Fondamenti dell'intelligenza artificiale agentica su AWS

AWS Guida prescrittiva

AWS Guida prescrittiva: Fondamenti dell'intelligenza artificiale agentica su AWS

Table of Contents

Fondamenti dell'intelligenza artificiale agentica su AWS

Aaron Sempf, Amazon Web Services

Luglio 2025 (cronologia del documento)

In un mondo di sistemi sempre più intelligenti, distribuiti e autonomi, il concetto di agente, un'entità in grado di percepire il proprio ambiente, ragionare sul proprio stato e agire con intenzione, è diventato fondamentale. Gli agenti non sono semplicemente programmi che eseguono istruzioni; sono entità orientate agli obiettivi e consapevoli del contesto che prendono decisioni per conto di utenti, sistemi o organizzazioni. La loro comparsa riflette un cambiamento nel modo in cui si crea e si pensa al software: un passaggio dalla logica procedurale e dall'automazione reattiva a sistemi che operano con autonomia e finalità.

All'intersezione tra intelligenza artificiale, sistemi distribuiti e ingegneria del software si colloca un potente paradigma noto come intelligenza artificiale agentica. Questa nuova generazione di sistemi intelligenti è costituita da agenti software in grado di adottare comportamenti adattivi, coordinazione complessa e processi decisionali delegati.

Questa guida introduce i principi che definiscono gli agenti software moderni e delinea la loro evoluzione verso l'intelligenza artificiale agentica. Per spiegare questo cambiamento, la guida fornisce il background concettuale e poi ripercorre l'evoluzione degli agenti software verso l'intelligenza artificiale agentica:

- Introduzione agli agenti software definisce gli agenti software, li confronta con i componenti software tradizionali e introduce le caratteristiche essenziali che differenziano il comportamento degli agenti dall'automazione tradizionale attingendo a framework consolidati.
- Lo scopo degli agenti software esamina il motivo per cui esistono gli agenti software, quali ruoli ricoprono, quali problemi risolvono e come consentono la delega intelligente, riducono il carico cognitivo e supportano il comportamento adattivo in ambienti dinamici.
- L'evoluzione degli agenti software ripercorre le tappe intellettuali e tecnologiche che hanno dato forma agli agenti software, dai primi concetti di autonomia e concorrenza all'emergere di sistemi multiagente e architetture formali di agenti, con conseguente convergenza con l'IA generativa.
- Gli agenti software all'intelligenza artificiale agentica introducono l'intelligenza artificiale agentica come culmine di decenni di progressi che combina modelli di agenti distribuiti con modelli di base, elaborazione senza server e protocolli di orchestrazione. Questa sezione descrive come questa

convergenza consenta una nuova generazione di agenti intelligenti che utilizzano strumenti che operano con autonomia, asincronicità e una vera agenzia su larga scala.

Destinatari principali

Questa guida è pensata per architetti, sviluppatori e leader tecnologici che desiderano comprendere la storia, i concetti principali e l'evoluzione degli agenti software verso l'intelligenza artificiale agentica prima di adottare questa tecnologia per le moderne soluzioni cloud su. AWS

Obiettivi

L'adozione di architetture agentiche aiuta le organizzazioni a:

- Accelera il time-to-value: automatizza e scala il knowledge work e riduci il lavoro manuale e la latenza.
- Migliora il coinvolgimento dei clienti: offri assistenti intelligenti in tutti i domini.
- Riduzione dei costi operativi: automatizza i flussi decisionali che in precedenza richiedevano l'input o la supervisione umani.
- Promuovi l'innovazione e la differenziazione: crea prodotti intelligenti che si adattano, apprendono e competono in tempo reale.
- Modernizza i flussi di lavoro esistenti: ristruttura gli script e i monoliti in agenti di ragionamento modulari.

Informazioni su questa serie di contenuti

Questa guida fa parte di una serie di pubblicazioni che forniscono progetti architettonici e linee guida tecniche per la creazione di agenti software basati sull'intelligenza artificiale. AWS La serie include quanto segue:

- Rendere operativa l'intelligenza artificiale agentica su AWS
- Fondamenti dell'intelligenza artificiale agentica su (questa guida) AWS
- Modelli e flussi di lavoro di intelligenza artificiale agentica su AWS
- Framework, protocolli e strumenti di intelligenza artificiale agentica su AWS
- Creazione di architetture serverless per l'intelligenza artificiale agentica su AWS

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

Per ulteriori informazioni su questa serie di contenuti, consulta Agentic AI.

Introduzione agli agenti software

Il concetto di agenti software si è evoluto in modo significativo dalle sue fondamenta in entità autonome negli anni '60 alla sua esplorazione formale all'inizio degli anni '90. Man mano che i sistemi digitali diventano sempre più complessi, dagli script deterministici alle applicazioni adattive e intelligenti, gli agenti software sono diventati elementi costitutivi essenziali per consentire comportamenti autonomi, sensibili al contesto e orientati agli obiettivi nei sistemi informatici. Nel contesto delle architetture native del cloud e potenziate dall'intelligenza artificiale, in particolare con l'avvento dell'intelligenza artificiale generativa, dei modelli linguistici di grandi dimensioni () e di piattaforme come Amazon BedrockLLMs, gli agenti software vengono ridefiniti attraverso nuove lenti di capacità e scalabilità.

Questa introduzione si basa sull'opera fondamentale Software Agents: An Overview di Hyacinth S. Nwana (Nwana 1996). Definisce gli agenti software, ne discute le radici concettuali ed estende la discussione a un quadro contemporaneo per definire tre principi generali degli agenti software moderni: autonomia, asincronicità e agenzia. Questi principi distinguono gli agenti software da altri tipi di servizi o applicazioni e consentono a questi agenti di operare con finalità, resilienza e intelligenza in ambienti distribuiti e in tempo reale.

In questa sezione

Dall'autonomia all'intelligenza distribuita

Prima che il termine agente software entrasse a far parte della corrente corrente, le prime ricerche informatiche esploravano l'idea di entità digitali autonome, ossia sistemi in grado di agire in modo indipendente, reagire agli input e prendere decisioni sulla base di regole o obiettivi interni. Queste prime idee gettarono le basi concettuali per quello che sarebbe diventato il paradigma degli agenti. (Per una cronologia storica, consultate la sezione L'evoluzione degli agenti software più avanti in questa guida.)

Primi concetti di autonomia

La nozione di macchine o programmi che agiscono indipendentemente dagli operatori umani affascina i progettisti di sistemi da decenni. I primi lavori nel campo della cibernetica, dell'intelligenza artificiale e dei sistemi di controllo hanno esaminato come il software potesse esibire un comportamento autoregolante, rispondere dinamicamente ai cambiamenti e funzionare senza la continua supervisione umana.

Queste idee hanno introdotto l'autonomia come attributo fondamentale dei sistemi intelligenti e hanno posto le basi per l'emergere di software in grado di decidere e agire, anziché limitarsi a reagire o eseguire.

Il modello dell'attore e l'esecuzione asincrona

Negli anni '70, il modello dell'attore, introdotto nel paper A Universal Modular ACTOR Formalism for Artificial Intelligence (Hewitt et al. 1973), forniva un quadro formale per pensare al calcolo decentralizzato e basato sui messaggi. In questo modello, gli attori sono entità indipendenti che comunicano esclusivamente tramite messaggi asincroni e consentono sistemi scalabili, concorrenti e tolleranti ai guasti.

Il modello dell'attore ha enfatizzato tre attributi chiave che continuano a influenzare la progettazione degli agenti moderni:

- Isolamento dello stato e del comportamento
- Interazione asincrona tra entità
- Creazione dinamica e delega di attività

Questi attributi erano in linea con le esigenze dei sistemi distribuiti e prefiguravano le caratteristiche operative degli agenti software negli ambienti nativi del cloud.

Intelligenza distribuita e sistemi multiagente

Man mano che i sistemi informatici diventavano più interconnessi dopo gli anni '60, i ricercatori hanno esplorato l'intelligenza artificiale distribuita (DAI). Questo campo si è concentrato su come più entità autonome potessero lavorare in modo collaborativo o competitivo all'interno di un sistema. DAI ha portato allo sviluppo di sistemi multiagente, in cui ogni agente ha obiettivi, percezioni e ragionamenti locali, ma opera anche all'interno di un ambiente più ampio e interconnesso.

Questa visione dell'intelligenza distribuita, in cui il processo decisionale è decentralizzato e il comportamento emergente deriva dall'interazione tra agenti, rimane fondamentale per il modo in cui i moderni sistemi basati su agenti vengono concepiti e costruiti.

La tipologia di Nwana e l'ascesa degli agenti software

La formalizzazione del concetto di agente software a metà degli anni '90 ha segnato una svolta nell'evoluzione dei sistemi intelligenti. Tra i contributi più influenti a questa formalizzazione c'è il paper fondamentale di Hyacinth S. Nwana, Software Agents: An Overview (Nwana 1996), che ha fornito uno dei primi framework completi per la categorizzazione e la comprensione degli agenti software in varie dimensioni.

In questo paper, Nwana analizza lo stato della ricerca sugli agenti software e identifica una crescente divergenza nel modo in cui gli agenti venivano definiti e implementati. Il paper evidenzia la necessità di un quadro concettuale comune e propone una tipologia che classifica gli agenti in base alle loro capacità chiave. Esamina i sistemi di agenti rappresentativi del mondo accademico e industriale, distingue gli agenti dai programmi e dagli oggetti tradizionali e delinea le sfide e le opportunità dell'informatica basata su agenti.

Nwana sottolinea che gli agenti software non sono un concetto monolitico, ma esistono secondo uno spettro di sofisticatezza e funzionalità. La tipologia serve a chiarire questo panorama e guidare la progettazione e la ricerca future.

Nwana definisce un agente software come un'entità software che funziona in modo continuo e autonomo in un particolare ambiente, che è spesso abitato da altri agenti e processi. Questa definizione enfatizza due caratteristiche chiave:

- Continuità: l'agente opera in modo persistente nel tempo, senza richiedere un intervento umano costante.
- Autonomia: l'agente ha la capacità di prendere decisioni e agire di conseguenza in modo indipendente, in base alla sua percezione dell'ambiente.

Questa definizione, combinata con la tipologia di agente di Nwana, enfatizza l'autorità delegata (attraverso l'autonomia) e la proattività come caratteristiche fondamentali degli agenti. Distingue tra agenti e subroutine o servizi evidenziando la capacità dell'agente di agire in modo indipendente per conto di un'altra entità e di avviare comportamenti finalizzati al perseguimento di obiettivi, invece di rispondere solo a comandi diretti.

La tipologia di agente di Nwana

Per differenziare ulteriormente i vari tipi di agenti, Nwana introduce un sistema di classificazione basato su sei attributi chiave:

- Autonomia: l'agente opera senza l'intervento diretto dell'uomo o di altri.
- Capacità sociale: l'agente interagisce con altri agenti o umani utilizzando meccanismi di comunicazione.
- Reattività: l'agente percepisce il suo ambiente e risponde tempestivamente.
- Proattività: l'agente mostra un comportamento mirato prendendo l'iniziativa.
- Adattabilità e apprendimento: l'agente migliora le proprie prestazioni nel tempo attraverso l'esperienza.
- Mobilità: l'agente può spostarsi tra diversi ambienti di sistema o reti.

