Modelli e flussi di lavoro di intelligenza artificiale agentica su AWS

AWS Guida prescrittiva

AWS Guida prescrittiva: Modelli e flussi di lavoro di intelligenza artificiale agentica su AWS

Table of Contents

Introduzione 1
Destinatari principali 1
Obiettivi 1
Informazioni su questa serie di contenuti 2
Schemi degli agenti 3
Agenti di ragionamento di base 4
Architecture 4
Description 5
Funzionalità 6
Limitazioni 6
Casi di utilizzo comune 6
Guida all'implementazione 7
Riepilogo 7
Architecture 7
Description 8
Funzionalità 9
Casi di utilizzo comune 9
Guida all'implementazione 9
Riepilogo 10
Agenti basati su strumenti per la chiamata di funzioni 10
Architecture 10
Description 11
Funzionalità 12
Casi di utilizzo comune 12
Guida all'implementazione 12
Riepilogo 13
Agenti basati su strumenti per server 13
Architecture 13
Description 14
Funzionalità 15
Casi di utilizzo comune 15
Guida all'implementazione 15
Riepilogo 16
Agenti per uso informatico 16

Architecture 16
Description 17
Funzionalità 18
Casi di utilizzo comune 18
Guida all'implementazione 19
Riepilogo 19
Agenti di codifica 19
Architecture 19
Description 20
Funzionalità 21
Casi di utilizzo comune 21
Guida all'implementazione 22
Riepilogo 22
Agenti vocali e vocali 22
Architecture 22
Description 23
Funzionalità 24
Casi di utilizzo comune 24
Guida all'implementazione 25
Riepilogo 25
Agenti di orchestrazione del flusso di lavoro 25
Architecture 26
Description 26
Funzionalità 27
Casi di utilizzo comune 27
Guida all'implementazione 27
Riepilogo 28
Agenti con memoria aumentata 28
Architecture 28
Description 29
Funzionalità 30
Casi di utilizzo comune 30
Implementazione di agenti con memoria aumentata 30
Implementazione dei prompt iniettati dalla memoria 31
Riepilogo 32
Agenti di simulazione e test 32

Architecture 32
Description 33
Funzionalità 34
Casi di utilizzo comune 34
Guida all'implementazione 35
Riepilogo 36
Agenti osservatori e di monitoraggio 36
Architecture 37
Description 37
Funzionalità 38
Casi di utilizzo comune 38
Guida all'implementazione 38
Riepilogo 39
Collaborazione tra più agenti 39
Description 41
Funzionalità 42
Casi di utilizzo comune 42
Guida all'implementazione 42
Riepilogo 43
Conclusioni 43
Da asporto 44
Flussi di lavoro LLM 45
Panoramica della cognizione aumentata con LLM 45
Flusso di lavoro per il concatenamento rapido 46
Descrizione 47
Funzionalità 47
Casi di utilizzo comune 48
Flusso di lavoro per il routing 48
Funzionalità 49
Casi di utilizzo comune 49
Flusso di lavoro per la parallelizzazione 49
Funzionalità 51
Casi di utilizzo comune 51
Flusso di lavoro per l'orchestrazione 51
Funzionalità 53
Casi di utilizzo comune 53

Flusso di lavoro per valutatori e cicli Reflect-Refine 53
Casi di utilizzo comune 54
Funzionalità 54
Conclusioni 54
Modelli di flusso di lavoro agentici 56
Dai sistemi basati sugli eventi a quelli basati sulla cognizione aumentata 56
Architettura basata su eventi 56
Flussi di lavoro basati sulla cognizione 57
Approfondimenti fondamentali 59
Rapido concatenamento degli schemi della saga 59
Coreografia saga 59
Schema di concatenamento rapido 60
Coreografia dell'agente 60
Da asporto 63
Schemi di invio dinamici di routing 63
Invio dinamico 64
Routing basato su LLM 65
Router agente 66
Da asporto 67
Parallelizzazione e schemi di dispersione 67
Scatter-gather 68
Parallelizzazione basata su LLM (cognizione dispersa e raccolta) 70
Parallelizzazione degli agenti 70
Take-away 71
Schemi di orchestrazione Saga 71
Orchestrazione di eventi 72
Sistema di agenti basato sui ruoli (orchestratore) 73
Supervisore 73
Cose da asporto 75
Evaluator Reflect-Refine Loop Patterns 75
Circuito di controllo del feedback 76
Circuito di controllo del feedback (valutatore) 78
Valutatore 78
Cose da asporto 79
Progettazione di flussi di lavoro agentici su AWS 79
Conclusioni 80

Cronologia dei documenti 82
Glossario 83
..... 83
A 84
B 87
C 89
D 92
E 96
F 98
G 100
H 101
I 103
L 105
M 106
O 111
P 113
Q 116
R 117
S 120
T 124
U 125
V 126
W 126
Z 128
..... cxxix

Modelli e flussi di lavoro di intelligenza artificiale agentica su AWS

Aaron Sempf e Andrew Hooker, Amazon Web Services

Luglio 2025 (cronologia dei documenti)

Le organizzazioni stanno adottando modelli linguistici di grandi dimensioni (LLMs) e agenti software per risolvere problemi dinamici e multidominio utilizzando una nuova disciplina architettonica chiamata agentic pattern. I pattern agentici sono modelli fondamentali e costrutti modulari utilizzati per progettare e orchestrare agenti di intelligenza artificiale orientati agli obiettivi in molti contesti.

Destinatari principali

Questa guida è destinata ad architetti, sviluppatori e leader di prodotto che desiderano creare applicazioni intelligenti che vadano oltre la logica statica, la logica simbolica e l'automazione deterministica.

Obiettivi

Questa guida fornisce un framework di progettazione e un approccio di implementazione per sistemi di agenti AI che operano in modo autonomo pur rimanendo controllabili e allineati agli obiettivi. Collega modelli architettonici basati sugli eventi con varie alternative agentiche, dimostrando come creare sistemi di agenti di livello di produzione utilizzando architetture native del cloud. In questa guida vengono trattati i seguenti argomenti:

- Modelli di agenti: i modelli di agente sono modelli di progettazione riutilizzabili che descrivono la struttura e il comportamento dei singoli agenti. Ciò include agenti di ragionamento, agenti di recupero potenziato, agenti di codifica, interfacce vocali, orchestratori di flussi di lavoro e sistemi collaborativi multiagente. Ogni modello illustra il modo in cui gli agenti percepiscono, ragionano, agiscono e apprendono, mappati. Servizi AWS
- Flussi di lavoro LLM: i flussi di lavoro si concentrano sul modo in cui gli agenti utilizzano il ragionamento. LLMs Esplorano le strategie di suggerimento e i meccanismi di pianificazione e descrivono come LLMs vengono utilizzati non solo per generare testo, ma anche per promuovere comportamenti strutturati, interpretabili e affidabili all'interno di un ciclo di agenti.

- Modelli di flusso di lavoro agentici: i modelli di flusso di lavoro descrivono come più agenti, strumenti e ambienti interagiscono per formare sistemi autonomi. Ciò include modelli per l'orchestrazione delle attività, la delega dei subagenti, il coordinamento basato sugli eventi, l'osservabilità e il controllo. Questi aspetti promuovono architetture AI scalabili, componibili e verificabili.

Informazioni su questa serie di contenuti

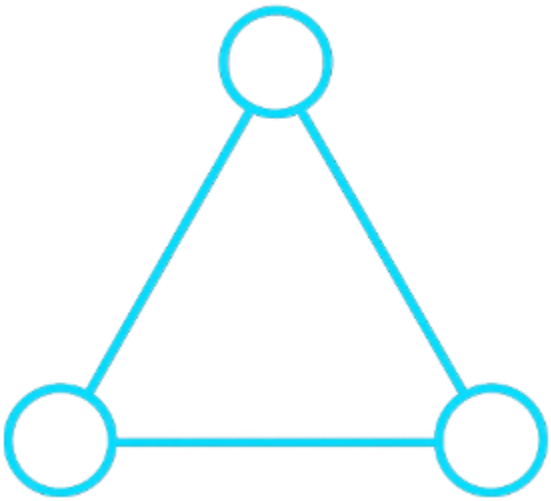
Questa guida fa parte di una serie sull'intelligenza artificiale agentica su. AWS Per ulteriori informazioni e per visualizzare le altre guide di questa serie, consulta Agentic AI sul sito Web Prescriptive Guidance. AWS

Schemi degli agenti

I pattern degli agenti sono elementi costitutivi riutilizzabili e componibili che possono essere adattati a domini, casi d'uso e livelli di complessità specifici. I sistemi Agentic differiscono, tuttavia, dalle applicazioni tradizionali. Alla base di tutti i progetti di agenti di intelligenza artificiale c'è un modello concettuale ancorato ai seguenti tre principi fondamentali:

- Asincrono: gli agenti operano in ambienti scarsamente accoppiati e ricchi di eventi
- Autonomia: gli agenti agiscono in modo indipendente, senza controllo umano o esterno
- Agenzia: gli agenti agiscono con uno scopo, per conto di un utente o di un sistema, verso obiettivi specifici

Il triangolo nel diagramma seguente rappresenta gli elementi costitutivi fondamentali di un agente software: percezione, ragione e azione. Ciò consente a un sistema agentico di osservare, prendere decisioni e agire all'interno del proprio ambiente.



In base alla progettazione, i modelli agentici forniscono un linguaggio di progettazione modulare per la creazione di sistemi di intelligenza artificiale, il che significa che sono accessibili, operativi, estensibili e pronti per la produzione. La progettazione di questi sistemi richiede un'attenzione particolare alle seguenti tre dimensioni correlate, che verranno ulteriormente discusse più avanti in questa guida.

In questa sezione

- Agenti di ragionamento di base

- Agenti basati su strumenti per la chiamata di funzioni
- Agenti basati su strumenti per server
- Agenti per uso informatico
- Agenti di codifica
- Agenti vocali e vocali
- Agenti di orchestrazione del flusso di lavoro
- Agenti con memoria aumentata
- Agenti di simulazione e test
- Agenti osservatori e di monitoraggio
- Collaborazione tra più agenti

