto anche come gateway LLM.

G

IA generativa

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

blocco geografico

Vedi restrizioni geografiche.

limitazioni geografiche (blocco geografico)

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

Flusso di lavoro di GitFlow

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

immagine dorata

Un'istantanea di un sistema o di un software 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 livello elevato che consente di governare risorse, policy e conformità tra le unità organizzative (OU). I guardrail preventivi applicano le policy per garantire l'allineamento agli

standard di conformità. Vengono implementati utilizzando le policy di controllo dei servizi e i limiti delle autorizzazioni IAM. I guardrail di rilevamento rilevano le violazioni delle policy e i problemi di conformità e generano avvisi per porvi rimedio. Sono implementati utilizzando Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector e controlli personalizzati AWS Lambda .

guardrail (AI)

Meccanismi di sicurezza che filtrano, convalidano e limitano gli input e gli output degli agenti per contribuire a garantire un comportamento dell'IA responsabile e sicuro.

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.

human-in-the-loop (HITL)

Un modello di flusso di lavoro in cui l'esecuzione degli agenti viene sospesa per la revisione e l'approvazione umana nei punti decisionali critici.

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, ad esempio dati in tempo reale o dati di traduzione recenti. Questi dati richiedono in genere un livello o una classe di storage ad alte prestazioni per fornire risposte rapide alle query.

hotfix

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

periodo di hypercare

Subito dopo la conversione, il periodo di tempo in cui un team di migrazione gestisce e monitora le applicazioni migrate nel cloud per risolvere eventuali problemi. In genere, questo periodo dura da 1 a 4 giorni. Al termine del periodo di hypercare, il team addetto alla migrazione in genere trasferisce la responsabilità delle applicazioni al team addetto alle operazioni cloud.

I

IaC

Vedi l'infrastruttura come codice.

Policy basata su identità

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

applicazione inattiva

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

IIoT

Vedi Industrial Internet of Things.

infrastruttura immutabile

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

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. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

migrazione incrementale

Una strategia di conversione in cui si esegue la migrazione dell'applicazione in piccole parti anziché eseguire una conversione singola e completa. Ad esempio, inizialmente potresti spostare solo alcuni microservizi o utenti nel nuovo sistema. Dopo aver verificato che tutto funzioni correttamente, puoi spostare in modo incrementale microservizi o utenti aggiuntivi fino alla disattivazione del sistema legacy. Questa strategia riduce i rischi associati alle migrazioni di grandi dimensioni.

Industria 4.0

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

infrastruttura

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

infrastruttura come codice (IaC)

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

Internet delle cose industriale (IIoT)

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, consulta Creazione di una strategia di trasformazione digitale dell'Internet delle cose industriale (IIoT).

VPC di ispezione

In un'architettura AWS multi-account, un VPC centralizzato che gestisce le ispezioni del traffico di rete tra VPC (uguali o diversi Regioni AWS), Internet e reti locali. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione 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, consulta 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 gli LLM.

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.

MCP

Vedi Model Context Protocol.

Model Context Protocol (MCP)

Un protocollo stateless per la comunicazione tra agenti e strumenti.

Server MCP

Un servizio che espone uno o più strumenti tramite il Model Context Protocol.

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, vedete 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 da macchina a macchina (M2M) leggero, basato sul publish/subscribemodello, per dispositivi IoT con risorse limitate.

microservizio

Un piccolo servizio indipendente che comunica tramite API ben definite ed è in genere di proprietà di piccoli team autonomi. 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 tramite un'interfaccia ben definita utilizzando API leggere. 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

Cross-functional team che semplificano la migrazione dei carichi di lavoro attraverso approcci automatizzati e agili. I team di Migration Factory includono in genere operazioni, analisti e

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

metadati di migrazione

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

modello di migrazione

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

Valutazione del portfolio di migrazione (MPA)

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

valutazione della preparazione alla migrazione (MRA)

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

strategia di migrazione

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

ML

Vedi machine learning.

modernizzazione

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

valutazione della preparazione alla modernizzazione

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

applicazioni monolitiche (monoliti)

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

MAPPA

Vedi Migration Portfolio Assessment.

MQTT

Vedi Message Queuing Telemetry Transport.

classificazione multiclasse

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

infrastruttura mutabile

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

O

OAC

Vedi Origin Access Control.

QUERCIA

Vedi Origin Access Identity.

OCM

Vedi gestione delle modifiche organizzative.

migrazione offline

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

OI

Vedi l'integrazione delle operazioni.

OLA

Vedi accordo a livello operativo.

migrazione online

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

OPC-UA

Vedi Open Process Communications - Unified Architecture.