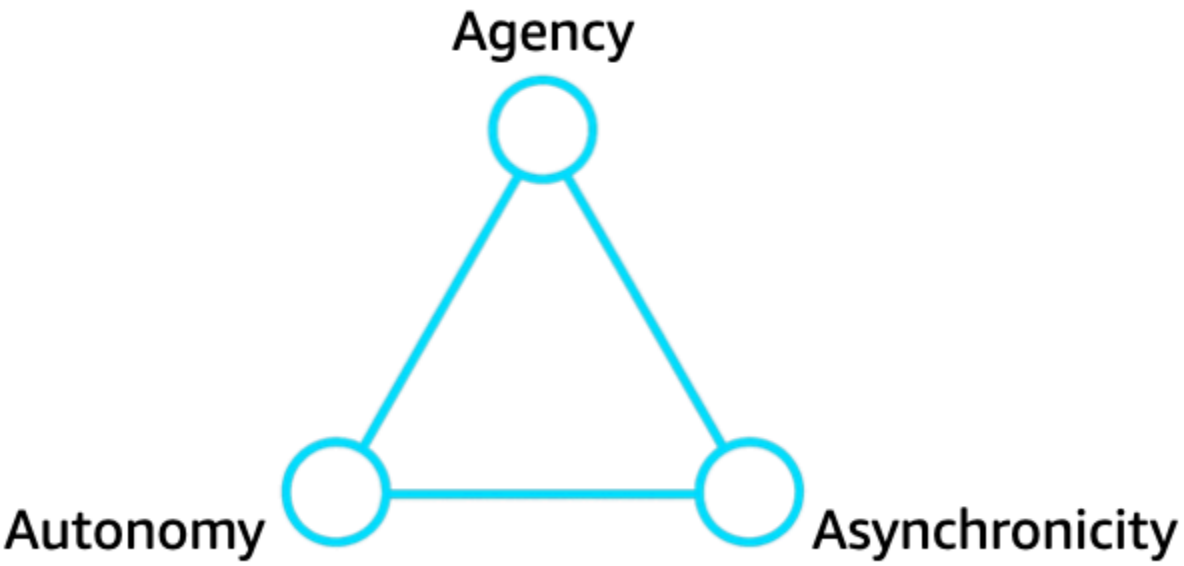
Dalla tipologia ai principi agentici moderni

Il lavoro di Nwana fungeva sia da tassonomia che da lente fondamentale attraverso la quale la comunità informatica poteva valutare le forme in evoluzione di agenzia nel software. La sua enfasi sull'autonomia, la proattività e il concetto di agire per conto di un utente o di un sistema hanno gettato le basi per quello che oggi consideriamo comportamento agentico.

Sebbene le tecnologie e gli ambienti siano cambiati, in particolare con l'avvento dell'intelligenza artificiale generativa, dell'infrastruttura serverless e dei framework di orchestrazione multiagente, le informazioni fondamentali del lavoro di Nwana rimangono pertinenti. Forniscono un ponte fondamentale tra la prima teoria degli agenti e i tre pilastri moderni degli agenti software.

I tre pilastri dei moderni agenti software

Nel contesto delle odierne piattaforme basate sull'intelligenza artificiale, delle architetture di microservizi e dei sistemi basati sugli eventi, gli agenti software possono essere definiti da tre principi interdipendenti che li distinguono dai servizi standard o dagli script di automazione: autonomia, asincronicità e agenzia. Nella figura seguente e nei diagrammi successivi, il triangolo rappresenta questi tre pilastri dei moderni agenti software.



Autonomia

Gli agenti moderni operano in modo indipendente. Prendono decisioni in base allo stato interno e al contesto ambientale senza richiedere suggerimenti umani. Ciò consente loro di reagire ai dati in tempo reale, gestire il proprio ciclo di vita e adattare il proprio comportamento in base agli obiettivi e agli input situazionali.

L'autonomia è alla base del comportamento degli agenti. Consente agli agenti di funzionare senza supervisione continua o flussi di controllo codificati.

Asincronicità

Gli agenti sono fondamentalmente asincroni. Ciò significa che rispondono a eventi, segnali e stimoli man mano che si verificano, senza fare affidamento sul blocco delle chiamate o sui flussi di lavoro lineari. Questa caratteristica consente comunicazioni scalabili e non bloccanti, reattività in ambienti distribuiti e accoppiamento libero tra i componenti.

Grazie all'asincronicità, gli agenti possono partecipare a sistemi in tempo reale e coordinarsi con altri servizi o agenti in modo fluido ed efficiente.

L'agenzia come principio determinante

L'autonomia e l'asincronicità sono necessarie, ma queste caratteristiche da sole non sono sufficienti a rendere un sistema un vero agente software. Il fattore di differenziazione fondamentale è l'agenzia, che introduce:

- Comportamento mirato: gli agenti perseguono obiettivi e valutano i progressi verso il loro raggiungimento.
- Processo decisionale: gli agenti valutano le opzioni e scelgono le azioni in base a regole, modelli o politiche apprese.
- Intento delegato: gli agenti agiscono per conto di una persona, di un sistema o di un'organizzazione e hanno un senso intrinseco dello scopo.
- Ragionamento contestuale: gli agenti incorporano la memoria o i modelli del loro ambiente per guidare il comportamento in modo intelligente.

Un sistema autonomo e asincrono potrebbe comunque essere un servizio reattivo. Ciò che lo rende un agente software è la sua capacità di agire con intenzione e scopo, di essere agentico.

Agenzia con uno scopo

I principi di autonomia, asincronicità e agenzia consentono ai sistemi di operare in modo intelligente, adattivo e indipendente in ambienti distribuiti. Questi principi affondano le radici in decenni di evoluzione concettuale e architettonica e ora sono alla base di molti dei sistemi di intelligenza artificiale più avanzati costruiti oggi.

In questa nuova era di intelligenza artificiale generativa, orchestrazione orientata agli obiettivi e collaborazione multiagente, è essenziale capire cosa rende un agente software veramente agentico. Riconoscere l'agenzia come caratteristica distintiva ci aiuta a superare l'automazione e ad entrare nel regno dell'intelligenza autonoma con uno scopo.

Lo scopo degli agenti software

Poiché i sistemi moderni sono diventati sempre più complessi, distribuiti e intelligenti, il ruolo degli agenti software ha acquisito importanza in tutti i domini che vanno dalle operazioni autonome alle tecnologie di assistenza all'utente. Ma qual è lo scopo alla base degli agenti software? Perché progettiamo sistemi che vanno oltre gli script, i servizi o i modelli statici e deleghiamo invece le attività a entità in grado di percepire, ragionare e agire?

Questa sezione esplora lo scopo fondamentale degli agenti software: consentire la delega intelligente delle attività all'interno di ambienti dinamici, con particolare attenzione all'autonomia, all'adattabilità e all'azione mirata. Introduce le basi concettuali degli agenti software, traccia la loro struttura cognitiva e delinea i problemi del mondo reale che sono in grado di risolvere in modo unico.

In questa sezione

Dal modello dell'attore alla cognizione dell'agente

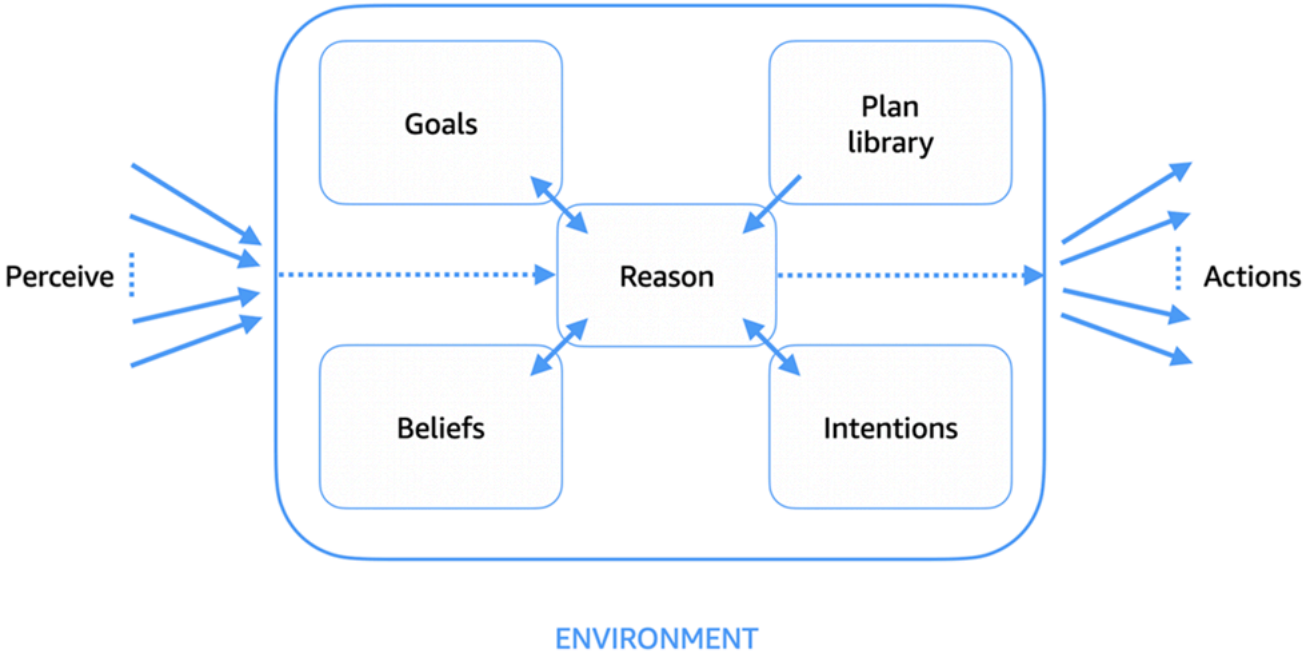
Lo scopo e la struttura degli agenti software si basano su idee emerse dai primi modelli di calcolo, in particolare sul modello dell'attore introdotto da Carl Hewitt negli anni '70 (Hewitt et al. 1973).

Il modello attoriale considera la computazione come un insieme di entità indipendenti, in esecuzione simultanea, chiamate attori. Ogni attore incapsula il proprio stato, interagisce esclusivamente attraverso lo scambio asincrono di messaggi e può creare nuovi attori e delegare attività.

Questo modello ha fornito le basi concettuali per il ragionamento, la reattività e l'isolamento decentralizzati, tutti elementi alla base dell'architettura comportamentale dei moderni agenti software.

La funzione dell'agente: percepire, ragionare, agire

Alla base di ogni agente software c'è un ciclo cognitivo che viene spesso descritto come il ciclo di percezione, ragione, azione. Questo processo è illustrato nello schema seguente. Definisce il modo in cui gli agenti operano autonomamente in ambienti dinamici.



- Percezione: gli agenti raccolgono informazioni (ad esempio eventi, input di sensori o segnali API) dall'ambiente e aggiornano il proprio stato o le proprie convinzioni interne.
- Motivo: gli agenti analizzano le convinzioni, gli obiettivi e le conoscenze contestuali attuali utilizzando una libreria di piani o un sistema logico. Questo processo potrebbe comportare la prioritizzazione degli obiettivi, la risoluzione dei conflitti o la selezione delle intenzioni.
- Azione: gli agenti selezionano ed eseguono azioni che li avvicinano al raggiungimento degli obiettivi delegati.

Questa architettura supporta la capacità degli agenti di funzionare oltre la rigida programmazione e consente un comportamento flessibile, sensibile al contesto e mirato. Forma la struttura mentale che guida gli scopi più ampi degli agenti software.

Collaborazione e intenzionalità autonome

Lo scopo degli agenti software è portare autonomia, consapevolezza del contesto e delega intelligente all'informatica moderna. Poiché gli agenti si basano sui principi del modello dell'attore e sono incorporati nel ciclo percepire, ragionare e agire, consentono sistemi non solo reattivi, ma anche proattivi e mirati.

Gli agenti consentono al software di decidere, adattarsi e agire in ambienti complessi. Rappresentano gli utenti, interpretano gli obiettivi e implementano le attività alla velocità delle macchine. Man mano

che ci addentriamo nell'era dell'intelligenza artificiale agentica, gli agenti software stanno diventando l'interfaccia operativa tra l'intento umano e l'azione digitale intelligente.

Delegare l'intento

A differenza dei componenti software tradizionali, gli agenti software esistono per agire per conto di qualcos'altro: un utente, un altro sistema o un servizio di livello superiore. Hanno un intento delegato, il che significa che:

- Operano in modo indipendente dopo l'iniziazione.
- Fai scelte in linea con gli obiettivi del delegante.
- Affronta l'incertezza e i compromessi nell'esecuzione.

Gli agenti colmano il divario tra istruzioni e risultati, il che consente agli utenti di esprimere l'intenzione con un livello di astrazione più elevato anziché richiedere istruzioni esplicite.