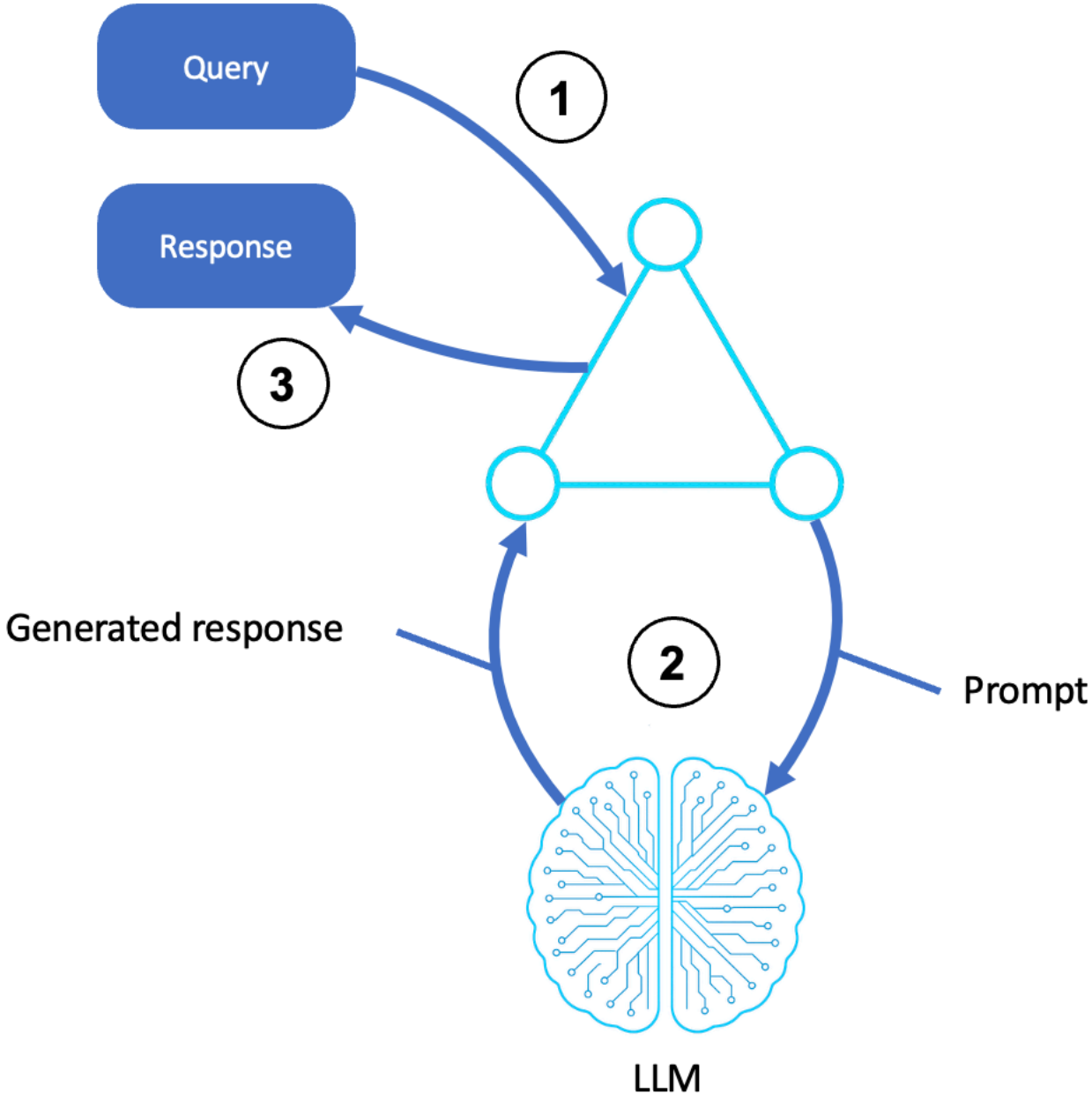
Agenti di ragionamento di base

Un agente di ragionamento di base è la forma più semplice di intelligenza artificiale agentica che esegue l'inferenza logica o il processo decisionale in risposta a una domanda. Accetta input da un utente o da un sistema ed elabora le domande e genera risposte utilizzando prompt strutturati.

Questo modello è utile per le attività che richiedono un ragionamento, una classificazione o un riepilogo in un unico passaggio in base a un determinato contesto. Non utilizza memoria, strumenti o gestione dello stato, il che lo rende privo di stato, leggero e altamente componibile in flussi di lavoro di grandi dimensioni.

Architecture

Il flusso di un agente di ragionamento di base è mostrato nel diagramma seguente:



Description

1. Riceve un input
   - Un utente, un sistema o un agente upstream invia una richiesta o un'istruzione.
   - L'input viene trasferito alla shell dell'agente o al livello di orchestrazione.
   - Questo passaggio include qualsiasi preelaborazione, creazione di modelli rapidi e identificazione degli obiettivi.

2. Richiama l'LLM
   - L'agente trasforma la query in un prompt strutturato e la invia a un LLM (ad esempio, tramite Amazon Bedrock).
   - L'LLM genera una risposta basata sul prompt utilizzando conoscenze e contesto preformati.
   - L'output generato può includere fasi di ragionamento (chain-of-thought), risposte finali o opzioni classificate.
3. Restituisce una risposta
   - L'output generato viene inoltrato all'interfaccia dell'agente.
   - Ciò può includere la formattazione, la postelaborazione o una risposta API.

Funzionalità

- Supporta il linguaggio naturale o l'input strutturato
- Utilizza la progettazione tempestiva per guidare il comportamento
- Senza stato e scalabile
- Può essere integrato nell'interfaccia utente, nella CLI e nelle APIs pipeline

Limitazioni

- Nessuna memoria o consapevolezza storica
- Nessuna interazione con strumenti o fonti di dati esterne
- Limitato a ciò che il LLM conosce al momento dell'inferenza

Casi di utilizzo comune

- Domande e risposte conversazionali
- Spiegazioni e riassunti delle politiche
- Linee guida per prendere decisioni
- Flussi di chatbot leggeri e automatizzati
- Classificazione, etichettatura e punteggio

Guida all'implementazione

È possibile utilizzare i seguenti strumenti e servizi per creare un agente di ragionamento di base:

- Amazon Bedrock per invocazioni LLM (Anthropic,, Meta) AI21
- Amazon API Gateway o AWS Lambda esporlo come microservizio stateless
- Modelli di prompt archiviati in Parameter Store o come Gestione dei segreti AWS codice

Riepilogo

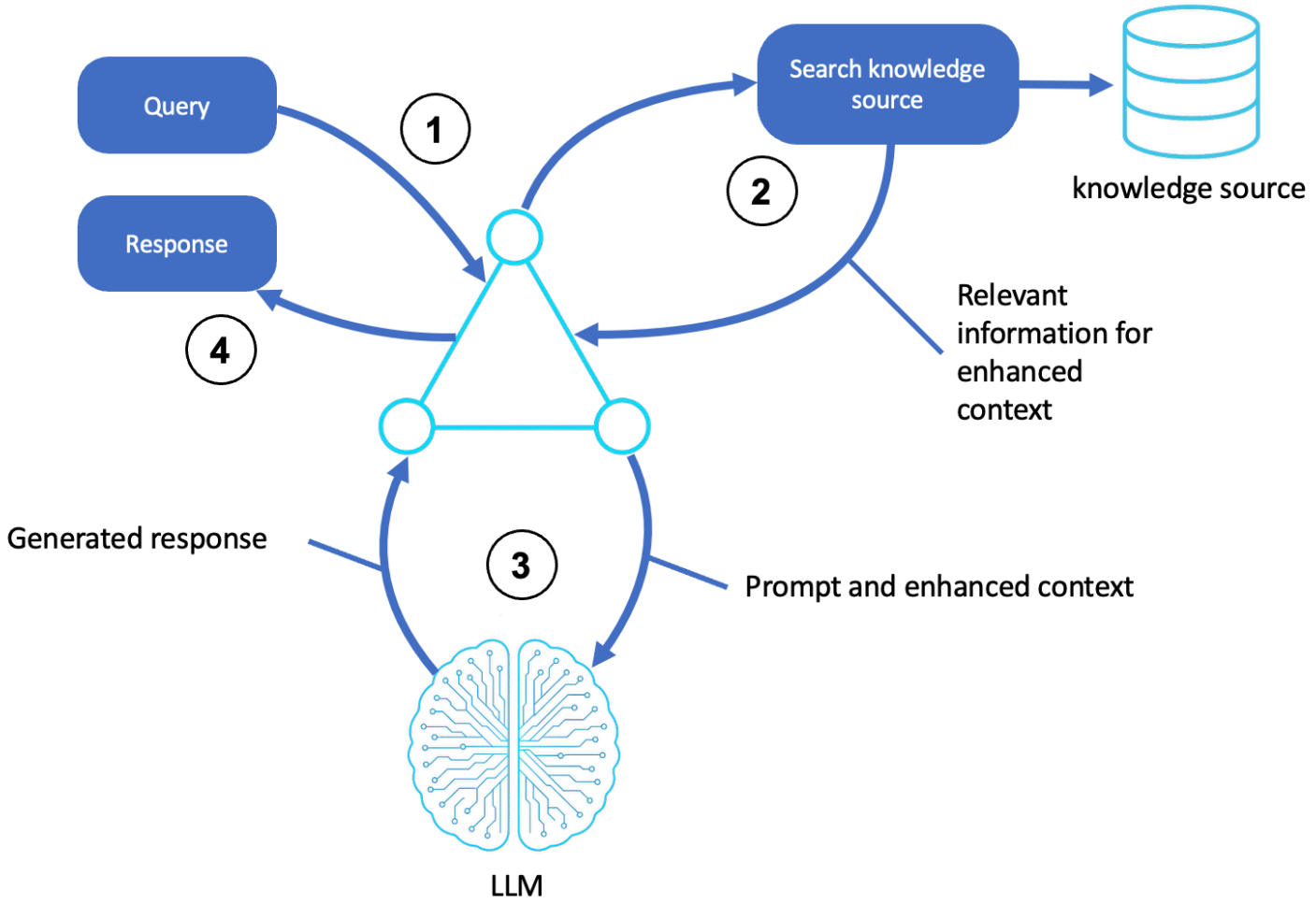
L'agente di ragionamento di base è fondamentale grazie alla sua struttura semplice. Ha funzionalità fondamentali che trasformano gli obiettivi in percorsi di ragionamento che portano a risultati intelligenti. Questo modello è spesso un punto di partenza per modelli avanzati, come agenti basati su strumenti e agenti che utilizzano la generazione aumentata di recupero (RAG). È anche un componente affidabile e modulare di flussi di lavoro di grandi dimensioni.

Agente RAG

La Retrieval-augmented generation (RAG) è una tecnica che combina il recupero delle informazioni con la generazione di testo per creare risposte accurate e contestuali. RAG consente agli agenti di recuperare informazioni esterne pertinenti prima di avviare il LLM. Estende la memoria effettiva e l'accuratezza del ragionamento di un agente basando le sue decisioni su informazioni fattuali o specifiche del dominio up-to-date. A differenza di Stateless, LLMs che si basa esclusivamente su pesi predefiniti, RAG dispone di un livello di ricerca delle conoscenze esterno che migliora dinamicamente le istruzioni in base al contesto.

Architecture

La logica del pattern RAG è illustrata nel diagramma seguente:



Description

1. Riceve una richiesta
   - Un utente o un sistema upstream invia una richiesta o un obiettivo all'agente.
   - La shell dell'agente accetta la richiesta e la formatta come richiesta di ragionamento.
2. Cerca una fonte esterna
   - L'agente identifica concetti e intenti a partire dall'interrogazione.
   - Interroga una fonte di conoscenza, ad esempio un archivio vettoriale, un database o un indice di documenti utilizzando la ricerca semantica o la corrispondenza di parole chiave.
   - I passaggi, i documenti o le entità più rilevanti vengono recuperati per essere utilizzati nella fase successiva.
3. Genera una risposta contestuale

- L'agente aumenta il prompt con le informazioni recuperate, formando un input contestuale per l'LLM.
- L'LLM elabora qualsiasi input utilizzando il ragionamento generativo (ad esempio o la riflessione) per produrre una risposta accurata. chain-of-thought

4. Restituisce l'output finale
   - L'agente prepara l'output inserendolo in qualsiasi intestazione di comunicazione o nella formattazione richiesta, quindi lo restituisce all'utente o al sistema chiamante.
   - (Facoltativo) I documenti recuperati e l'output LLM possono essere registrati, valutati e archiviati in memoria per future interrogazioni.

Funzionalità

- Output basato sui fatti anche in domini a coda lunga o specifici dell'azienda
- Estensione della memoria senza ottimizzazione del modello
- Contesto dinamico basato su ogni query e stato dell'utente
- Pienamente compatibile con database vettoriali, indici semantici e filtraggio dei metadati

Casi di utilizzo comune

- Assistenti alla conoscenza aziendale
- Bot per la conformità normativa
- Copiloti dell'assistenza clienti
- Chatbot con funzionalità di ricerca ottimizzate
- Agenti di documentazione per sviluppatori

Guida all'implementazione

Utilizzate i seguenti strumenti e servizi per creare un agente che utilizzi RAG:

- Amazon Bedrock per invocazioni LLM
- Amazon Kendra OpenSearch o Amazon Aurora per la documentazione o una ricerca di dati strutturati

- Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) per l'archiviazione di documenti
- AWS Lambda per orchestrare la ricerca, i prompt e l'inferenza LLM
- Integrazioni basate sulla conoscenza con agenti (utilizzando plug-in di memoria, retriever semantici o Amazon Bedrock)

Riepilogo

Agent RAG collega il ragionamento basato su modelli statici all'intelligenza dinamica del mondo reale. Fornisce agli agenti la capacità di cercare ciò che non sanno, sintetizzare le risposte sulla base delle conoscenze acquisite e produrre risposte verificabili e altamente affidabili.