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

Un protocollo di comunicazione da macchina a macchina (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 Framework. AWS Well-Architected

tecnologia operativa (OT)

Sistemi hardware e software che interagiscono con l'ambiente fisico per controllare operazioni, apparecchiature e 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 in tutto tutti i bucket S3 Regioni AWS, la crittografia lato server con AWS KMS (SSE-KMS) e le richieste dinamiche PUT e dirette al bucket S3. 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. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

P

limite delle autorizzazioni

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

informazioni di identificazione personale (PII)

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

Informazioni che consentono l'identificazione personale degli utenti

Visualizza le informazioni di identificazione personale.

playbook

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

PLC

Vedi controllore logico programmabile.

PLM

Vedi la gestione del ciclo di vita del prodotto.

policy

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

persistenza poliglotta

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

valutazione del portfolio

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

predicate

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

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 ingegneristico dei sistemi che tiene conto della privacy durante l'intero processo di sviluppo.

zone ospitate private

Un container che contiene informazioni su come si desidera che Amazon Route 53 risponda alle query DNS per un dominio e i relativi sottodomini all'interno di uno o più VPC. Per ulteriori informazioni, consulta Utilizzo delle zone ospitate private nella documentazione di Route 53.

controllo proattivo

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

gestione del ciclo di vita del prodotto (PLM)

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

Ambiente di produzione

Vedi ambiente.

controllore logico programmabile (PLC)

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

concatenamento rapido

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

pseudonimizzazione

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

publish/subscribe (pub/sub)

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

Q

Piano di query

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

regressione del piano di query

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

R

Matrice RACI

Vedi responsabile, responsabile, consultato, informato (RACI).

RAG

Vedi Retrieval Augmented Generation.

ransomware

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

Matrice RASCI

Vedi responsabile, responsabile, consultato, informato (RACI).

RCAC

Vedi controllo dell'accesso a righe e colonne.

replica di lettura

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

riprogettare

Vedi 7 Rs.

obiettivo del punto di ripristino (RPO)

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

obiettivo del tempo di ripristino (RTO)

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

rifattorizzare

Vedi 7 R.

Region

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

regressione

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

riospitare

Vedi 7 R.

rilascio

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

trasferisco

Vedi 7 Rs.

ripiattaforma

Vedi 7 Rs.

riacquisto

Vedi 7 Rs.

resilienza

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

policy basata su risorse

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

matrice di assegnazione di responsabilità (RACI)

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

controllo reattivo

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

retain

Vedi 7 R.

andare in pensione

Vedi 7 Rs.

Retrieval Augmented Generation (RAG)

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

rotazione

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

controllo dell'accesso a righe e colonne (RCAC)

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

RPO

Vedi obiettivo del punto di ripristino.

VERSO

Vedi obiettivo del tempo di ripristino.

runbook

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

S

SAML 2.0

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

SCADA

Vedi controllo di supervisione e acquisizione dati.

SCP

Vedi la politica di controllo del servizio.

Secret

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

sicurezza fin dalla progettazione

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

controllo di sicurezza

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

rafforzamento della sicurezza

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

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

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

automazione della risposta alla sicurezza

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

Crittografia lato server

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

Policy di controllo dei servizi (SCP)

Una policy che fornisce il controllo centralizzato sulle autorizzazioni per tutti gli account di un'organizzazione in AWS Organizations. Le SCP definiscono i guardrail o fissano i limiti alle azioni che un amministratore può delegare a utenti o ruoli. Puoi utilizzare le SCP 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.

Shadow AI

Applicazioni di intelligenza artificiale non autorizzate create o utilizzate al di fuori dei canali regolamentati all'interno di un'organizzazione.

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.

modello split-and-seed

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 Modernizzare i servizi Web Microsoft ASP.NET (ASMX) legacy in modo incrementale utilizzando contenitori e Amazon API Gateway.

sottorete

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

controllo di supervisione e acquisizione dati (SCADA)

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

crittografia simmetrica

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

test sintetici

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

prompt di sistema

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

T

tag

Key-value coppie 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.

strumento

Una funzione o API che un agente può richiamare per eseguire operazioni in sistemi esterni.

Transit Gateway

Un hub di transito di rete che è possibile utilizzare per collegare i VPC e le reti on-premise. Per ulteriori informazioni, consulta Cos'è un gateway di transito nella AWS Transit Gateway documentazione.

flusso di lavoro basato su trunk

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

Accesso attendibile

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

regolazione

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

team da due pizze

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

U

incertezza

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

compiti indifferenziati

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

ambienti superiori

Vedi ambiente.

V

vacuum

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

controllo delle versioni

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

Peering VPC

Una connessione tra due VPC che consente di instradare il traffico tramite 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 interrogazioni 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.