Operare in ambienti dinamici e imprevedibili

Gli agenti software sono progettati per ambienti in cui le condizioni cambiano costantemente, i dati arrivano in tempo reale e il controllo e il contesto sono distribuiti.

A differenza dei programmi statici che richiedono input esatti o un'esecuzione sincrona, gli agenti si adattano all'ambiente circostante e rispondono in modo dinamico. Si tratta di una funzionalità fondamentale nell'infrastruttura nativa del cloud, nell'edge computing, nelle reti Internet of Things (IoT) e nei sistemi decisionali in tempo reale.

Ridurre il carico cognitivo umano

Uno degli scopi principali degli agenti software è ridurre il carico cognitivo e operativo sugli esseri umani. Gli agenti possono:

- Monitorare continuamente sistemi e flussi di lavoro.
- Rileva e rispondi a condizioni predefinite o emergenti.
- Automatizza le decisioni ripetitive e ad alto volume.
- Reagisci ai cambiamenti ambientali con una latenza minima.

Quando il processo decisionale passa dagli utenti agli agenti, i sistemi diventano più reattivi, resilienti e incentrati sull'uomo e possono adattarsi in tempo reale a nuove informazioni o interruzioni. Ciò

consente tempi di reazione più rapidi e una maggiore continuità operativa in ambienti ad alta complessità o su larga scala. Il risultato è uno spostamento dell'attenzione umana, dal processo decisionale a livello microscopico alla supervisione strategica e alla risoluzione creativa dei problemi.

Abilitare l'intelligenza distribuita

La capacità degli agenti software di operare individualmente o collettivamente consente la progettazione di sistemi multiagente (MAS) che si coordinano tra ambienti o organizzazioni. Questi sistemi possono distribuire le attività in modo intelligente e negoziare, cooperare o competere verso obiettivi compositi.

Ad esempio, in un sistema di catena di approvvigionamento globale, i singoli agenti gestiscono le fabbriche, le spedizioni, i magazzini e le consegne dell'ultimo miglio. Ogni agente opera con autonomia locale: gli agenti di fabbrica ottimizzano la produzione in base ai vincoli di risorse, gli agenti di magazzino regolano i flussi di inventario in tempo reale e gli agenti di consegna reindirizzano le spedizioni in base al traffico e alla disponibilità dei clienti.

Questi agenti comunicano e si coordinano dinamicamente e si adattano a interruzioni come ritardi nei porti o guasti dei camion senza un controllo centralizzato. L'intelligenza complessiva del sistema emerge da queste interazioni e consente una logistica resiliente e ottimizzata che va oltre le capacità di un singolo componente.

In questo modello, gli agenti agiscono come nodi in un tessuto di intelligence più ampio. Formano sistemi emergenti in grado di risolvere problemi che nessun singolo componente potrebbe gestire da solo.

Agire con uno scopo, non solo con una reazione

L'automazione da sola non è sufficiente nei sistemi complessi. Lo scopo principale di un agente software è agire con uno scopo e valutare gli obiettivi, valutare il contesto e fare scelte informate. Ciò significa che gli agenti software perseguono obiettivi anziché rispondere solo ai fattori scatenanti. Possono rivedere le convinzioni e le intenzioni sulla base dell'esperienza o del feedback. In questo contesto, le convinzioni si riferiscono alla rappresentazione interna dell'ambiente da parte dell'agente (ad esempio, «il pacchetto X è nel magazzino A»), in base alle sue percezioni (input e sensori). Le intenzioni si riferiscono ai piani scelti dall'agente per raggiungere un obiettivo (ad esempio, «utilizzare il percorso di consegna B e informare il destinatario»). Gli agenti possono anche intensificare, rinviare o adattare le azioni necessarie.

Questa intenzionalità è ciò che rende gli agenti software non solo esecutori reattivi, ma collaboratori autonomi in sistemi intelligenti.

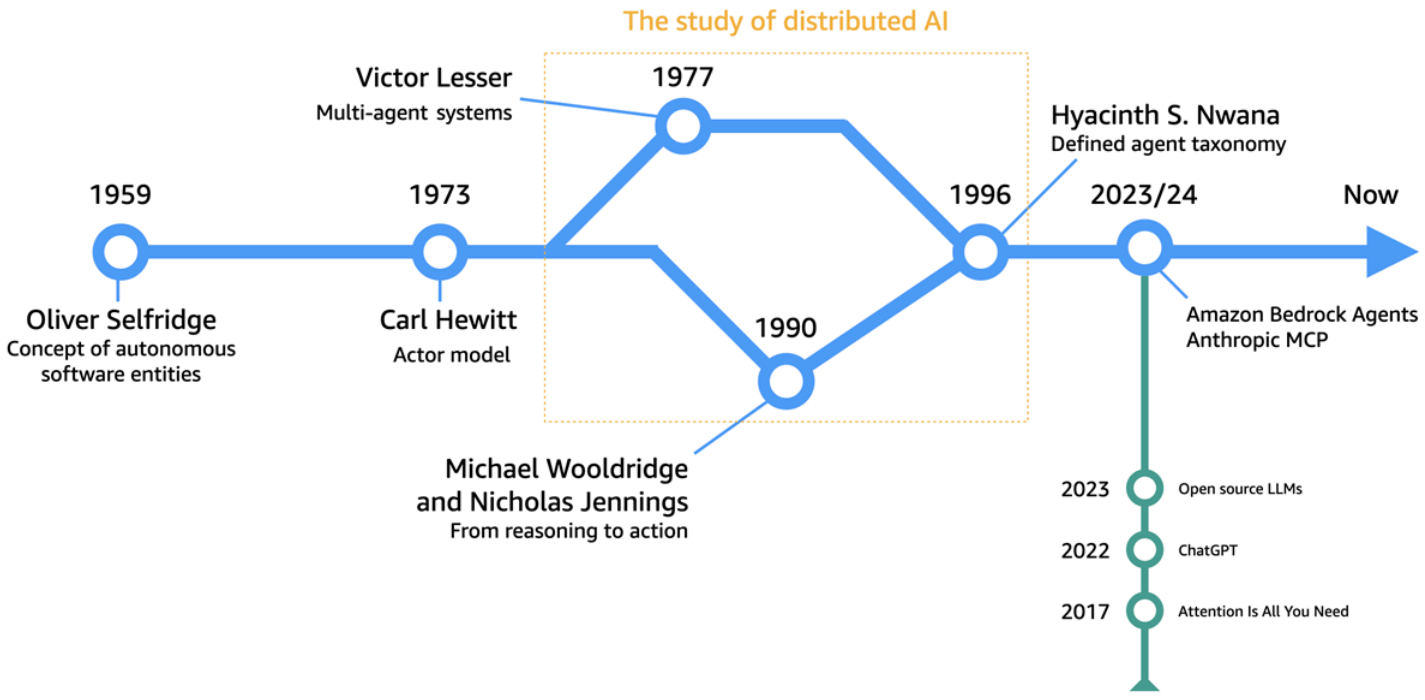
L'evoluzione degli agenti software

Il passaggio da semplici sistemi automatizzati ad agenti software intelligenti, autonomi e orientati agli obiettivi riflette decenni di evoluzione nell'informatica, nell'intelligenza artificiale e nei sistemi distribuiti.

Questa evoluzione è stata seguita dall'avvento dell'apprendimento automatico, che ha spostato il paradigma dalle regole artigianali al riconoscimento di schemi statistici. Questi sistemi potevano imparare dai dati e consentire progressi nella percezione, nella classificazione e nel processo decisionale.

I modelli linguistici di grandi dimensioni (LLMs) rappresentano una convergenza di scala, architettura e apprendimento senza supervisione. LLMs può ragionare, generare e adattare le attività con una formazione specifica minima o nulla. Grazie alla combinazione LLMs con un'infrastruttura scalabile nativa per il cloud e le architetture componibili, stiamo ora realizzando la visione completa dell'intelligenza artificiale agentica: agenti software intelligenti in grado di operare con autonomia, consapevolezza del contesto e adattabilità su scala aziendale.

Questa sezione esplora la storia degli agenti software dalla teoria di base alla pratica moderna, come illustrato nel diagramma seguente. Evidenzia la convergenza tra l'intelligenza artificiale distribuita (DAI) e l'IA generativa basata su trasformatori e identifica le tappe fondamentali che hanno plasmato l'emergere dell'intelligenza artificiale agentica.



In questa sezione

Fondamenti degli agenti software

1959 — Oliver Selfridge: la nascita dell'autonomia nel software

Le radici degli agenti software risalgono a Oliver Selfridge, che introdusse il concetto di entità software autonome (demoni), programmi in grado di percepire il loro ambiente e di agire in modo indipendente (Selfridge 1959). I suoi primi lavori sulla percezione e l'apprendimento delle macchine hanno gettato le basi filosofiche per le future nozioni di agenti come sistemi indipendenti e intelligenti.

1973 — Carl Hewitt: l'attore modello

Un progresso fondamentale è avvenuto con il modello di attore di Carl Hewitt (Hewitt et al. 1973), che è un modello computazionale formale che descrive gli agenti come entità indipendenti e concorrenti. In questo modello, gli agenti possono incapsulare il proprio stato e il proprio comportamento, comunicare utilizzando lo scambio asincrono di messaggi e creare dinamicamente altri attori e delegare loro compiti.

Il modello ad attori ha fornito sia la base teorica che il paradigma architetturale per sistemi distribuiti basati su agenti. Questo modello prefigurava le moderne implementazioni di concorrenza come il linguaggio di programmazione Erlang e il framework Akka.

Maturare il campo: dal ragionamento all'azione

1977 — Victor Lesser: sistemi multiagente

Alla fine degli anni '70, è emersa l'intelligenza artificiale distribuita (DAI). È stata sostenuta da Victor Lesser, che è ampiamente riconosciuto per i sistemi pionieristici multiagente (MAS). Il suo lavoro si è concentrato sul modo in cui le entità software indipendenti potevano cooperare, coordinarsi e negoziare (vedere la sezione Risorse). Questo sviluppo ha portato a sistemi in grado di risolvere collettivamente problemi complessi: un passo avanti essenziale nella creazione di intelligenza distribuita.

Anni '90 — Michael Wooldridge e Nicholas Jennings: lo spettro degli agenti

Negli anni '90, il campo dell'intelligenza distribuita era maturato grazie ai contributi di ricercatori come Michael Wooldridge e Nicholas Jennings. Questi studiosi hanno classificato gli agenti lungo uno spettro, da quelli reattivi a quelli deliberativi, dai sistemi non cognitivi agli agenti ragionanti orientati agli obiettivi (Wooldridge e Jennings 1995). Il loro lavoro ha sottolineato che gli agenti non erano più idee astratte, ma venivano applicati in un'ampia gamma di domini pratici, dalla robotica al software aziendale.

Questi ricercatori hanno anche introdotto uno spostamento dell'attenzione: dal ragionamento centralizzato all'azione distribuita. Gli agenti non erano più solo pensatori, erano agenti che operavano in ambienti in tempo reale con autonomia e determinazione.

1996 — Hyacinth S. Nwana: formalizzazione del concetto di agente

Nel 1996, Hyacinth S. Nwana pubblicò l'influente paper Software Agents: An Overview, che forniva la classificazione degli agenti più completa fino ad oggi. La sua tipologia includeva attributi come autonomia, abilità sociale, reattività, proattività, apprendimento e mobilità e distingueva tra agenti software e costrutti software tradizionali.

Nwana offriva anche una definizione ormai ampiamente accettata, parafrasata: un agente software è un programma per computer basato su software che agisce per un utente o un altro programma in un rapporto di agenzia, che deriva dalla nozione di delega.

Questa formalizzazione è stata fondamentale per la transizione degli agenti software da costrutti teorici ad applicazioni del mondo reale. Ha dato origine a una generazione di sistemi basati su agenti in settori quali le telecomunicazioni, l'automazione del flusso di lavoro e gli assistenti intelligenti.