I pattern RAG sono la base per la creazione di agenti intelligenti che scalano l'accesso alla conoscenza senza riqualificazione. Spesso è un precursore di schemi di orchestrazione più complessi che coinvolgono l'uso di strumenti, la pianificazione e la memoria a lungo termine.

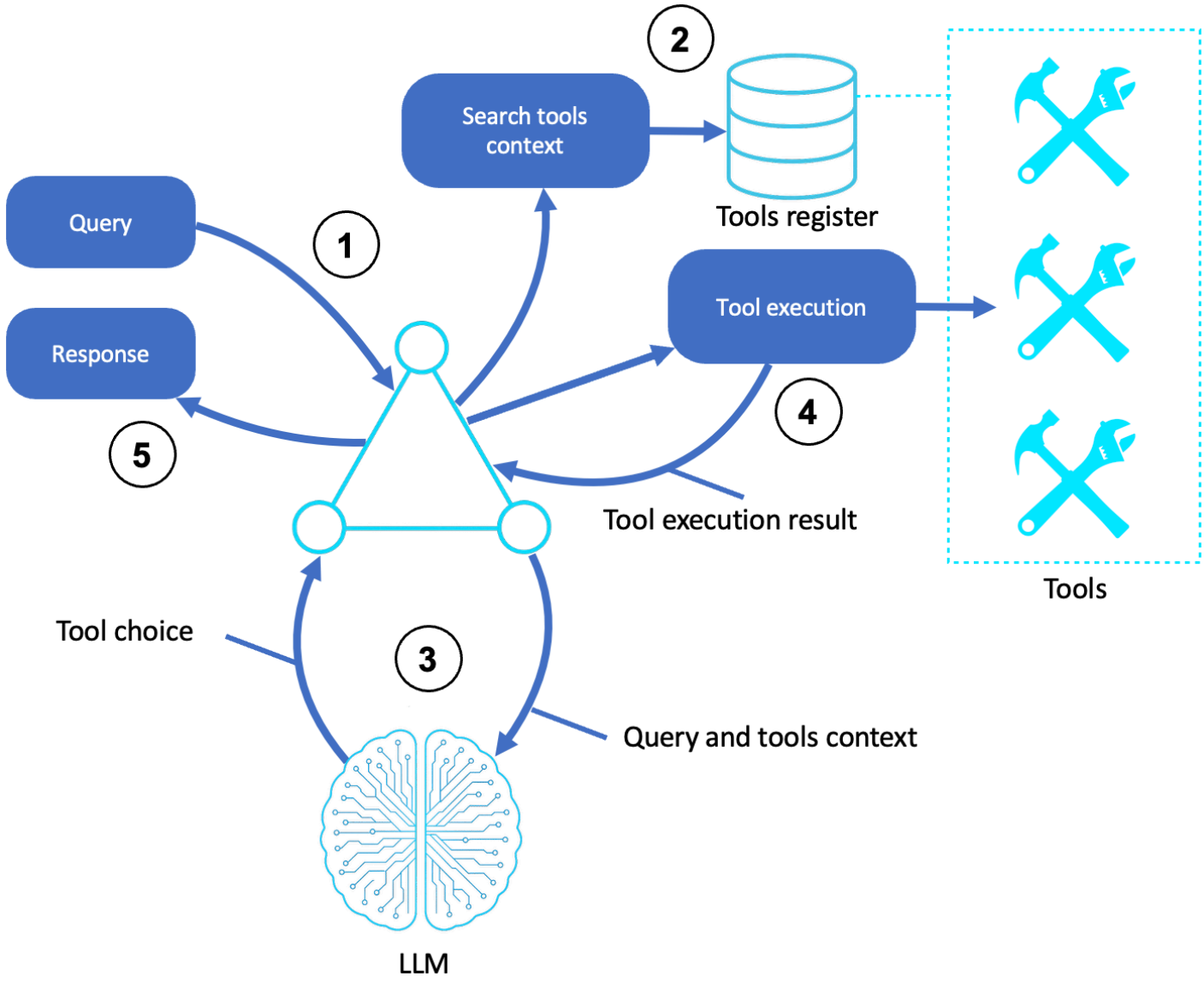
Agenti basati su strumenti per la chiamata di funzioni

Gli agenti basati su strumenti estendono le capacità degli agenti di ragionamento richiamando funzioni esterne o per completare attività che vanno oltre il ragionamento basato APIs esclusivamente sul linguaggio. Questo modello utilizza un LLM per decidere quale strumento utilizzare, quindi genera argomenti di chiamata e incorpora l'output di uno strumento nel relativo ciclo di ragionamento.

Questo modello consente agli agenti di agire anziché limitarsi a fornire risposte. L'interfaccia dello strumento rappresenta qualsiasi funzionalità richiamabile, dai calcoli aritmetici e dalle ricerche nei database ai servizi esterni e cloud. APIs

Architecture

Nel diagramma seguente è illustrato un agente basato su strumenti per la chiamata di funzioni:



Description

1. Riceve una richiesta
   - L'agente riceve una richiesta o un'attività in linguaggio naturale dall'utente o dal sistema chiamante.
2. Cerca strumenti
   - L'agente utilizza metadati interni o un registro degli strumenti per cercare gli strumenti, gli schemi e le funzionalità pertinenti disponibili.
3. Seleziona e richiama gli strumenti

- L'LLM riceve i metadati della query e dello strumento (ad esempio, nomi delle funzioni, tipi di input e descrizioni) nel relativo prompt.
- Sceglie lo strumento più pertinente, costruisce argomenti di input e restituisce una chiamata di funzione strutturata.

4. Esegue lo strumento scelto
   - La shell dell'agente o il tool runner esegue la funzione selezionata e restituisce il risultato (ad esempio, un output API, un valore del database o un calcolo).
5. Restituisce una risposta
   - L'LLM trasmette i risultati all'agente, direttamente o come parte di un prompt aggiornato. Quindi restituisce un risultato in linguaggio naturale.

Funzionalità

- Selezione dinamica degli utensili in base al contesto dell'attività
- Richiesta basata su schemi (OpenAPI, schema JSON, interfaccia funzionale) AWS
- Interpretazione dei risultati e concatenamento degli output nel ragionamento
- Operazioni senza stato o con riconoscimento della sessione

Casi di utilizzo comune

- Assistenti virtuali con accesso esterno ai dati
- Calcolatori e stimatori finanziari
- Knowledge worker basati su API
- LLMs che richiamano AWS Lambda, SageMaker endpoint Amazon e servizi SaaS

Guida all'implementazione

Usa quanto segue per creare agenti basati su strumenti per chiamare le funzioni:

- Amazon Bedrock con supporto per le chiamate di funzioni (Anthropic Claude)
- AWS Lambda come backend per l'esecuzione degli strumenti
- Amazon API Gateway o AWS Step Functions per l'orchestrazione degli strumenti

- Amazon DynamoDB o Amazon Relational Database Service (Amazon RDS) per i metadati degli strumenti sensibili al contesto
- Amazon EventBridge Pipeline o AWS Step Functions che mappano gli stati per indirizzare gli output

Riepilogo

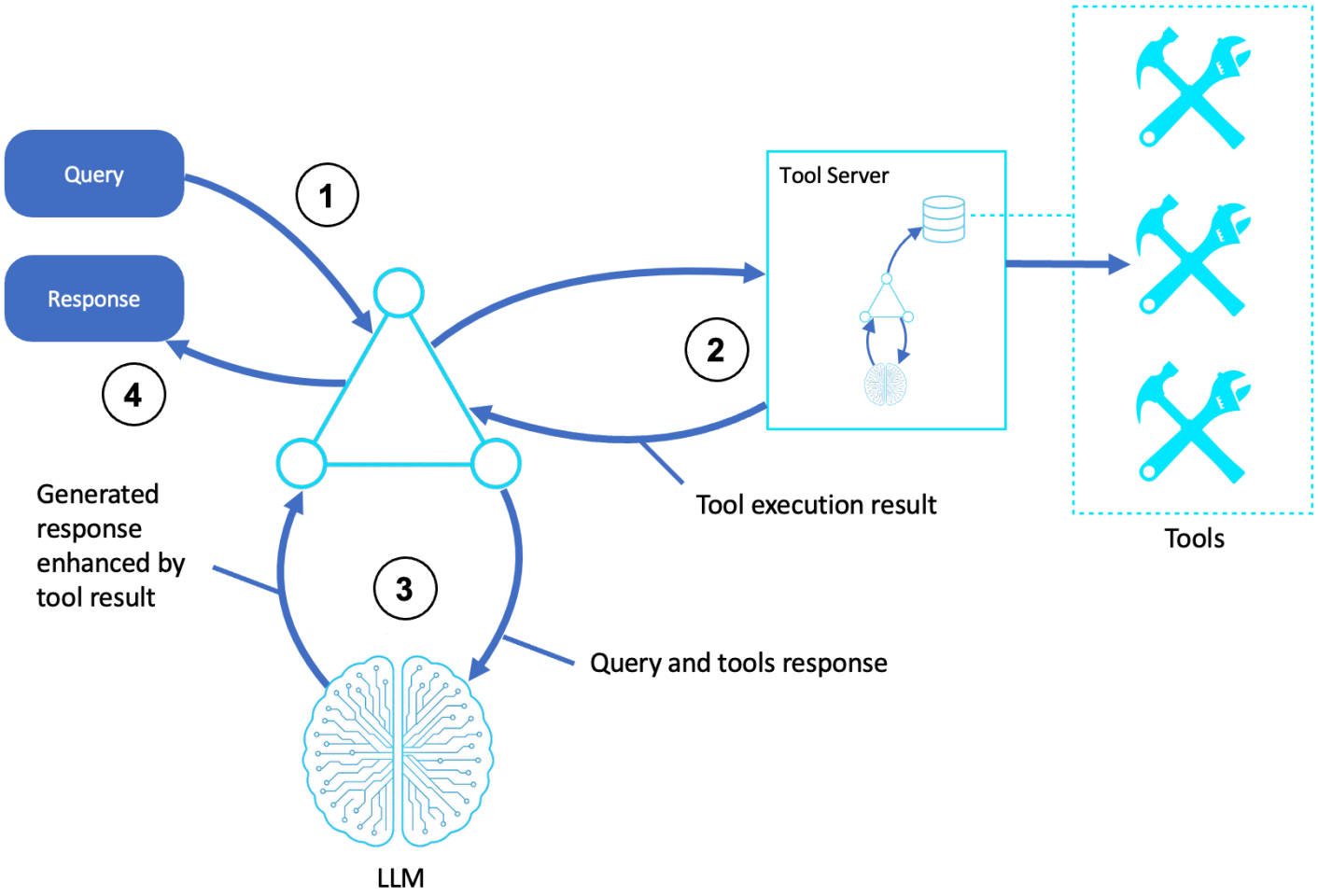
Gli agenti di chiamata di funzioni basati su strumenti rappresentano il passaggio dalla comprensione del linguaggio all'esecuzione di azioni. Questi agenti richiamano strumenti dinamici e sensibili al contesto pur mantenendo il ragionamento LLM, trasformando gli assistenti passivi in sistemi che completano le attività, accedono ai servizi e integrano le operazioni aziendali. Questo modello è una componente importante dell'intelligenza artificiale agentica in ambito aziendale, specialmente se combinato con schemi dichiarativi, framework di autorizzazione e sistemi multiagente.

Agenti basati su strumenti per server

Gli agenti per server basati su strumenti migliorano gli agenti che richiamano le funzioni delegando l'esecuzione degli strumenti a un server esterno dotato di un ambiente di runtime dedicato per strumenti, script e agenti compositi. A differenza delle chiamate di funzioni in linea che l'agent loop seleziona e richiama, gli agenti basati su server esternalizzano la logica e la pipeline di esecuzione ad altri agenti o sistemi. Ciò fornisce funzionalità avanzate come il concatenamento di più strumenti, l'esecuzione isolata e il ragionamento specializzato. I tool server sono ideali per azioni complesse, basate sullo stato o che richiedono molte risorse, in cui gli strumenti stessi possono comportare modelli di intelligenza artificiale, regole aziendali o ambienti separati.

Architecture

Di seguito è riportato uno schema per agenti basati su strumenti per server:



Description

1. Riceve una richiesta
   - Un utente o un sistema invia una richiesta alla shell dell'agente.
   - L'agente interpreta la query e si prepara a inviarla a un server di strumenti.
2. Esegue i processi del tool server
   - L'agente invia l'operazione, insieme ai parametri strutturati, a un server di strumenti.
   - Il tool server può quindi:
     - Esegui script o logica in sistemi di elaborazione dedicati (ad esempio AWS Lambda, container o Amazon) SageMaker
     - Utilizza il proprio subagente con il ragionamento LLM per selezionare ed eseguire gli strumenti
     - Gestisci dipendenze, nuovi tentativi o flussi di esecuzione in più fasi
     - Invia i risultati all'agente principale quando l'attività è completa

3. Utilizza il ragionamento LLM con l'output dello strumento
    - L'agente richiama un LLM, passando la query originale e il risultato del toolserver come parte del prompt.
    - L'LLM sintetizza una risposta che incorpora le informazioni appena acquisite.
4. Restituisce una risposta
    - L'agente restituisce una risposta in linguaggio naturale o strutturata all'utente o al sistema chiamante.
    - (Facoltativo) I risultati possono essere archiviati in memoria o nei registri di controllo.

Funzionalità

- Gli strumenti vengono richiamati al di fuori del ciclo di esecuzione dell'agente principale
- L'esecuzione degli strumenti può coinvolgere chiamate LLM, catene logiche o subagenti
- L'agente funge da controller o dispatcher, non solo da strumento wrapper
- Consente la componibilità, la scalabilità e l'isolamento della logica

Casi di utilizzo comune

- Orchestrazione di catene di modelli (ad esempio, combinando LLM, visione e codice)
- Pipeline di automazione basate sull'intelligenza artificiale
- DevOps agenti assistenti con script runner
- Agenti complessi di calcolo, simulazione o ottimizzazione finanziaria
- Strumenti multimodali (ad esempio, combinando audio, documentazione e azione)

Guida all'implementazione

È possibile creare questo modello utilizzando quanto segue: Servizi AWS

- Amazon Bedrock (host dell'agente e inferenza LLM)
- AWS Lambda, Amazon ECS o Amazon SageMaker endpoint come runtime del server degli strumenti AWS Fargate
- Amazon API Gateway o AWS App Runner per esporre il server degli strumenti APIs

- Amazon EventBridge per la messaggistica disaccoppiata agent-to-tool
- AWS Step Functions o AWS AppFabric per comporre una logica multiagente sul server degli strumenti

Riepilogo

Gli agenti basati su strumenti che utilizzano server sono altamente modulari e scalabili. Dissociano la logica decisionale dall'esecuzione, il che consente all'agente principale di rimanere leggero mentre trasferisce le azioni complesse o sensibili su altri sistemi. Questo è importante per l'intelligenza artificiale agentica di livello aziendale, specialmente in ambienti che richiedono governance, osservabilità, isolamento, composizione dinamica o qualsiasi combinazione di questi elementi.

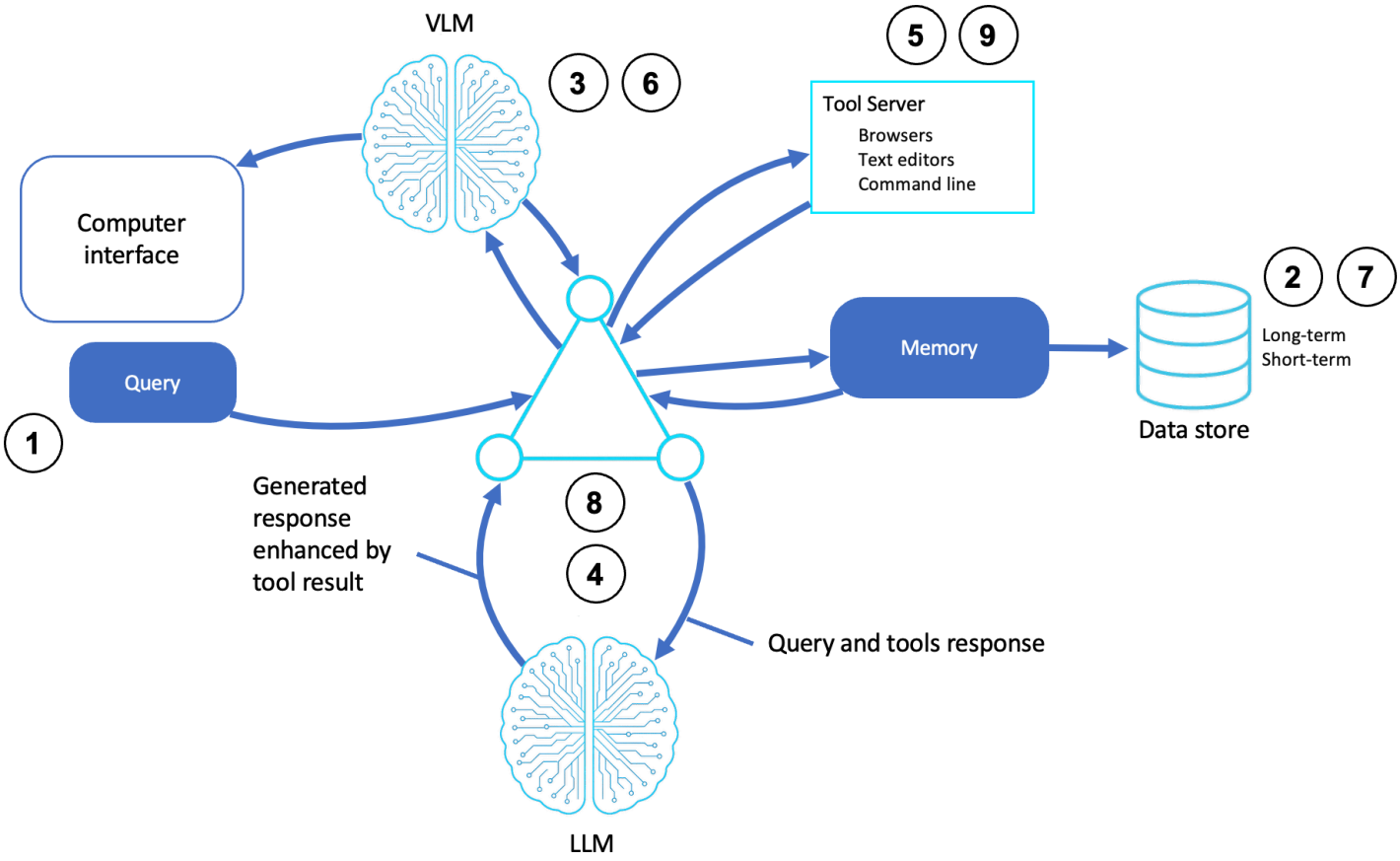
Agenti per uso informatico

Gli agenti che utilizzano computer possono simulare o controllare ambienti digitali come browser, terminali, file system e applicazioni. Questi agenti interpretano l'intento dell'utente, interagiscono con interfacce visive e testuali ed eseguono azioni mirate combinando ragionamenti LLM, modelli di linguaggio visivo (VLMs) e server di strumenti che eseguono comandi o simulano eventi di input.

Questo modello è importante per le automazioni di intelligenza artificiale pratiche, in cui l'agente non funziona solo come assistente ma anche come proxy che esegue azioni come farebbe un essere umano, spesso utilizzando gli stessi strumenti e ambienti.

Architecture

Nel diagramma seguente è illustrato un modello di agente utilizzato dal computer:



Description

1. Riceve un'interrogazione
   - Un'attività o una richiesta viene fornita tramite un'interfaccia utente, un'API o un'interfaccia in linguaggio naturale.
2. Accede alla memoria
   - L'agente recupera la memoria a breve e lungo termine per richiamare comandi, obiettivi e stati del sistema passati.
3. Analizza il contesto visivo
   - Un VLM osserva lo schermo del computer, lo stato del sistema o gli elementi dell'interfaccia utente per comprendere un determinato contesto e identificare gli elementi utilizzabili.
4. Motivi tramite un LLM
   - L'LLM combina la query, lo stato della memoria, lo strumento e la risposta del server per determinare l'azione successiva.
5. Interagisce con il server degli strumenti

- L'agente richiama strumenti ospitati su un server, che possono includere quanto segue:
  - Browser (ad esempio, Chrome headless) e ambienti shell
  - Editor di testo e codice
  - Interfacce di script personalizzate

6. Aggiorna gli input visivi
   - Se l'interfaccia utente del sistema cambia o sono necessarie ulteriori osservazioni, il VLM può rianalizzare lo stato dello schermo o i buffer di testo.
7. Aggiorna la memoria
   - Le nuove informazioni, gli stati del sistema o il feedback degli utenti vengono scritti nella memoria a breve e lungo termine.
8. Formula decisioni e spiegazioni finali
   - L'LLM sintetizza i risultati o consiglia azioni in base all'output dell'interrogazione e dello strumento.
9. Restituisce una risposta
   - L'agente restituisce i risultati all'interfaccia (ad esempio, un'attività completata, una conferma o un contenuto generato).

Funzionalità

- Ragionamento multimodale con input visivi e testuali
- Controllo delle applicazioni tramite input simulati o basati su API
- Gestione della memoria per uno stato persistente
- Autonomia nell'esecuzione della sequenza (flussi a più fasi)

Casi di utilizzo comune

- Sviluppatori di intelligenza artificiale che scrivono ed eseguono codice in IDEs
- Agenti per uso informatico per flussi di lavoro digitali ripetitivi
- Utenti simulati per il test del software e il controllo della qualità
- Agenti di accessibilità per la navigazione UIs tramite istruzioni vocali o di alto livello
- Automazione robotica intelligente dei processi (RPA) potenziata dal ragionamento

Guida all'implementazione

- È possibile creare questo modello utilizzando quanto segue: Servizi AWS
- Amazon Bedrock per la pianificazione e il ragionamento basati su LLM
- Amazon Elastic Compute Cloud (Amazon EC2) o SageMaker notebook Amazon per eseguire server di strumenti con ambienti di interfaccia utente simulati AWS Lambda
- Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) o Amazon DynamoDB per la persistenza della memoria
- Amazon Rekognition (o modelli personalizzati) per l'analisi delle immagini dell'interfaccia utente in scenari ibridi
- Amazon CloudWatch Logs o AWS X-Ray per l'osservabilità e gli audit trail

Riepilogo

Gli agenti che utilizzano i computer agiscono come operatori digitali autonomi, colmando il divario tra le interazioni uomo-computer e le azioni basate sull'intelligenza artificiale. Incorporando memoria e orchestrazione degli strumenti, questi agenti possono interagire in modo adattivo con sistemi progettati per gli esseri umani VLMs, eseguire azioni, aggiornare file, navigare nei menu e generare risposte.