Il lavoro di Nwana si colloca nel punto di convergenza tra le prime ricerche sull'intelligenza artificiale distribuita e le architetture operative degli agenti moderni. È un ponte cruciale tra la teoria cognitiva degli agenti e la loro implementazione pratica nei sistemi odierni.

Una linea temporale parallela: l'ascesa di modelli linguistici di grandi dimensioni

Mentre i framework degli agenti si evolvevano, stava avvenendo una rivoluzione parallela e convergente nell'elaborazione del linguaggio naturale e nell'apprendimento automatico:

- 2017 — transformers: il paper Attention Is All You Need (Vaswani et al. 2017) ha introdotto l'architettura dei trasformatori, che ha notevolmente migliorato il modo in cui le macchine elaborano e generano il linguaggio.
- 2022 — ChatGPT: OpenAI ha rilasciato un'interfaccia basata su chat per GPT-3.5 chiamata ChatGPT, che ha consentito una conversazione naturale e interattiva con un sistema di intelligenza artificiale generico.
- 2023 — open source LLMs: le versioni di Llama, Falcon e Mistral hanno reso ampiamente accessibili modelli potenti e hanno accelerato lo sviluppo di framework di agenti in ambienti open source e aziendali.

Queste innovazioni hanno trasformato i modelli linguistici in motori di ragionamento in grado di analizzare il contesto, pianificare azioni e concatenare le risposte, e sono diventate fattori chiave per la creazione di agenti software intelligenti. LLMs

Le tempistiche convergono: l'emergere dell'intelligenza artificiale agentica

2023-2024: piattaforme per agenti di livello aziendale

La convergenza tra architetture di agenti software distribuiti e basate su trasformatori è culminata nella nascita dell'intelligenza artificiale agentica. LLMs

- Amazon Bedrock Agents ha introdotto un modo completamente gestito per creare agenti software orientati agli obiettivi e che utilizzano strumenti utilizzando modelli di base di Amazon Bedrock.
- Il Model Context Protocol (MCP) di Anthropic ha definito un metodo per consentire a modelli di linguaggio di grandi dimensioni di accedere e interagire con strumenti, ambienti e memoria esterni. Questo è fondamentale per un comportamento contestuale, persistente e autonomo.

Queste due pietre miliari rappresentano la sintesi di agenzia e intelligenza. Gli agenti non erano più limitati ai flussi di lavoro statici o all'automazione rigida. Ora potevano ragionare in più fasi, coordinarsi con gli strumenti e APIs mantenere lo stato contestuale, nonché apprendere e adattarsi nel tempo.

Gennaio-giugno 2025: funzionalità aziendali ampliate

Nella prima metà del 2025, il panorama dell'intelligenza artificiale agentica si è notevolmente ampliato con nuove funzionalità aziendali. Nel febbraio 2025, Anthropic ha rilasciato Claude 3.7 Sonnet, che è stato il primo modello di ragionamento ibrido sul mercato, e la specifica MCP ha ottenuto un'adozione diffusa.

Assistenti di codifica AI come Amazon Q Developer, Cursor e MCP WindSurf integrato per standardizzare la generazione di codice, l'analisi degli archivi e i flussi di lavoro di sviluppo. La versione MCP di marzo 2025 ha introdotto importanti funzionalità pronte per le aziende, tra cui l'integrazione della sicurezza OAuth 2.1, tipi di risorse estesi per diversi accessi ai dati e opzioni di connettività avanzate tramite Streamable HTTP. Basandosi su queste fondamenta, AWS ha annunciato a maggio 2025 di entrare a far parte del comitato direttivo di MCP e di contribuire a nuove funzionalità di comunicazione. agent-to-agent Ciò rafforza ulteriormente la posizione del protocollo come standard di settore per l'interoperabilità dell'intelligenza artificiale agentica.

A maggio 2025, AWS ha rafforzato le opzioni dei clienti per la creazione di flussi di lavoro di intelligenza artificiale agentici mediante l'open source del framework Strands Agents. Questo framework indipendente dal provider e indipendente dal modello consente agli sviluppatori di utilizzare modelli di base su più piattaforme mantenendo al contempo una profonda integrazione dei servizi. AWS Come evidenziato nel blog AWS Open Source, Strands Agents segue una filosofia di progettazione basata sul modello che pone i modelli di base al centro dell'intelligence degli agenti. Ciò semplifica per i clienti la creazione e l'implementazione di agenti di intelligenza artificiale sofisticati per i loro casi d'uso specifici.

Emergenza: intelligenza artificiale agentica

L'evoluzione degli agenti software, dalle prime idee di autonomia all'orchestrazione moderna basata su LLM, è stata lunga e stratificata. Ciò che è iniziato con la visione di Oliver Selfridge di percepire i programmi è diventato un solido ecosistema di agenti software intelligenti, sensibili al contesto e orientati agli obiettivi, in grado di collaborare, adattarsi e ragionare.

La convergenza tra intelligenza artificiale distribuita (DAI) e intelligenza artificiale generativa basata su trasformatori segna l'inizio di una nuova era in cui gli agenti software non sono più solo strumenti, ma attori autonomi in sistemi intelligenti.

L'intelligenza artificiale agentica rappresenta la prossima evoluzione dei sistemi software. Fornisce una classe di agenti intelligenti autonomi, asincroni e agentici, in grado di agire con intenti delegati e operare in modo mirato in ambienti dinamici e distribuiti. Agentic AI unifica quanto segue:

- La discendenza architettonica dei sistemi multiagente e il modello dell'attore
- Il modello cognitivo del percepire, ragionare, agire
- La potenza generativa dei trasformatori e LLMs
- La flessibilità operativa dell'elaborazione nativa per il cloud e senza server

Agenti software per l'intelligenza artificiale agentica

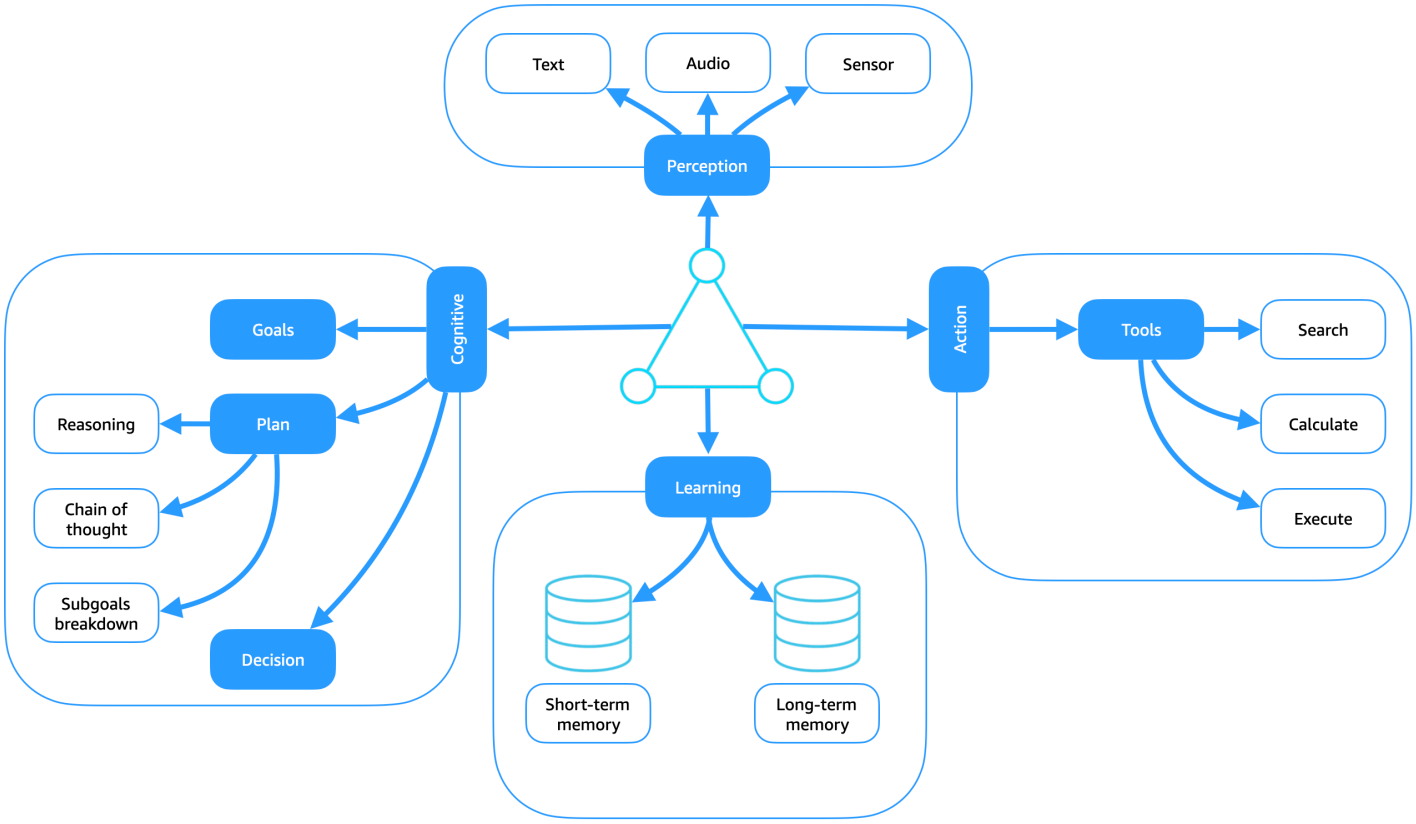
Gli agenti software sono entità digitali autonome progettate per percepire il loro ambiente, ragionare sui propri obiettivi e agire di conseguenza. A differenza dei programmi software tradizionali che seguono una logica fissa, gli agenti adattano il loro comportamento in base a input contestuali e quadri decisionali. Ciò li rende ideali per ambienti dinamici e distribuiti come i sistemi nativi del cloud, la robotica, l'automazione intelligente e, ora, l'orchestrazione generativa dell'intelligenza artificiale.

Questa sezione introduce gli elementi costitutivi principali degli agenti software e spiega come questi componenti interagiscono all'interno delle architetture tradizionali basate sul modello percepire, ragionare, agire. Descrive come l'intelligenza artificiale generativa, in particolare i modelli linguistici di grandi dimensioni (LLMs), abbia trasformato il modo in cui gli agenti software ragionano e pianificano. Ciò segna un passaggio fondamentale dai sistemi basati su regole all'intelligenza acquisita e basata sui dati dell'IA agentica.

In questa sezione

Elementi costitutivi fondamentali degli agenti software

Il diagramma seguente presenta i moduli funzionali chiave presenti nella maggior parte degli agenti intelligenti. Ogni componente contribuisce alla capacità dell'agente di operare autonomamente in ambienti complessi.



Nel contesto del ciclo percepire, ragionare, agire, la capacità di ragionamento di un agente è distribuita tra i suoi moduli cognitivi e di apprendimento. Attraverso l'integrazione della memoria e dell'apprendimento, l'agente sviluppa un ragionamento adattivo basato sull'esperienza passata. Quando l'agente agisce all'interno del suo ambiente, crea un ciclo di feedback emergente: ogni azione influenza le percezioni future e l'esperienza risultante viene incorporata nella memoria e nei modelli interni attraverso il modulo di apprendimento. Questo ciclo continuo di percezione, ragionamento e azione consente all'agente di migliorare nel tempo e completa l'intero ciclo di percezione, ragione e azione.