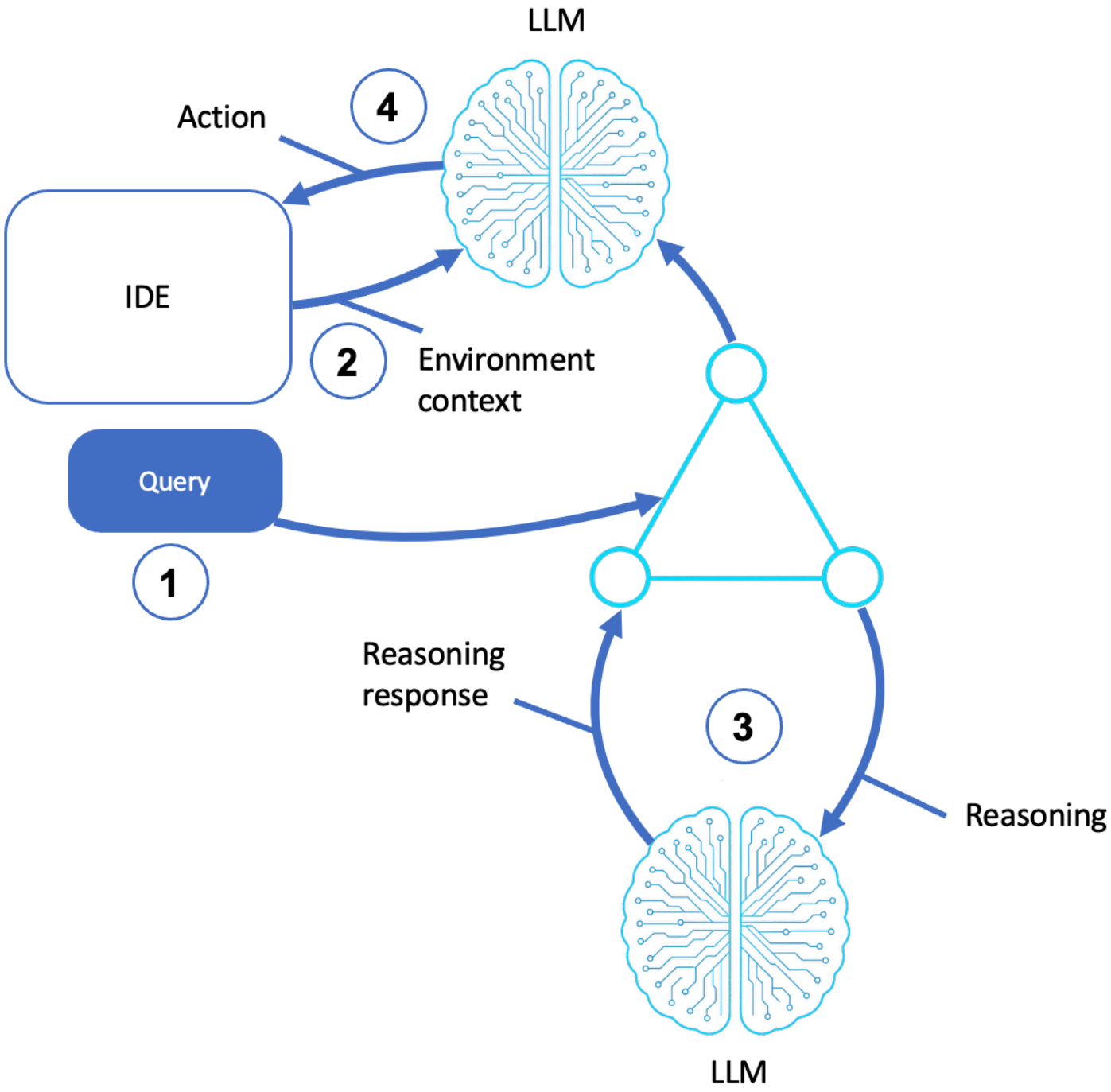
Agenti di codifica

Gli agenti di codifica possono ragionare sulle attività di programmazione, generare o modificare codice e interagire con ambienti di sviluppo, come e. IDEs CLIs Questi agenti combinano la comprensione del linguaggio naturale con il ragionamento strutturato per assistere, potenziare e automatizzare lo sviluppo del software, dalla generazione di funzioni alla correzione di bug e alla creazione di test.

A differenza degli strumenti di completamento automatico, gli agenti di codifica interpretano attivamente gli obiettivi degli utenti, interrogano il contesto nell'ambiente di sviluppo (ad esempio, apre i file e rileva gli errori), identificano i requisiti e quindi propongono ed eseguono azioni.

Architecture

Un modello di agente di codifica è illustrato nel diagramma seguente:



Description

1. Riceve una richiesta

- L'utente fornisce istruzioni in linguaggio naturale tramite una palette di comandi, una finestra di chat o una CLI (ad esempio, «Aggiungi registrazione a questa funzione» o «Refactor for readability»).

2. Estrae il contesto dell'ambiente
   - L'agente raccoglie il contesto dall'IDE, inclusi file attivi, posizione del cursore, frammenti di codice e tabelle dei simboli.
   - Emette messaggi di errore, risultati dei test e output di altri agenti.
3. Ragionamento LLM
   - L'agente invia un prompt, incluso l'interrogazione e il contesto ambientale, a un LLM.
     - L'LLM esegue un ragionamento per determinare quanto segue:
     - Cosa deve cambiare
     - Come generare una soluzione
     - Qualsiasi fase di rifattorizzazione, riscrittura o codifica
4. Esegue azioni
   - L'LLM restituisce l'output all'agente e lo importa nell'ambiente IDE o di runtime.
   - Ciò può includere l'inserimento o la modifica di codice, la generazione di commenti o documentazione e l'attivazione di attività successive di compilazione, test e linting.

Funzionalità

- Elevata consapevolezza contestuale (ad esempio, stato dell'IDE, cursore e albero della sintassi)
- Ragionamento iterativo degli obiettivi e del feedback
- Pianificazione opzionale del codice e separazione delle azioni (ad esempio, prima ragionare e poi agire)
- Funziona in flussi di lavoro di sviluppo sincroni o asincroni

Casi di utilizzo comune

- Generazione di codice dalle descrizioni delle attività
- Rifattorizzazione e ottimizzazione del codice
- Generazione e convalida di test case
- Spiegazioni e debug degli errori

- Assistenti alla documentazione
- Copiloti di programmazione accoppiati

Guida all'implementazione

- È possibile creare questo modello utilizzando i seguenti strumenti e: Servizi AWS
- Amazon Bedrock per la generazione e il ragionamento basati su LLM
- Amazon Q Developer per suggerimenti e completamenti di codifica
- AWS Lambda o Amazon Elastic Container Service (Amazon ECS) per l'esecuzione e il test di ambienti sandbox
- AWS Cloud9, estensioni VS Code o integrazioni IDE personalizzate per ospitare e valutare il contesto
- Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) per l'archiviazione di richieste, risposte e cronologia delle revisioni intermedie

Riepilogo

Gli agenti di codifica sono nuovi strumenti di sviluppo basati sull'intelligenza artificiale in grado di interpretare il linguaggio naturale, analizzare il contesto, generare modifiche al codice in più fasi e integrarsi con il ciclo di vita dello sviluppo del software.

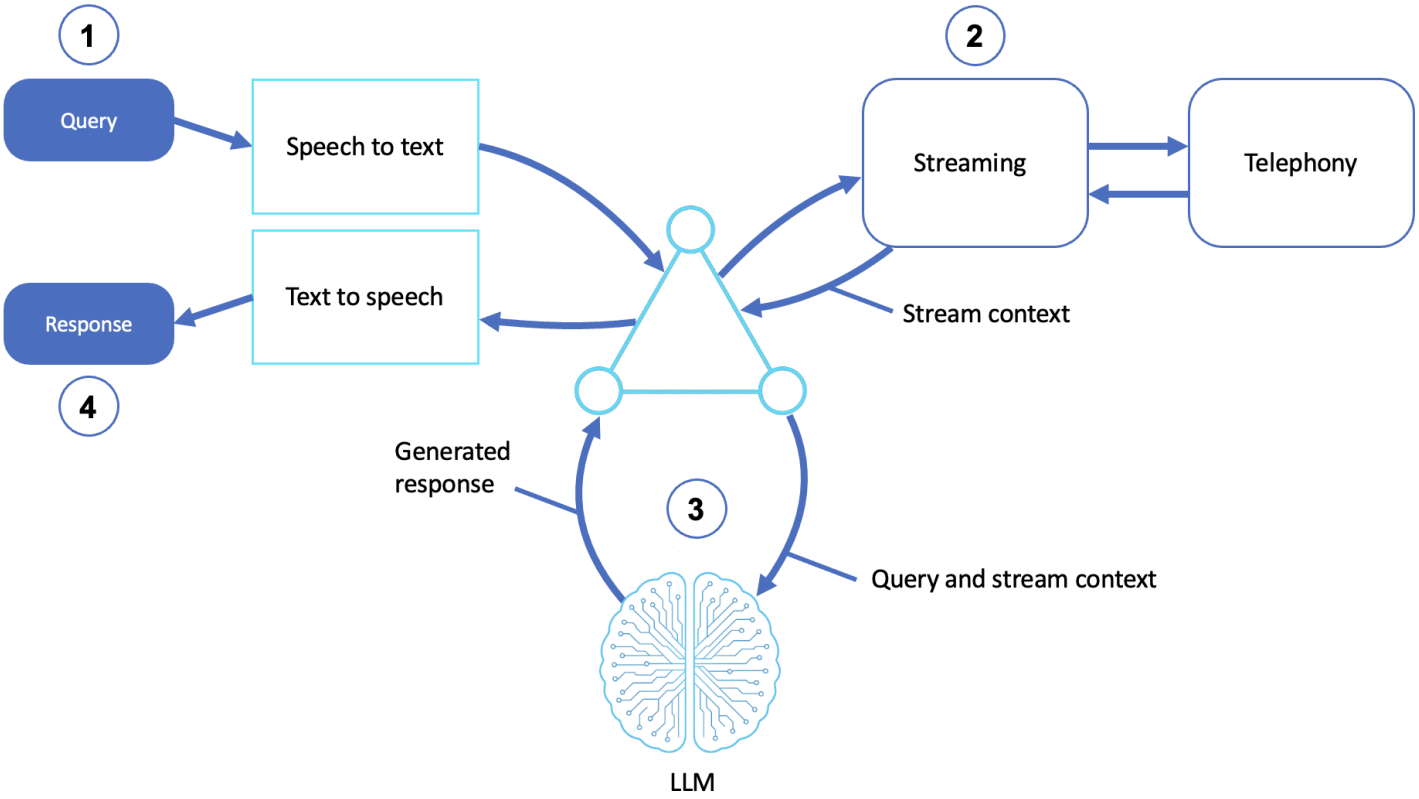
Agenti vocali e vocali

Gli agenti vocali e vocali interagiscono con gli utenti attraverso il dialogo vocale. Questi agenti integrano il riconoscimento vocale, la comprensione del linguaggio naturale e la sintesi vocale per consentire l'intelligenza artificiale conversazionale su telefonia, dispositivi mobili, web e piattaforme integrate.

Gli agenti vocali sono particolarmente efficaci in ambienti a mani libere, in tempo reale o basati sull'accessibilità. Combinando interfacce di streaming con ragionamenti basati su LLM, facilitano interazioni ricche e dinamiche che risultano naturali per gli utenti.

Architecture

Un agente vocale e vocale è illustrato nel diagramma seguente:



Description

1. Riceve una richiesta vocale
   - L'utente invia una richiesta a un telefono, un microfono o un sistema integrato.
   - Un modulo speech-to-text (STT) converte l'audio in testo.
2. Integra il contesto di streaming e telefonia
   - L'agente utilizza un'interfaccia di streaming per gestire l'audio I/O in tempo reale.
   - Se viene implementata in un contact center o in un contesto di telecomunicazioni, l'integrazione della telefonia gestisce il routing delle sessioni, l'input multifrequenza a due toni (DTMF) e il trasporto multimediale.

Nota: DTMF si riferisce ai toni generati quando si premono i pulsanti sulla tastiera di un telefono. Nel contesto dell'integrazione del contesto di streaming e telefonia negli agenti vocali, il DTMF viene utilizzato come meccanismo di immissione del segnale durante una telefonata, in particolare nei sistemi di risposta vocale interattiva (IVR). Gli ingressi DTMF consentono all'agente di:

- Riconoscere le selezioni del menu (ad esempio, «Premi 1» per la fatturazione). Premi 2 per ricevere assistenza.»)
- Raccogli input numerici (ad esempio numeri di conto e numeri di conferma) PINs
- Attiva flussi di lavoro o transizioni di stato nei flussi di chiamata
- Passa dalla voce al tono tattile quando necessario

1. Motivi tramite il contesto del flusso LLM
   - La query viene inviata all'agente, che la trasmette, insieme a tutti i metadati della sessione (ad esempio, ID chiamante, contesto precedente), a un LLM.
   - L'LLM genera una risposta, possibilmente utilizzando una chain-of-thought strategia o una memoria a più turni se l'interazione è in corso.
2. Restituisce una risposta vocale
   - L'agente converte la sua risposta in voce utilizzando text-to-speech (TTS).
   - Restituisce l'audio all'utente tramite un canale vocale.

Funzionalità

- Comprensione e generazione del parlato in tempo reale
- Multilingue I/O con supporto STT e TTS
- Integrazione con telefonia o streaming APIs
- Consapevolezza della sessione e trasferimento della memoria tra un turno e l'altro

Casi di utilizzo comune

- Sistemi IVR conversazionali
- Addetti alla reception e programmatori di appuntamenti virtuali
- Agenti dell'helpdesk con comandi vocali
- Assistenti vocali indossabili
- Interfacce vocali per case intelligenti e strumenti di accessibilità

Guida all'implementazione

È possibile creare questo modello utilizzando i seguenti strumenti e: Servizi AWS

- Amazon Lex V2 o Amazon Transcribe per STT
- Amazon Polly per TTS
- Amazon Chime SDK, Amazon Connect o Amazon Interactive Video Service (Amazon IVS) per streaming e telefonia
- Amazon Bedrock per ragionare con Anthropic o altri AI21 modelli di base
- AWS Lambda per connettere STT, LLM, TTS e il contesto della sessione

(Facoltativo) I miglioramenti aggiuntivi possono includere quanto segue:

- Amazon Kendra OpenSearch o per RAG sensibile al contesto
- Amazon DynamoDB per la memoria di sessione
- Amazon CloudWatch Logs e AWS X-Ray per la tracciabilità

Riepilogo

Gli agenti vocali e vocali sono sistemi intelligenti che interagiscono attraverso conversazioni naturali. Integrando le interfacce vocali con il ragionamento LLM e l'infrastruttura di streaming in tempo reale, gli agenti vocali consentono interazioni fluide, accessibili e scalabili.

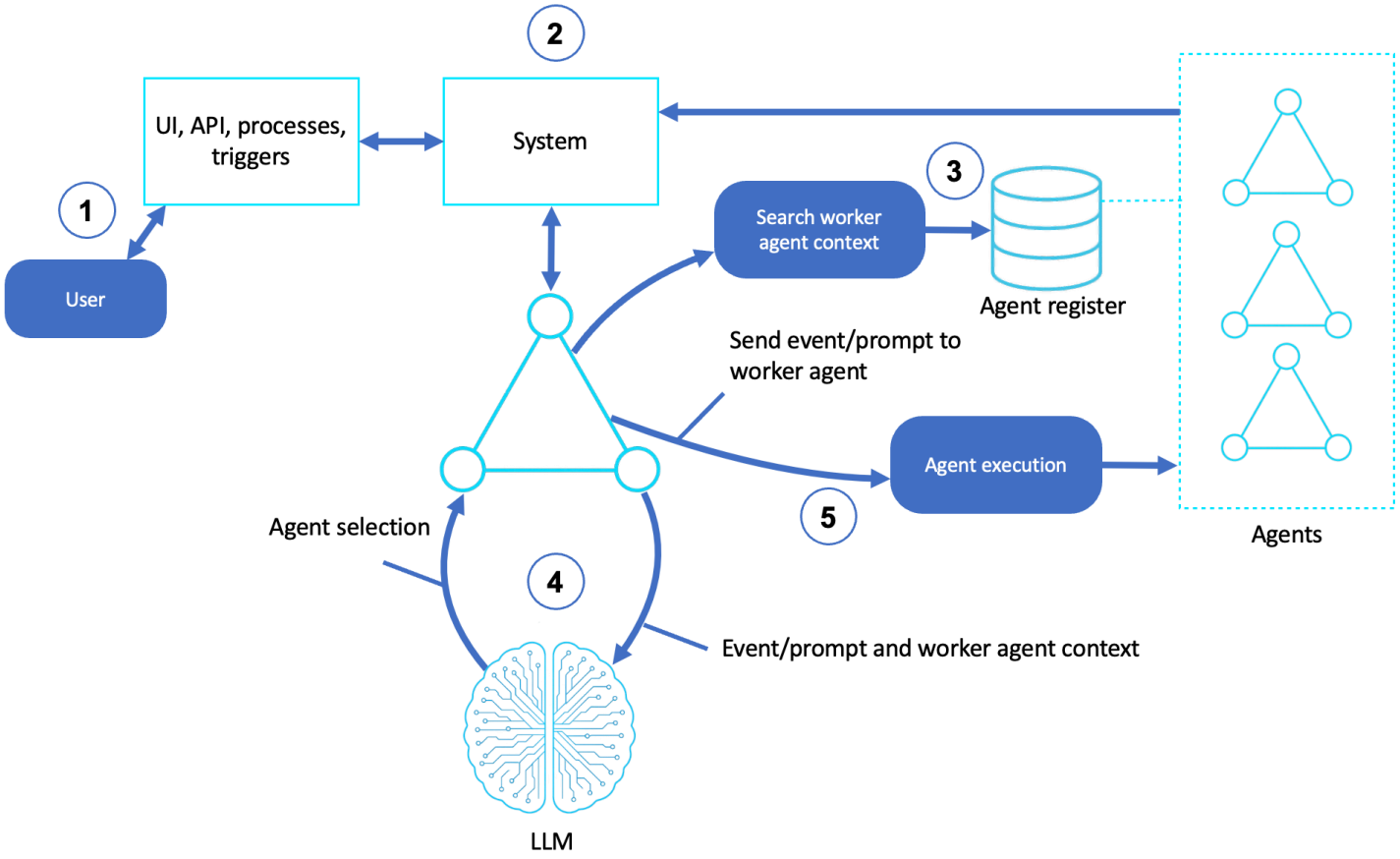
Agenti di orchestrazione del flusso di lavoro

Gli agenti di orchestrazione del flusso di lavoro gestiscono e coordinano attività, processi e servizi in più fasi su sistemi distribuiti. Piuttosto che ragionare e agire isolatamente, questi agenti delegano il lavoro a subagenti o altri sistemi, mantengono il contesto di esecuzione e si adattano in base a risultati intermedi.

Questi agenti sono una parte fondamentale dei flussi di automazione. Sono particolarmente utili quando si gestiscono attività di lunga durata, composizioni multiagente e integrazioni tra domini in cui è necessario chiamare vari agenti e strumenti in sequenza o in modo condizionale.

Architecture

Un agente di orchestrazione del flusso di lavoro è illustrato nel diagramma seguente:



Description

1. Riceve l'input dell'utente
   - Un utente (o un trigger esterno) avvia un'attività tramite un'interfaccia utente, un'API o un evento di sistema.
2. Gestisce gli eventi di sistema
   - Un componente di sistema riceve la richiesta ed emette un evento o un comando che richiede l'orchestrazione.
3. Recupera il contesto
   - L'agente di workflow interroga le knowledge base e i registri degli agenti per trovare l'agente di lavoro giusto per l'attività in base a metadati, dominio e percentuale di successo precedente.
4. Seleziona un agente LLM

- Un LLM aiuta a selezionare l'agente o il piano di flusso di lavoro migliore analizzando la descrizione delle attività e le opzioni disponibili.
- Può anche formulare istruzioni specifiche per l'attività da inviare a un agente selezionato.

5. Delega ed esegue
   - L'agente di lavoro scelto riceve l'evento o il prompt e inizia a eseguire i comandi.
   - Può tenere traccia dello stato di esecuzione, riprovare in caso di errore e passare i risultati intermedi all'agente successivo della sequenza.

Funzionalità

- Composizione degli agenti (ad esempio supervisori, agenti collaboratori e strumenti)
- Esecuzione pianificata o basata sugli eventi
- Monitoraggio della memoria e dello stato nel tempo
- Orchestrazione gerarchica o parallela delle attività (sincrona rispetto ai flussi di lavoro asincroni)
- Selezione e concatenamento dinamici degli agenti

Casi di utilizzo comune

- Automazione in più fasi (ad esempio, inserimento e creazione di report di dati)
- Instradamento e intensificazione dell'assistenza clienti (ad esempio,) agent-as-coordinator
- Gli agenti di intelligenza artificiale si coordinano con umani e bot all'interno dello stesso ciclo
- Automatizza i processi aziendali utilizzando la logica basata su LLM
- I sistemi ibridi combinano agenti di intelligenza artificiale e strumenti di orchestrazione tradizionali

Guida all'implementazione

È possibile creare questo modello utilizzando i seguenti strumenti e: Servizi AWS

- Amazon Bedrock per il ragionamento e la selezione degli agenti
- AWS Step Functions o Amazon EventBridge per la composizione del flusso di lavoro
- AWS Lambda come unità di esecuzione o task runner

- Amazon DynamoDB, Amazon Simple Storage Service (Amazon S3) o Amazon RDS per tenere traccia di stati e risultati
- AWS AppFabric o Amazon AppFlow per il coordinamento tra sistemi
- (Facoltativo) Usa Amazon SageMaker run agent per ospitare worker agent specifici del dominio

Riepilogo

Gli agenti del flusso di lavoro coordinano, adattano e allineano gli obiettivi in ambienti con più agenti. Ciò significa che gli agenti di intelligenza artificiale possono collaborare, adattarsi alle condizioni di runtime e fornire risultati complessi attraverso flussi di lavoro modulari e spiegabili.