Modulo di percezione

Il modulo di percezione consente all'agente di interfacciarsi con il proprio ambiente attraverso diverse modalità di input come testo, audio e sensori. Questi input costituiscono i dati grezzi su cui si basano tutti i ragionamenti e le azioni. Gli input di testo possono includere istruzioni in linguaggio naturale, comandi strutturati o documenti. Gli ingressi audio comprendono istruzioni vocali o suoni ambientali. Gli ingressi dei sensori includono dati fisici come feed visivi, segnali di movimento o coordinate GPS. La funzione principale della percezione è estrarre caratteristiche e rappresentazioni significative da questi dati grezzi. Ciò consente all'agente di costruire una comprensione accurata e pratica del contesto attuale. Il processo potrebbe comportare l'estrazione di caratteristiche, il riconoscimento

di oggetti o eventi e l'interpretazione semantica e costituisce la prima fase fondamentale del ciclo percepire, ragionare, agire. Una percezione efficace garantisce che il ragionamento e il processo decisionale a valle siano fondati su una consapevolezza situazionale pertinente. up-to-date

Modulo cognitivo

Il modulo cognitivo funge da nucleo deliberativo dell'agente software. È responsabile dell'interpretazione delle percezioni, della formazione degli intenti e della guida di comportamenti intenzionali attraverso una pianificazione e un processo decisionale orientati agli obiettivi. Questo modulo trasforma gli input in processi di ragionamento strutturati, che consentono all'agente di operare intenzionalmente anziché in modo reattivo. Questi processi sono gestiti attraverso tre sottomoduli chiave: obiettivi, pianificazione e processo decisionale.

Sottomodulo degli obiettivi

Il sottomodulo degli obiettivi definisce l'intento e la direzione dell'agente. Gli obiettivi possono essere espliciti (ad esempio, «raggiungere una località» o «inviare un rapporto») o impliciti (ad esempio, «massimizzare il coinvolgimento degli utenti» o «ridurre al minimo la latenza»). Sono fondamentali per il ciclo di ragionamento dell'agente e forniscono uno stato obiettivo per la sua pianificazione e le sue decisioni.

L'agente valuta continuamente i progressi verso i propri obiettivi e potrebbe ridefinire le priorità o rigenerare gli obiettivi sulla base di nuove percezioni o apprendimenti. Questa consapevolezza degli obiettivi mantiene l'agente adattabile in ambienti dinamici.

Sottomodulo di pianificazione

Il sottomodulo di pianificazione costruisce strategie per raggiungere gli obiettivi attuali dell'agente. Genera sequenze di azioni, scompone le attività gerarchicamente e seleziona tra piani predefiniti o generati dinamicamente.

Per operare efficacemente in ambienti non deterministici o mutevoli, la pianificazione non è statica. Gli agenti moderni possono generare chain-of-thought sequenze, introdurre obiettivi secondari come fasi intermedie e rivedere i piani in tempo reale quando le condizioni cambiano.

Questo sottomodulo si collega strettamente alla memoria e all'apprendimento e consente all'agente di affinare la propria pianificazione nel tempo sulla base dei risultati passati.

Sottomodulo decisionale

Il sottomodulo decisionale valuta i piani e le azioni disponibili per selezionare la fase successiva più appropriata. Integra gli input provenienti dalla percezione, dal piano attuale, dagli obiettivi dell'agente e dal contesto ambientale.

Conti decisionali per:

- Compromessi tra obiettivi contrastanti
- Soglie di confidenza (ad esempio, incertezza nella percezione)
- Conseguenze delle azioni
- L'esperienza acquisita dall'agente

A seconda dell'architettura, gli agenti potrebbero fare affidamento sul ragionamento simbolico, sull'euristica, sull'apprendimento per rinforzo o sui modelli linguistici (LLMs) per prendere decisioni informate. Questo processo garantisce che il comportamento dell'agente rimanga consapevole del contesto, allineato agli obiettivi e adattivo.

Modulo d'azione

Il modulo d'azione è responsabile dell'esecuzione delle decisioni selezionate dall'agente e dell'interfaccia con il mondo esterno o i sistemi interni per produrre effetti significativi. Rappresenta la fase dell'atto del ciclo percepire, ragionare, agire, in cui l'intento si trasforma in comportamento.

Quando il modulo cognitivo seleziona un'azione, il modulo d'azione coordina l'esecuzione tramite sottomoduli specializzati, in cui ogni sottomodulo si allinea con l'ambiente integrato dell'agente:

- Attuazione fisica: per gli agenti integrati in sistemi robotici o dispositivi IoT, questo sottomodulo traduce le decisioni in movimenti fisici reali o istruzioni a livello di hardware.

  Esempi: pilotare un robot, attivare una valvola, accendere un sensore.
- Interazione integrata: questo sottomodulo gestisce azioni non fisiche ma visibili esternamente, come l'interazione con sistemi software, piattaforme o. APIs

  Esempi: invio di un comando a un servizio cloud, aggiornamento di un database, invio di un rapporto chiamando un'API.
- Richiamo di strumenti: gli agenti spesso estendono le proprie funzionalità utilizzando strumenti specializzati per eseguire attività secondarie come le seguenti:

- Ricerca: interrogazione di fonti di conoscenza strutturate o non strutturate
- Riepilogo: compressione di input di testo di grandi dimensioni in panoramiche di alto livello
- Calcolo: esecuzione di calcoli logici, numerici o simbolici

L'invocazione di strumenti consente una composizione comportamentale complessa attraverso competenze modulari e richiamabili.

Modulo di apprendimento

Il modulo di apprendimento consente agli agenti di adattarsi, generalizzare e migliorare nel tempo in base all'esperienza. Supporta il processo di ragionamento perfezionando continuamente i modelli interni, le strategie e le politiche decisionali dell'agente utilizzando il feedback derivante dalla percezione e dall'azione.

Questo modulo opera in coordinamento con la memoria a breve e lungo termine:

- Memoria a breve termine: memorizza il contesto transitorio, come lo stato del dialogo, le informazioni sulle attività correnti e le osservazioni recenti. Aiuta l'agente a mantenere la continuità all'interno delle interazioni e delle attività.
- Memoria a lungo termine: codifica la conoscenza persistente delle esperienze passate, inclusi gli obiettivi raggiunti in precedenza, i risultati delle azioni e gli stati ambientali. La memoria a lungo termine consente all'agente di riconoscere schemi, riutilizzare le strategie ed evitare di ripetere gli errori.

Modalità di apprendimento

Il modulo di apprendimento supporta una serie di paradigmi, come l'apprendimento supervisionato, non supervisionato e per rinforzo, che supportano diversi ambienti e ruoli degli agenti:

- Apprendimento supervisionato: aggiorna i modelli interni sulla base di esempi etichettati, spesso tratti da feedback umani o set di dati di formazione.

  Esempio: imparare a classificare le intenzioni degli utenti in base alle conversazioni precedenti.
- Apprendimento senza supervisione: identifica modelli o strutture nascoste nei dati senza etichette esplicite.

  Esempio: raggruppamento di segnali ambientali per rilevare anomalie.

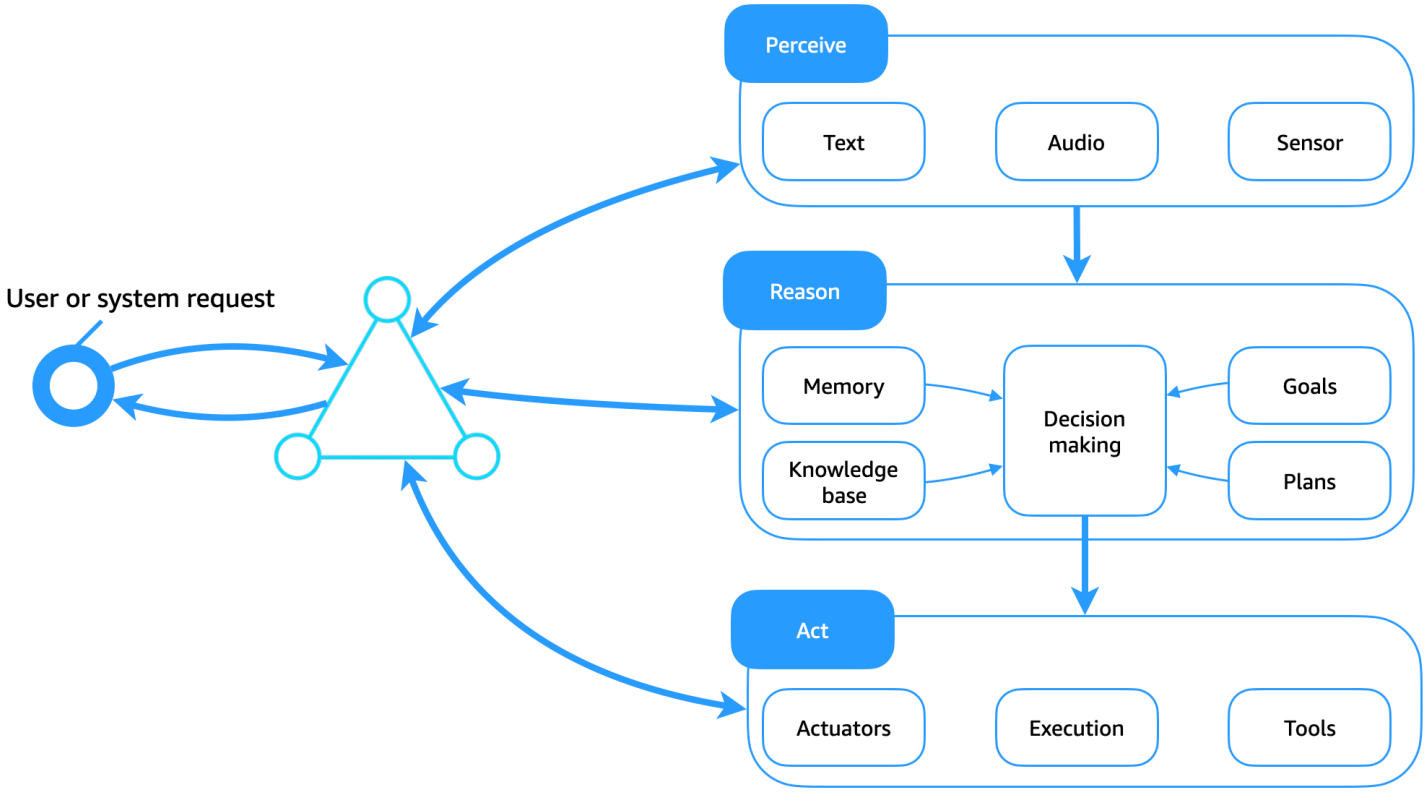
- Apprendimento per rinforzo: ottimizza il comportamento attraverso tentativi ed errori massimizzando la ricompensa cumulativa in ambienti interattivi.

  Esempio: imparare quale strategia porta al completamento più rapido delle attività.

L'apprendimento si integra perfettamente con il modulo cognitivo dell'agente. Perfeziona le strategie di pianificazione basate sui risultati passati, migliora il processo decisionale attraverso la valutazione del successo storico e migliora continuamente la mappatura tra percezione e azione. Attraverso questo ciclo chiuso di apprendimento e feedback, gli agenti si evolvono oltre l'esecuzione reattiva per diventare sistemi che si autovalutano e sono in grado di adattarsi a nuovi obiettivi, condizioni e contesti nel tempo.

Architettura tradizionale degli agenti: percepire, ragionare, agire

Il diagramma seguente illustra come gli elementi costitutivi discussi nella sezione precedente operano nell'ambito del ciclo percepire, ragionare, agire.



Modulo Perceive

Il modulo di percezione funge da interfaccia sensoriale dell'agente con il mondo esterno. Trasforma gli input ambientali grezzi in rappresentazioni strutturate che informano il ragionamento. Ciò include la gestione di dati multimodali come testo, audio o segnali dei sensori.

- L'immissione di testo può provenire da comandi utente, documenti o dialoghi.
- L'ingresso audio include istruzioni vocali o suoni ambientali.
- L'input del sensore acquisisce segnali del mondo reale come movimento, feed visivi o GPS.

Una volta acquisito l'input grezzo, il processo di percezione esegue l'estrazione delle caratteristiche, seguita dal riconoscimento di oggetti o eventi e dall'interpretazione semantica per creare un modello significativo della situazione attuale. Questi risultati forniscono un contesto strutturato per il processo decisionale a valle e ancorano il ragionamento dell'agente alle osservazioni del mondo reale.

Modulo Reason

Il modulo reason è il nucleo cognitivo dell'agente. Valuta il contesto, formula l'intento e determina le azioni appropriate. Questo modulo orchestra il comportamento basato sugli obiettivi utilizzando sia le conoscenze apprese che il ragionamento.

Il modulo reason è costituito da sottomoduli strettamente integrati:

- Memoria: mantiene lo stato del dialogo, il contesto dell'attività e la cronologia episodica in formati sia a breve che a lungo termine.
- Knowledge base: fornisce l'accesso a regole simboliche, ontologie o modelli appresi (come incorporamenti, fatti e politiche).
- Obiettivi e piani: definisce i risultati desiderati e costruisce strategie d'azione per raggiungerli. Gli obiettivi possono essere aggiornati dinamicamente e i piani possono essere modificati in modo adattivo in base al feedback.
- Processo decisionale: funge da motore arbitrale centrale soppesando le opzioni, valutando i compromessi e selezionando l'azione successiva. Questo sottomodulo tiene conto delle soglie di confidenza, dell'allineamento degli obiettivi e dei vincoli contestuali.

Insieme, questi componenti consentono all'agente di ragionare sul proprio ambiente, aggiornare le convinzioni, selezionare percorsi e comportarsi in modo coerente e adattivo. Il modulo reason colma il divario tra percezione e comportamento.

Modulo Act

Il modulo act esegue la decisione selezionata dall'agente interfacciandosi con l'ambiente digitale o fisico per eseguire le attività. È qui che l'intenzione diventa azione.

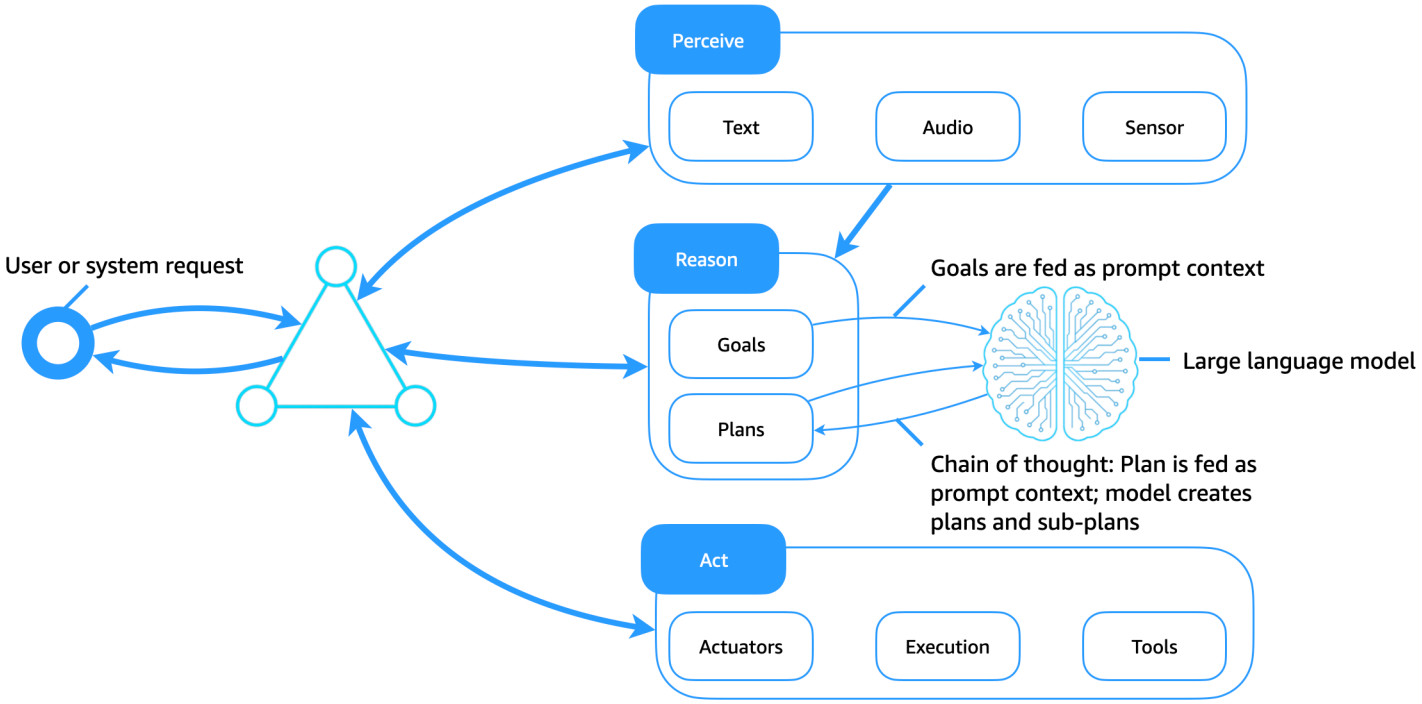
Questo modulo include tre canali funzionali:

- Attuatori: per gli agenti che hanno una presenza fisica (come robot e dispositivi IoT), controlla le interazioni a livello di hardware come movimento, manipolazione o segnalazione.
- Esecuzione: gestisce le azioni basate sul software, tra cui l'invocazione, l'invio di comandi e l'aggiornamento dei sistemi. APIs
- Strumenti: abilita funzionalità funzionali come ricerca, riepilogo, esecuzione del codice, calcolo e gestione dei documenti. Questi strumenti sono spesso dinamici e sensibili al contesto, il che estende l'utilità dell'agente.

Le uscite del modulo act reimmettono nell'ambiente e chiudono il ciclo. Questi risultati vengono nuovamente percepiti dall'agente. Aggiornano lo stato interno dell'agente e informano le decisioni future, completando così il ciclo di percezione, ragione, azione.

Agenti di intelligenza artificiale generativa: sostituiscono la logica simbolica con LLMs

Il diagramma seguente illustra come i modelli linguistici di grandi dimensioni (LLMs) fungano ora da nucleo cognitivo flessibile e intelligente per gli agenti software. A differenza dei tradizionali sistemi logici simbolici, che si basano su librerie di piani statiche e regole codificate manualmente, LLMs consentono il ragionamento adattivo, la pianificazione contestuale e l'uso dinamico degli strumenti, che trasformano il modo in cui gli agenti percepiscono, ragionano e agiscono.



Miglioramenti chiave

Questa architettura migliora l'architettura tradizionale degli agenti come segue:

- LLMs come motori cognitivi: gli obiettivi, i piani e le domande vengono passati al modello come contesto immediato. L'LLM genera percorsi di ragionamento (come la catena di pensiero), scompone le attività in obiettivi secondari e decide le azioni successive.
- Utilizzo dello strumento tramite richiesta: LLMs può essere indirizzato tramite agenti di utilizzo dello strumento o ragionamento e azione (ReAct) che richiedono di chiamare e cercare, interrogare, calcolare APIs e interpretare gli output.
- Pianificazione sensibile al contesto: gli agenti generano o rivedono i piani in modo dinamico in base all'obiettivo attuale, all'ambiente di input e al feedback dell'agente, senza richiedere librerie di piani codificate.
- Richiedi il contesto come memoria: invece di utilizzare basi di conoscenza simboliche, gli agenti codificano la memoria, i piani e gli obiettivi come token di richiesta che vengono passati al modello.
- Apprendimento mediante un apprendimento contestuale in pochi passaggi: LLMs adattate i comportamenti mediante una progettazione tempestiva, che riduce la necessità di riqualificazione esplicita o di rigide librerie di piani.

Raggiungere una memoria a lungo termine negli agenti basati su LLM

A differenza degli agenti tradizionali, che immagazzinavano la memoria a lungo termine in basi di conoscenza strutturate, gli agenti di intelligenza artificiale generativa devono funzionare entro i limiti della finestra di contesto di. LLMs Per estendere la memoria e supportare l'intelligenza persistente, gli agenti di intelligenza artificiale generativa utilizzano diverse tecniche complementari: agent store, Retrieval-Augmented Generation (RAG), apprendimento contestuale e concatenamento rapido e preformazione.

Agent store: memoria esterna a lungo termine

Lo stato dell'agente, la cronologia degli utenti, le decisioni e i risultati vengono archiviati in un archivio di memoria a lungo termine per agenti (ad esempio un database vettoriale, un archivio di oggetti o un archivio di documenti). Le memorie pertinenti vengono recuperate su richiesta e inserite nel contesto del prompt LLM in fase di esecuzione. Questo crea un ciclo di memoria persistente, in cui l'agente mantiene la continuità tra sessioni, attività o interazioni.

STRACCIO

RAG migliora le prestazioni LLM combinando le conoscenze recuperate con capacità generative. Quando viene emesso un obiettivo o una richiesta, l'agente cerca un indice di recupero (ad esempio, tramite una ricerca semantica di documenti, conversazioni precedenti o conoscenze strutturate). I risultati recuperati vengono aggiunti al prompt LLM, che basa la generazione su fatti esterni o su un contesto personalizzato. Questo metodo estende la memoria effettiva dell'agente e migliora l'affidabilità e la correttezza dei fatti.

Apprendimento contestuale e concatenamento rapido

Gli agenti mantengono la memoria a breve termine utilizzando il contesto dei token all'interno della sessione e il concatenamento strutturato dei prompt. Gli elementi contestuali, come il piano attuale, i risultati delle azioni precedenti e lo stato dell'agente, vengono trasmessi tra le chiamate per guidare il comportamento.

Preformazione e perfezionamento continui

Per gli agenti specifici del dominio, è LLMs possibile continuare la formazione preliminare su raccolte personalizzate come registri, dati aziendali o documentazione di prodotto. In alternativa, la messa a punto delle istruzioni o l'apprendimento per rinforzo basato sul feedback umano (RLHF) possono incorporare un comportamento simile a quello di un agente direttamente nel modello. Ciò sposta

i modelli di ragionamento dalla logica del momento immediato alla rappresentazione interna del modello, riduce la lunghezza dei prompt e migliora l'efficienza.

Vantaggi combinati dell'intelligenza artificiale agentica

Queste tecniche, se utilizzate insieme, consentono agli agenti di intelligenza artificiale generativa di:

- Mantieni la consapevolezza contestuale nel tempo.
- Adatta il comportamento in base alla cronologia o alle preferenze degli utenti.
- Prendi decisioni utilizzando up-to-date conoscenze concrete o private.
- Adattate ai casi d'uso aziendali con comportamenti persistenti, conformi e spiegabili.

LLMs Grazie all'utilizzo della memoria esterna, ai livelli di recupero e alla formazione continua, gli agenti possono raggiungere un livello di continuità cognitiva e uno scopo che non poteva essere raggiunto in precedenza attraverso i soli sistemi simbolici.

Confronto tra l'IA tradizionale e gli agenti software e l'intelligenza artificiale agentica

La tabella seguente fornisce un confronto dettagliato tra intelligenza artificiale tradizionale, agenti software e intelligenza artificiale agentica.