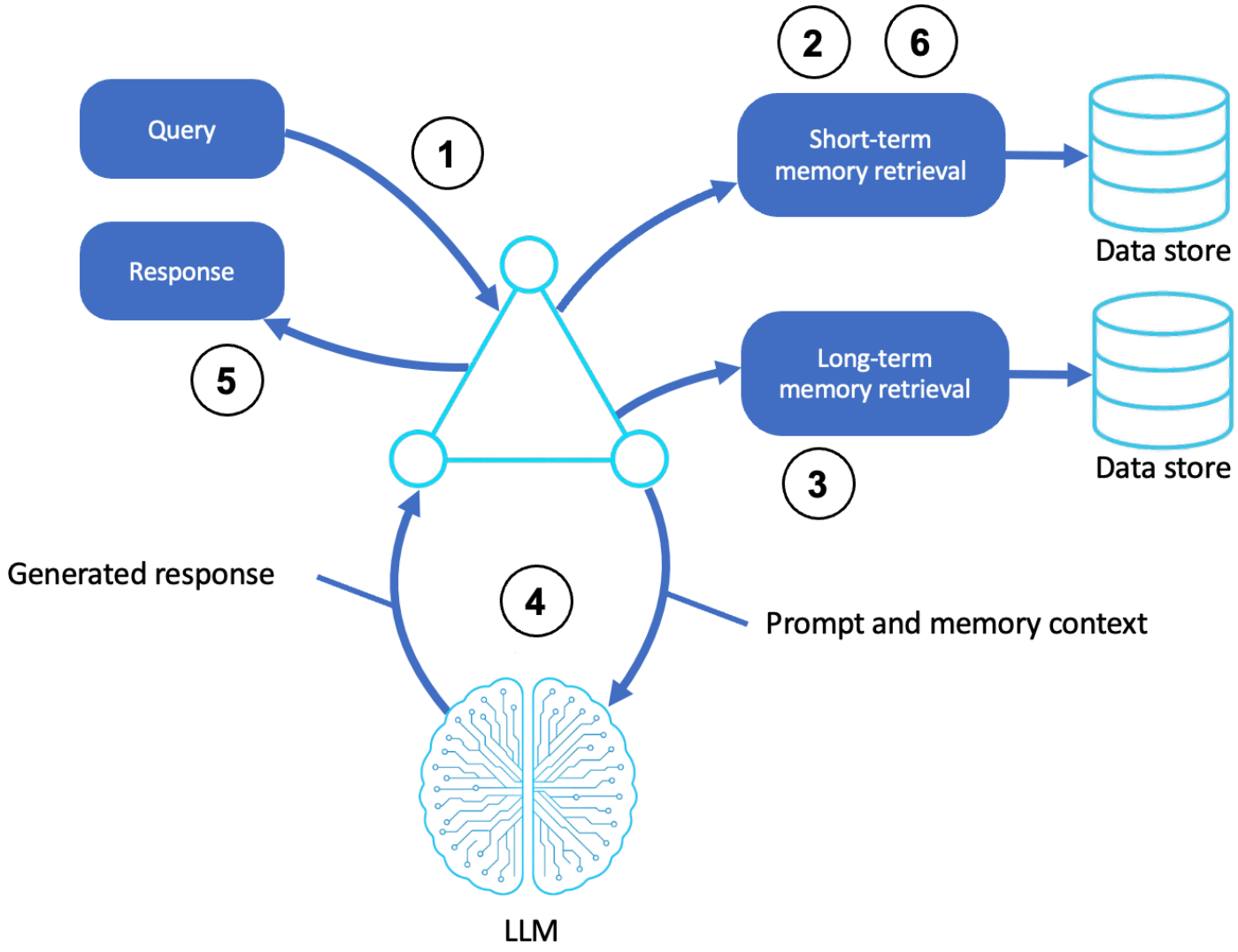
Agenti con memoria aumentata

Gli agenti con memoria aumentata sono potenziati dalla capacità di archiviare, recuperare e ragionare utilizzando la memoria a breve e lungo termine. Ciò consente loro di mantenere il contesto tra più attività, sessioni e interazioni, il che produce risposte più coerenti, personalizzate e strategiche.

A differenza degli agenti stateless, gli agenti con memoria aumentata si adattano facendo riferimento a dati storici, imparano dai risultati precedenti e prendono decisioni in linea con gli obiettivi, le preferenze e l'ambiente dell'utente.

Architecture

Un agente con memoria aumentata è illustrato nel diagramma seguente:



Description

1. Riceve input o eventi
   - L'agente riceve una richiesta dell'utente o un evento di sistema. Può trattarsi di un testo, di un trigger dell'API o di una modifica ambientale.
2. Recupera la memoria a breve termine
   - L'agente recupera la cronologia delle conversazioni recenti, il contesto delle attività o lo stato del sistema pertinenti alla sessione o al flusso di lavoro.
3. Recupera la memoria a lungo termine
   - L'agente interroga la memoria a lungo termine (ad esempio, database vettoriali e archivi chiave-valore) per informazioni storiche, come le seguenti:

    - Preferenze dell'utente
    - Decisioni e risultati passati
    - Concetti, riassunti o esperienze appresi
4. Motivi tramite il LLM
    - Il contesto di memoria è incorporato nel prompt LLM e consente all'agente di ragionare in base sia agli input correnti che alle conoscenze precedenti.
5. Genera uscite
    - L'agente produce una risposta, un piano o un'azione contestualmente consapevoli e personalizzati in base alla cronologia delle attività e agli input dell'utente.
6. Aggiorna la memoria
    - Nuove informazioni, come obiettivi aggiornati, segnali di successo e fallimento e risposte strutturate, vengono archiviate per attività future.

Funzionalità

- Continuità della sessione tra conversazioni o eventi
- Persistenza degli obiettivi nel tempo
- Consapevolezza contestuale basata su uno stato in evoluzione
- Adattabilità basata su successi e fallimenti precedenti
- Personalizzazione in linea con le preferenze e la cronologia dell'utente

Casi di utilizzo comune

- Copiloti conversazionali che ricordano le preferenze dell'utente
- Agenti di codifica che tengono traccia delle modifiche alla codebase
- Agenti del flusso di lavoro che si adattano in base alla cronologia delle attività
- Gemelli digitali che si evolvono a partire dalla conoscenza del sistema
- Agenti di ricerca che evitano recuperi ridondanti

Implementazione di agenti con memoria aumentata

Utilizza i seguenti strumenti e Servizi AWS per gli agenti con memoria aumentata:

Livello di memoria	Servizio AWS	Scopo
A breve termine	Contesto Amazon DynamoDB, Redis, Amazon Bedrock	Recupero rapido degli stati di interazione recenti
A lungo termine (strutturato)	Amazon Aurora, Amazon DynamoDB, Amazon Neptune	Fatti, relazioni e registri
A lungo termine (semantico)	OpenSearch, PostgreSQL, Pinecone	Recupero basato sull'inco rporamento (ovvero RAG)
Storage	Simple Storage Service (Amazon S3)	Archiviazione di trascrizioni, memorie strutturate e file
Orchestrazione	AWS Lambda oppure AWS Step Functions	Gestione dell'iniezione di memoria e del ciclo di vita degli aggiornamenti
Ragionamento	Amazon Bedrock	Claude o Mistral antropico con richieste di memoria

Implementazione dei prompt iniettati dalla memoria

Per integrare la memoria nel ragionamento degli agenti, utilizzate una combinazione di stati strutturati e iniezione di contesto potenziata dal recupero:

- Includi lo stato più recente dell'agente e la cronologia recente dei dialoghi come input strutturato durante la creazione del prompt per il modello linguistico, in modo che possa ragionare con il contesto completo.
- Usa la generazione aumentata di recupero (RAG) per estrarre documenti o fatti pertinenti dalla memoria a lungo termine.
- Riassumi i piani, il contesto e le interazioni precedenti per la compressione e la pertinenza.
- Inserisci moduli di memoria esterni, come archivi vettoriali o log strutturati, durante l'inferenza per guidare il processo decisionale.

Riepilogo

Gli agenti con memoria aumentata mantengono la continuità del pensiero imparando dall'esperienza e ricordando il contesto dell'utente. Questi agenti superano l'intelligenza reattiva utilizzando la collaborazione a lungo termine, la personalizzazione e il ragionamento strategico. In termini di intelligenza artificiale agentica, la memoria consente agli agenti di comportarsi più come controparti digitali adattive e meno come strumenti stateless.

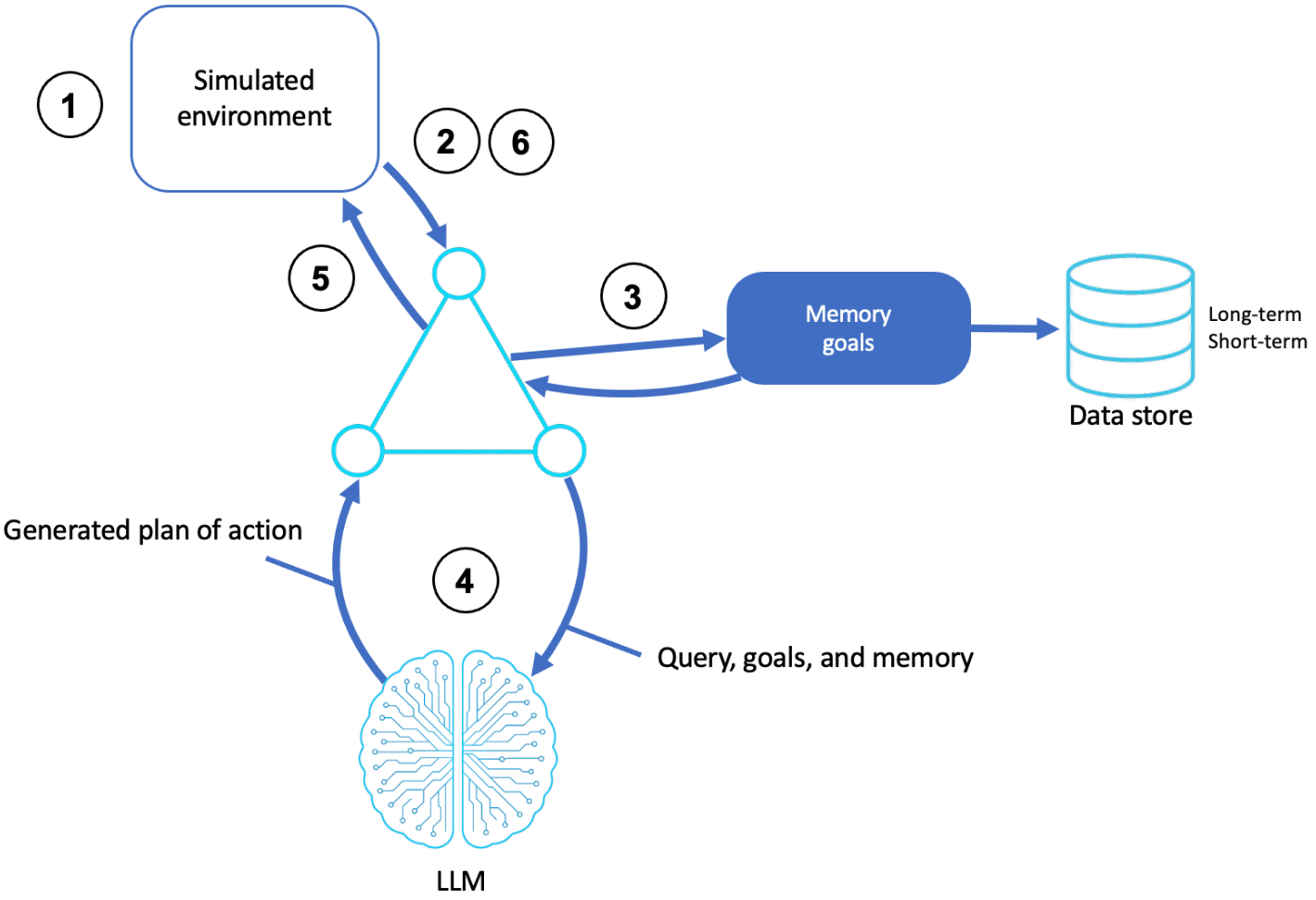
Agenti di simulazione e test

Gli agenti di simulazione e di test operano all'interno di ambienti virtualizzati o controllati in cui ragionano, agiscono e imparano. Questi agenti simulano il comportamento, modellano i risultati e addestrano le strategie in contesti ripetibili prima di applicarli agli ambienti del mondo reale.

Questo modello è utile per lo sviluppo iterativo, l'apprendimento per rinforzo (RL), la valutazione decisionale autonoma e i test comportamentali emergenti. Gli agenti di simulazione spesso operano a circuito chiuso, ricevono feedback dall'ambiente e adattano il loro comportamento di conseguenza, il che li rende fondamentali per attività che coinvolgono il ragionamento spaziale, il controllo in tempo reale o la dinamica di sistemi complessi.