Caratteristica	AI tradizionale	Agenti software	AI agentica
Esempi	Filtri antispam, classificatori di immagini, motori di raccomandazione	Chatbot, pianificatori di attività, agenti di monitoraggio	Assistenti AI, agenti di sviluppo autonomi, orchestrazioni LLM multiagente
Modello di esecuzione	Batch o sincrono	Basato sugli eventi o pianificato	Asincrono, basato sugli eventi e sugli obiettivi
Autonomia	Limitato; spesso richiede un'orches trazione umana o esterna	Medio; funziona in modo indipendente entro limiti predefiniti	Alto; agisce in modo indipendente con strategie adattive

Caratteristica	AI tradizionale	Agenti software	AI agentica
Reattività	Reattivo ai dati di input	Reattivo all'ambiente e agli eventi	Reattivo e proattivo; anticipa e avvia azioni
Proattività	Raro	Presente in alcuni sistemi	Attributo principale; guida il comportam ento mirato agli obiettivi
Communication	Minimo; di solito autonomo o legato all'API	Messaggistica interagente o agente-umano	Interazione e multiagente human-in-the-loop avanzate
Processo decisionale	Solo inferenza del modello (classifi cazione, previsione e così via)	Ragionamento simbolico o decisioni basate su regole o sceneggiate	Ragionamento contestuale, basato sugli obiettivi e dinamico (spesso migliorato con LLM)
Intento delegato	No; esegue attività definite direttamente dall'utente	Parziale; agisce per conto di utenti o sistemi con portata limitata	Sì; agisce con obiettivi delegati, spesso tra servizi, utenti o sistemi
Apprendimento e adattamento	Spesso incentrato sul modello (ad esempio, formazione ML)	A volte adattivo	Apprendimento, memoria o ragioname nto incorporati (ad esempio feedback, autocorrezione)
Agenzia	Nessuna; strumenti per umani	Implicito o di base	Esplicito; opera con scopi, obiettivi e autodirezione

Caratteristica	AI tradizionale	Agenti software	AI agentica
Consapevolezza del contesto	Bassa; stateless o basata su istantanee	Moderato; tracciamento dello stato in parte	Alto; utilizza modelli di memoria, contesto situazionale e ambiente
Ruolo di infrastruttura	Incorporato in app o pipeline di analisi	Middleware o componente del livello di servizio	Agent Mesh componibile integrato con sistemi cloud, serverless o edge

In sintesi:

- L'intelligenza artificiale tradizionale è incentrata sugli strumenti e funzionalmente limitata. Si concentra sulla previsione o sulla classificazione.
- Gli agenti software tradizionali introducono l'autonomia e la comunicazione di base, ma spesso sono vincolati da regole o statici.
- L'intelligenza artificiale agentica unisce autonomia, asincronia e agenzia. Consente entità intelligenti e orientate agli obiettivi in grado di ragionare, agire e adattarsi all'interno di sistemi complessi. Ciò rende l'intelligenza artificiale agentica ideale per le future future basate sull'intelligenza artificiale e native del cloud.

Fasi successive

Questa guida ha discusso la storia e le basi dell'intelligenza artificiale agentica, che rappresenta l'evoluzione degli agenti software tradizionali in sistemi autonomi e intelligenti alimentati dall'intelligenza artificiale generativa. Ha descritto come i primi agenti software seguissero regole e logiche predefinite per automatizzare le attività entro limiti prestabiliti e ha spiegato come l'intelligenza artificiale agentica si basi su queste basi incorporando modelli linguistici di grandi dimensioni, che consentono agli agenti di ragionare, apprendere e adattarsi dinamicamente in ambienti aperti.

Puoi approfondire l'intelligenza artificiale agentica consultando le seguenti pubblicazioni di questa serie:

- L'operazionalizzazione dell'intelligenza artificiale agentica su AWS fornisce una strategia organizzativa per trasformare l'intelligenza artificiale agentica da esperimenti isolati in un'infrastruttura generatrice di valore su scala aziendale.
- I modelli e i flussi di lavoro di Agentic AI su AWS illustrano i modelli fondamentali e i costrutti modulari utilizzati per progettare, comporre e orchestrare agenti di intelligenza artificiale orientati agli obiettivi.
- I framework, i protocolli e gli strumenti di Agentic AI AWS coprono le basi del software, i toolkit e i protocolli da prendere in considerazione quando si creano soluzioni di intelligenza artificiale agentica.
- Building serverless architecture for agentic AI on AWS illustra le architetture serverless come base naturale dei moderni carichi di lavoro di intelligenza artificiale e descrive come è possibile creare architetture serverless native per l'intelligenza artificiale in. Cloud AWS
- La creazione di architetture multi-tenant per l'intelligenza artificiale agentica su descrive l'uso degli agenti di intelligenza artificiale in ambienti multi-tenant, incluse considerazioni sull' AWShosting, modelli di implementazione e piani di controllo.

Risorse

Per ulteriori informazioni sui concetti discussi in questa guida, consulta le guide e gli articoli seguenti.

AWS riferimenti

- Agenti Amazon Bedrock
- Amazon Q Developer
- Strands Agents SDK

Altri riferimenti

- Hewitt, Carl, Peter Bishop e Richard Steiger. «Un formalismo universale modulare ACTOR per l'intelligenza artificiale». Atti della 3a Conferenza Internazionale Congiunta sull'Intelligenza Artificiale (1973): 235-245. https://www.ijcai. org/Proceedings/73/Papers/027B.pdf
- Lesser, Victor R., pubblicazioni pertinenti (vedi elenco completo):
  - Lesser, Victor R. e Daniel D. Corkill. «Sistemi distribuiti cooperativi e funzionalmente accurati». Transazioni IEEE su Systems, Man e Cybernetics 11, n. 1 (1981): 81-96. https://ieeexplore.ieee.org/abstract/document/4308581
  - Decker, Keith S. e Victor R. Lesser. «La comunicazione al servizio del coordinamento». Workshop AAAI sulla pianificazione della comunicazione interagente (1994). https://www.researchgate. net/profile/Victor-Lesser/publication/2768884_Communication_in_the_Service_of_Coordination/links/00b7d51cc2a0750cb4000000/Communication-in-the-Service-of-Coordination.pdf
  - Durfee, Edmund H., Victor R. Lesser e Daniel D. Corkill. «Tendenze nella risoluzione cooperativa distribuita dei problemi». Transazioni IEEE sulla conoscenza e l'ingegneria dei dati (1989). http://mas.cs.umass. edu/Documents/ieee-tkde89.pdf
  - Durfee, Edmund H., V.R. Lesser e D.D. Corkill, «Intelligenza artificiale distribuita». Cooperazione attraverso la comunicazione in una rete distribuita di risoluzione dei problemi (1987): 29-58. https://www.academia. edu/download/79885643/durf94_1.pdf
  - Laasri, Brigitte, Hassan Laasri, Susan Lander e Victor Lesser. «Un modello generico per agenti negoziali intelligenti». Giornale internazionale dei sistemi informativi cooperativi 01, n. 02 (1992): 291-317. https://doi.org/10.1142/S0218215792000210

  - Lander, Susan E. e Victor R. Lesser. «Comprensione del ruolo della negoziazione nella ricerca distribuita tra agenti eterogenei». IJCAI'93: Atti della tredicesima conferenza internazionale congiunta sull'intelligenza artificiale (1993): 438-444. https://www.ijcai. org/Proceedings/93-1/Papers/062.pdf
  - Lander, Susan, Victor R. Lesser e Margaret E. Connell. «Strategie di risoluzione dei conflitti per agenti esperti cooperanti» CKBS'90: Atti della Conferenza di lavoro internazionale sui sistemi cooperativi basati sulla conoscenza (ottobre 1990): 183-200. https://doi.org/10.1007/978-1-4471-1831-2_10
  - Prasad, MV Nagendra, Victor Lesser e Susan E. Lander. «Esperimenti di apprendimento in un sistema multiagente eterogeneo». Workshop IJCAI-95 sull'adattamento e l'apprendimento nei sistemi multiagente (1995): 59-64. https://www.researchgate.net/publication/2784280_Learning_Experiments_in_a_Heterogeneous_Multi-agent_System
- Nwana, Hyacinth S. «Agenti software: una panoramica». Knowledge Engineering Review 11, n. 3 (ottobre/novembre 1996): 205-244. https://teaching.shu.ac. uk/aces/rh1/elearning/multiagents/introduction/nwana.pdf
- Selfridge, Oliver G. «Pandemonium: un paradigma per l'apprendimento». Meccanizzazione dei processi di pensiero: atti di un simposio tenuto presso il National Physical Laboratory 1 (1959): 511-529. https://aitopics. org/download/classics. ----SEP----:504e1bac
- Vaswani, Ashish, Noam Shazer, Niki Parmar, Jakob Uszkoreit, Lion Jones, Aidan N. Gomez, Lukasz Kaiser e Illia Polosukhin. «L'attenzione è tutto ciò di cui hai bisogno». Atti della 31a conferenza sui sistemi di elaborazione delle informazioni neurali (NIPS). Progressi nei sistemi di elaborazione delle informazioni neurali 30 (2017): 5998-6008. https://proceedings.neurips. cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- Wooldridge, Michael e Nicholas R. Jennings. «Agenti intelligenti: teoria e pratica». Knowledge Engineering Review 10, n. 2 (gennaio 1995): 115-152. https://www.cs.cmu.edu/ ~ _intelligent_agents.pdf motionplanning/papers/sbp_papers/integrated1/woodridge

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:

- Rifattorizzare/riprogettare: trasferisci un'applicazione e modifica la sua architettura sfruttando appieno le funzionalità native del cloud per migliorare l'agilità, le prestazioni e la scalabilità. Ciò comporta in genere la portabilità del sistema operativo e del database. Esempio: migra il tuo database Oracle locale all'edizione compatibile con Amazon Aurora PostgreSQL.
- Ridefinire la piattaforma (lift and reshape): trasferisci un'applicazione nel cloud e introduci un certo livello di ottimizzazione per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale ad Amazon Relational Database Service (Amazon RDS) per Oracle in. Cloud AWS
- Riacquistare (drop and shop): passa a un prodotto diverso, in genere effettuando la transizione da una licenza tradizionale a un modello SaaS. Esempio: migra il tuo sistema di gestione delle relazioni con i clienti (CRM) su Salesforce.com.
- Eseguire il rehosting (lift and shift): trasferisci un'applicazione sul cloud senza apportare modifiche per sfruttare le funzionalità del cloud. Esempio: migra il database Oracle locale su Oracle su un'istanza in. EC2 Cloud AWS
- Trasferire (eseguire il rehosting a livello hypervisor): trasferisci l'infrastruttura sul cloud senza acquistare nuovo hardware, riscrivere le applicazioni o modificare le operazioni esistenti. Si esegue la migrazione dei server da una piattaforma locale a un servizio cloud per la stessa piattaforma. Esempio: migra un'applicazione 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

ABAC

Vedi controllo degli accessi basato sugli attributi.

servizi astratti

Vedi servizi gestiti.

ACIDO

Vedi atomicità, consistenza, isolamento, durata.

migrazione attiva-attiva

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

migrazione attiva-passiva

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

funzione di aggregazione

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

Intelligenza artificiale

Vedi intelligenza artificiale.

AIOps

Guarda le operazioni di intelligenza artificiale.

anonimizzazione

Il processo di eliminazione permanente delle informazioni personali in un set di dati. L'anonimizzazione può aiutare a proteggere la privacy personale. I dati anonimi non sono più considerati dati personali.

anti-modello

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

controllo delle applicazioni

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

portfolio di applicazioni

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

intelligenza artificiale (IA)

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

operazioni di intelligenza artificiale (AIOps)

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

crittografia asimmetrica

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

atomicità, consistenza, isolamento, durabilità (ACID)

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

Controllo degli accessi basato su attributi (ABAC)

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

fonte di dati autorevole

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

Zona di disponibilità

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

AWS Cloud Adoption Framework (CAF)AWS

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

AWS Workload Qualification Framework (WQF)AWS

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

B

bot difettoso

Un bot che ha lo scopo di interrompere o causare danni a individui o organizzazioni.

BCP

Vedi la pianificazione della continuità operativa.

grafico comportamentale

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

sistema big-endian

Un sistema che memorizza per primo il byte più importante. Vedi anche endianness.

Classificazione binaria

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

filtro Bloom

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

implementazione blu/verde

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

bot

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

botnet

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

ramo

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

accesso break-glass

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

strategia brownfield

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

cache del buffer

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

capacità di business

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

pianificazione della continuità operativa (BCP)

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

C

CAF

Vedi Cloud Adoption AWS Framework.

implementazione canaria

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

CCoE

Vedi Cloud Center of Excellence.

CDC

Vedi Change Data Capture.

Change Data Capture (CDC)

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

ingegneria del caos

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

CI/CD

Vedi integrazione continua e distribuzione continua.

classificazione

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

crittografia lato client

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

Centro di eccellenza cloud (CCoE)

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

cloud computing

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

modello operativo cloud

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

fasi di adozione del cloud

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

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

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

CMDB

Vedi database di gestione della configurazione.

repository di codice

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

cache fredda

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

dati freddi

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

visione artificiale (CV)

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

deriva della configurazione

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

database di gestione della configurazione (CMDB)

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

Pacchetto di conformità

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

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

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

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

CV

Vedi visione artificiale.

D

dati a riposo

Dati stazionari nella rete, ad esempio i dati archiviati.

classificazione dei dati

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

deriva dei dati

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

dati in transito

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

rete di dati

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

riduzione al minimo dei dati

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

perimetro dei dati

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

pre-elaborazione dei dati

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

provenienza dei dati

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

soggetto dei dati

Un individuo i cui dati vengono raccolti ed elaborati.

data warehouse

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

linguaggio di definizione del database (DDL)

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

linguaggio di manipolazione del database (DML)

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

DDL

Vedi linguaggio di definizione del database.

deep ensemble

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

deep learning

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

defense-in-depth

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

amministratore delegato

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

implementazione

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

Ambiente di sviluppo

Vedi ambiente.

controllo di rilevamento

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

mappatura del flusso di valore dello sviluppo (DVSM)

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

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

gemello digitale

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

tabella delle dimensioni

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

disastro

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

disaster recovery (DR)

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

DML

Vedi linguaggio di manipolazione del database.

progettazione basata sul dominio

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

DOTT.

Vedi disaster recovery.

rilevamento della deriva

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

DVSM

Vedi la mappatura del flusso di valore dello sviluppo.

E

EDA

Vedi analisi esplorativa dei dati.

MODIFICA

Vedi scambio elettronico di dati.

edge computing

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

scambio elettronico di dati (EDI)

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

crittografia

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

chiave crittografica

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

endianità

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

endpoint

Vedi service endpoint.

servizio endpoint

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

pianificazione delle risorse aziendali (ERP)

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

crittografia envelope

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

ambiente

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

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

- ambienti superiori: tutti gli ambienti a cui possono accedere utenti diversi dal team di sviluppo principale. Si può trattare di un ambiente di produzione, ambienti di preproduzione e ambienti per i test di accettazione da parte degli utenti.

epica

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

ERP

Vedi pianificazione delle risorse aziendali.

analisi esplorativa dei dati (EDA)

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

F

tabella dei fatti

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

fallire velocemente

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

limite di isolamento dei guasti

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

ramo di funzionalità

Vedi filiale.

caratteristiche

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

importanza delle caratteristiche

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

trasformazione delle funzionalità

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

prompt con pochi scatti

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

FGAC

Vedi il controllo granulare degli accessi.

controllo granulare degli accessi (FGAC)

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

migrazione flash-cut

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

FM

Vedi modello di base.

modello di fondazione (FM)

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

G

IA generativa

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

blocco geografico

Vedi restrizioni geografiche.

limitazioni geografiche (blocco geografico)

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

Flusso di lavoro di GitFlow

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

immagine dorata

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

strategia greenfield

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

guardrail

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

H

AH

Vedi disponibilità elevata.

migrazione di database eterogenea

Migrazione del database di origine in un database di destinazione che utilizza un motore di database diverso (ad esempio, da Oracle ad Amazon Aurora). La migrazione eterogenea fa in genere parte di uno sforzo di riprogettazione e la conversione dello schema può essere un'attività complessa. AWS offre AWS SCT che aiuta con le conversioni dello schema.

alta disponibilità (HA)

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

modernizzazione storica

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

dati di blocco

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

migrazione di database omogenea

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

dati caldi

Dati a cui si accede frequentemente, 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

Considera l'infrastruttura come codice.

Policy basata su identità

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

applicazione inattiva

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

IIoT

Vedi Industrial Internet of Things.

infrastruttura immutabile

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

VPC in ingresso (ingress)

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

migrazione incrementale

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

Industria 4.0

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

infrastruttura

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

infrastruttura come codice (IaC)

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

IIoInternet delle cose industriale (T)

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

VPC di ispezione

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

Internet of Things (IoT)

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

interpretabilità

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

IoT

Vedi Internet of Things.

libreria di informazioni IT (ITIL)

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

gestione dei servizi IT (ITSM)

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

ITIL

Vedi la libreria di informazioni IT.

ITSM

Vedi Gestione dei servizi IT.

L

controllo degli accessi basato su etichette (LBAC)

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

zona di destinazione

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

modello linguistico di grandi dimensioni (LLM)

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

migrazione su larga scala

Una migrazione di 300 o più server.

BIANCO

Vedi controllo degli accessi basato su etichette.

Privilegio minimo

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

eseguire il rehosting (lift and shift)

Vedi 7 R.

sistema little-endian

Un sistema che memorizza per primo il byte meno importante. Vedi anche endianità.

LLM

Vedi modello linguistico di grandi dimensioni.

ambienti inferiori

Vedi ambiente.

M

machine learning (ML)

Un tipo di intelligenza artificiale che utilizza algoritmi e tecniche per il riconoscimento e l'apprendimento di schemi. Il machine learning analizza e apprende dai dati registrati, come i dati dell'Internet delle cose (IoT), per generare un modello statistico basato su modelli. Per ulteriori informazioni, consulta la sezione Machine learning.

ramo principale

Vedi filiale.

malware

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

servizi gestiti

Servizi AWS per cui AWS gestisce il livello di infrastruttura, il sistema operativo e le piattaforme e si accede agli endpoint per archiviare e recuperare i dati. Amazon Simple Storage Service

(Amazon S3) Simple Storage Service (Amazon S3) e Amazon DynamoDB sono esempi di servizi gestiti. Questi sono noti anche come servizi astratti.

sistema di esecuzione della produzione (MES)

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

MAP

Vedi Migration Acceleration Program.

meccanismo

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

account membro

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

MEH

Vedi sistema di esecuzione della produzione.

Message Queuing Telemetry Transport (MQTT)

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

microservizio

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

architettura di microservizi

Un approccio alla creazione di un'applicazione con componenti indipendenti che eseguono ogni processo applicativo come microservizio. Questi microservizi comunicano attraverso un'interfaccia

ben definita utilizzando sistemi leggeri. APIs Ogni microservizio in questa architettura può essere aggiornato, distribuito e dimensionato per soddisfare la richiesta di funzioni specifiche di un'applicazione. Per ulteriori informazioni, vedere Implementazione dei microservizi su. AWS

Programma di accelerazione della migrazione (MAP)

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

migrazione su larga scala

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

fabbrica di migrazione

Team interfunzionali che semplificano la migrazione dei carichi di lavoro attraverso approcci automatizzati e agili. I team di Migration Factory in genere includono addetti alle operazioni, analisti e proprietari aziendali, ingegneri addetti alla migrazione, sviluppatori e DevOps professionisti che lavorano nell'ambito degli sprint. Tra il 20% e il 50% di un portfolio di applicazioni aziendali è costituito da schemi ripetuti che possono essere ottimizzati con un approccio di fabbrica. Per ulteriori informazioni, consulta la discussione sulle fabbriche di migrazione e la Guida alla fabbrica di migrazione al cloud in questo set di contenuti.

metadati di migrazione

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

modello di migrazione

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

Valutazione del portfolio di migrazione (MPA)

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

valutazione della preparazione alla migrazione (MRA)

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

strategia di migrazione

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

ML

Vedi machine learning.

modernizzazione

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

valutazione della preparazione alla modernizzazione

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

applicazioni monolitiche (monoliti)

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

MAPPA

Vedi Migration Portfolio Assessment.

MQTT

Vedi Message Queuing Telemetry Transport.

classificazione multiclasse

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

infrastruttura mutabile

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

O

OAC

Vedi Origin Access Control.

QUERCIA

Vedi Origin Access Identity.

OCM

Vedi gestione delle modifiche organizzative.

migrazione offline

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

OI

Vedi l'integrazione delle operazioni.

OLA

Vedi accordo a livello operativo.

migrazione online

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

OPC-UA

Vedi Open Process Communications - Unified Architecture.

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

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

accordo a livello operativo (OLA)

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

revisione della prontezza operativa (ORR)

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

tecnologia operativa (OT)

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

integrazione delle operazioni (OI)

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

trail organizzativo

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

gestione del cambiamento organizzativo (OCM)

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

controllo dell'accesso all'origine (OAC)

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

identità di accesso origine (OAI)

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

ORR

Vedi la revisione della prontezza operativa.

- NON

Vedi la tecnologia operativa.

VPC in uscita (egress)

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

P

limite delle autorizzazioni

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

informazioni di identificazione personale (PII)

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

Informazioni che consentono l'identificazione personale degli utenti

Visualizza le informazioni di identificazione personale.

playbook

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

PLC

Vedi controllore logico programmabile.

PLM

Vedi la gestione del ciclo di vita del prodotto.

policy

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

persistenza poliglotta

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

valutazione del portfolio

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

predicate

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

predicato pushdown

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

controllo preventivo

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

principale

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

privacy fin dalla progettazione

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

zone ospitate private

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

controllo proattivo

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

gestione del ciclo di vita del prodotto (PLM)

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

Ambiente di produzione

Vedi ambiente.

controllore logico programmabile (PLC)

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

concatenamento rapido

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

pseudonimizzazione

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

publish/subscribe (pub/sub)

Un modello che consente comunicazioni asincrone tra microservizi per migliorare la scalabilità e la reattività. Ad esempio, in un MES basato su microservizi, un microservizio può pubblicare

messaggi di eventi su un canale a cui altri microservizi possono abbonarsi. Il sistema può aggiungere nuovi microservizi senza modificare il servizio di pubblicazione.

Q

Piano di query

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

regressione del piano di query

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

R

Matrice RACI

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

STRACCIO

Vedi Retrieval Augmented Generation.

ransomware

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

Matrice RASCI

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

RCAC

Vedi controllo dell'accesso a righe e colonne.

replica di lettura

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

riprogettare

Vedi 7 Rs.

obiettivo del punto di ripristino (RPO)

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

obiettivo del tempo di ripristino (RTO)

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

rifattorizzare

Vedi 7 R.

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 EC2 Amazon o la rotazione delle credenziali.

Crittografia lato server

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

Policy di controllo dei servizi (SCP)

Una politica che fornisce il controllo centralizzato sulle autorizzazioni per tutti gli account di un'organizzazione in. AWS Organizations SCPs definire barriere o fissare limiti alle azioni

che un amministratore può delegare a utenti o ruoli. È possibile utilizzarli SCPs come elenchi consentiti o elenchi di rifiuto, per specificare quali servizi o azioni sono consentiti o proibiti. Per ulteriori informazioni, consulta le politiche di controllo del servizio nella AWS Organizations documentazione.

endpoint del servizio

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

accordo sul livello di servizio (SLA)

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

indicatore del livello di servizio (SLI)

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

obiettivo a livello di servizio (SLO)

Una metrica target che rappresenta lo stato di un servizio, misurato da un indicatore del livello di servizio.

Modello di responsabilità condivisa

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

SIEM

Vedi il sistema di gestione delle informazioni e degli eventi sulla sicurezza.

punto di errore singolo (SPOF)

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

SLAM

Vedi il contratto sul livello di servizio.

SLI

Vedi l'indicatore del livello di servizio.

LENTA

Vedi obiettivo del livello di servizio.

split-and-seed modello

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

SPOF

Vedi punto di errore singolo.

schema a stella

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

modello del fico strangolatore

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

sottorete

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

controllo di supervisione e acquisizione dati (SCADA)

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

crittografia simmetrica

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

test sintetici

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

prompt di sistema

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

T

tags

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

variabile di destinazione

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

elenco di attività

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

Ambiente di test

Vedi ambiente.

training

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

Transit Gateway

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

flusso di lavoro basato su trunk

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

Accesso attendibile

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

regolazione

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

team da due pizze

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

U

incertezza

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

compiti indifferenziati

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

ambienti superiori

Vedi ambiente.

V

vacuum

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

controllo delle versioni

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

Peering VPC

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

vulnerabilità

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

W

cache calda

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

dati caldi

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

funzione finestra

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

Carico di lavoro

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

flusso di lavoro

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

VERME

Vedi scrivere una volta, leggere molti.

WQF

Vedi AWS Workload Qualification Framework.

scrivi una volta, leggi molte (WORM)

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

Z

exploit zero-day

Un attacco, in genere malware, che sfrutta una vulnerabilità zero-day.

vulnerabilità zero-day

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

prompt zero-shot

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

applicazione zombie

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

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