Architecture

Il diagramma seguente mostra un agente di simulazione o di test:



Description

1. Avvia un ambiente
   - L'agente avvia un ambiente simulato (ad esempio, un mondo 3D, un motore fisico, una sandbox CLI o un flusso di dati sintetici).
   - L'agente viene caricato nell'ambiente con un'attività, un obiettivo o una politica iniziali.
2. Percepisce l'agente
   - L'agente percepisce lo stato attuale tramite telemetria di simulazione (ad esempio, emulazione di sensori, fotocamera virtuale e registri strutturati).
3. Recupera obiettivo e memoria
   - L'agente recupera l'obiettivo assegnato, le istruzioni sullo scenario o l'obiettivo contestuale.
   - Può anche recuperare la memoria precedente, inclusa la seguente:
     - Strategie o politiche a lungo termine

    - Mappe ambientali o vincoli noti
    - Successi o fallimenti passati derivanti da simulazioni simili
4. Motivi e piani
    - Un LLM interpreta lo stato simulato, gli obiettivi del compito e le conoscenze apprese.
    - Genera un piano d'azione o un comando di controllo.
5. Esegue azioni simulate
    - L'agente esegue il piano, modifica lo stato, naviga nello spazio o interagisce con entità virtuali.
6. Impara
    - L'agente valuta i risultati delle azioni
    - A seconda della configurazione dell'agente, può eseguire le seguenti operazioni:
        - Esegui RL
        - Registra i risultati per future ottimizzazioni
        - Adatta le strategie in tempo reale

Funzionalità

- Funziona all'interno di ambienti sintetici o virtuali
- Supporta trial-and-error l'apprendimento, il perfezionamento delle politiche e la modellazione del sistema
- Test a basso rischio per il comportamento, la gestione dei guasti e i casi limite
- Consente l'analisi del comportamento degli agenti emergenti in configurazioni con più agenti
- Supporta sia il controllo che l'esplorazione a circuito chiuso human-in-the-loop

Casi di utilizzo comune

- Apprendimento rinforzato per robotica, droni e giochi
- Addestramento con veicoli autonomi su strade virtuali
- Scenari simulati UIs o CLIs destinati DevOps a un banco di prova
- Esperimenti comportamentali emergenti nelle simulazioni sociali
- Convalida di sicurezza della logica decisionale prima della produzione

Guida all'implementazione

È possibile creare un agente di simulazione e test bed utilizzando i seguenti strumenti e: Servizi AWS

Componente	Servizio AWS	Scopo
Ambiente	Amazon ECS EC2, Amazon o un simulatore personalizzato in Amazon SageMaker Studio Lab	Esegui mondi virtuali (Gazebo, Unity, Unreal) o sandbox CLIs
Logica dell'agente	Amazon Bedrock SageMaker, Amazon o AWS Lambda	Pianificatori o agenti RL basati su LLM
Circuito di feedback	Amazon SageMaker Reinforcement Learning CloudWatch, Amazon o log personalizzati	Monitoraggio delle ricompense, punteggio dei risultati e registrazione del comportamento
Memoria e replay	Amazon S3, Amazon DynamoDB o Amazon RDS	Dati persistenti sullo stato, sulla cronologia degli episodi o sullo scenario
Visualizzazione	CloudWatch Pannelli di controllo Amazon o notebook Amazon SageMaker	Osserva le modifiche alle politiche, i risultati e le metriche di formazione

Le seguenti sono applicazioni aggiuntive:

- AWS SimSpace Weaverper simulazioni spaziali su larga scala
- AWS IoT Coreper testare dispositivi ombra
- Amazon SageMaker Experiments per la valutazione e il benchmarking degli agenti

Riepilogo

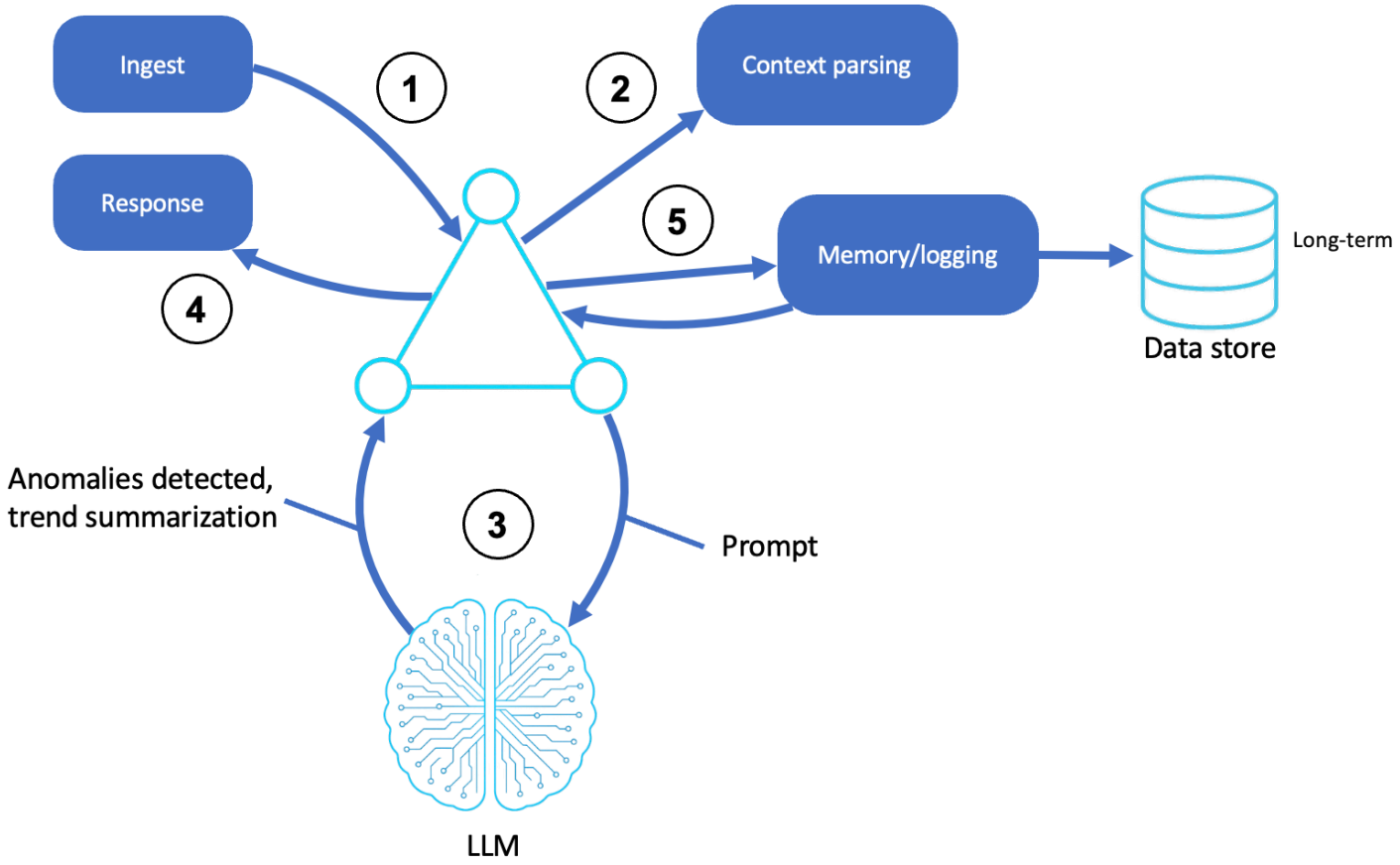
Gli agenti di simulazione e di test servono per l'esplorazione strutturata prima di essere implementati nei sistemi di produzione. Utilizzate questi agenti per addestrare politiche di navigazione autonome, testare i processi aziendali in ambienti sintetici e valutare gli sciami per i modelli di coordinamento.

Agenti osservatori e di monitoraggio

Gli osservatori e gli agenti di monitoraggio osservano passivamente i sistemi, gli ambienti e le interazioni per rilevare modelli, generare approfondimenti e innescare azioni. In qualità di osservatori intelligenti, migliorano gli avvisi, la diagnostica e gli audit senza avviare direttamente il comportamento.

Questi agenti eccellono laddove il monitoraggio tradizionale manca di adattabilità o ragionamento, in particolare per il AI-in-the-loop monitoraggio, il rilevamento delle anomalie, la supervisione della conformità e l'intelligence sulla sicurezza. Gli agenti osservatori sono ascoltatori di eventi che monitorano continuamente la telemetria del sistema e le interazioni con gli utenti. L'agente dipende dalla percezione, dall'interpretazione e dall'escalation o dalla segnalazione condizionale.

Architecture



Description

1. Acquisisci telemetria
   - L'agente riceve input da una o più fonti di sistema, come le seguenti:
     - Registri (applicazione, infrastruttura, sicurezza)
     - Metriche (prestazioni, latenza, utilizzo)
     - Eventi (chiamate API, azioni dell'utente, dati dei sensori)
2. Analizza il contesto
   - L'input non elaborato viene analizzato, strutturato e arricchito con metadati, ad esempio timestamp, identità dell'attore, stato del sistema e ID di traccia.
3. Motivi per utilizzare LLM
   - L'agente utilizza un LLM o un modulo logico per interpretare gli input analizzati identificando le anomalie, riepilogando le tendenze e correlandoli tra tracce distribuite o finestre temporali.

4. Classificare o avvisare
   - L'agente determina se il comportamento osservato giustifica quanto segue:
     - Un avviso o un'escalation
     - Un rapporto o un aggiornamento del pannello di controllo
     - Un trigger di risposta (ad esempio, riparazione automatica e applicazione delle politiche)
5. Registra la memoria o i loop di feedback
   - L'agente archivia eventi e decisioni per l'apprendimento a lungo termine, gli audit o riferimenti futuri per altri agenti.

Funzionalità

- Passivo e non invasivo (l'agente non agisce direttamente)
- Altamente scalabile e asincrono
- Correlazione basata sull'intelligenza artificiale tra segnali rumorosi o distribuiti
- Supporta audit, conformità e informazioni in tempo reale
- Può alimentare agenti a valle o flussi di lavoro umani

Casi di utilizzo comune

- Osservabilità potenziata dall'intelligenza artificiale per microservizi e APIs
- Monitoraggio della deriva del modello, della violazione delle politiche o del comportamento out-of-band
- Analisi delle attività o riepiloghi delle interazioni con i clienti
- Agenti di revisione del codice che monitorano gli impegni o le implementazioni
- Monitoraggio dei registri di sicurezza o conformità mediante il ragionamento LLM

Guida all'implementazione

È possibile creare un osservatore e un agente di monitoraggio utilizzando i seguenti strumenti e: Servizi AWS

Componente	Servizio AWS	Scopo

Ingestione di eventi	Amazon EventBridge, Amazon CloudWatch Logs, Amazon Kinesis, Amazon S3	Inserisci telemetria strutturata e non strutturata
Pre-elaborazione	AWS Lambda, AWS Glue, AWS Step Functions	Trasforma i dati grezzi in prompt strutturati
Motore di ragionamento	Amazon Bedrock, Amazon SageMaker, AWS Lambda	Analizza gli eventi, classific a i comportamenti, genera approfondimenti
Archiviazione e memoria	Amazon S3, Amazon DynamoDB, OpenSearch	Osservazioni, riepiloghi e risultati persistenti
Avvisi e intensificazione	Amazon SNS, AWS AppFabric Amazon EventBridge	Attiva sistemi o agenti downstream

Le seguenti sono applicazioni aggiuntive:

- AWS Security Hub CSPMper il monitoraggio dei registri di sicurezza
- Amazon Quick Suite per la visualizzazione degli output degli agenti

Riepilogo

Gli osservatori e gli agenti di monitoraggio tengono traccia dei sistemi e dei comportamenti in tempo reale. Rilevano anomalie, controllano la sicurezza e raccolgono informazioni operative identificando modelli che gli umani o le regole potrebbero trascurare. Questa funzionalità consente di creare sistemi in grado di adattarsi alle mutevoli condizioni e di prendere decisioni basate su un'analisi completa dei dati.

Collaborazione tra più agenti

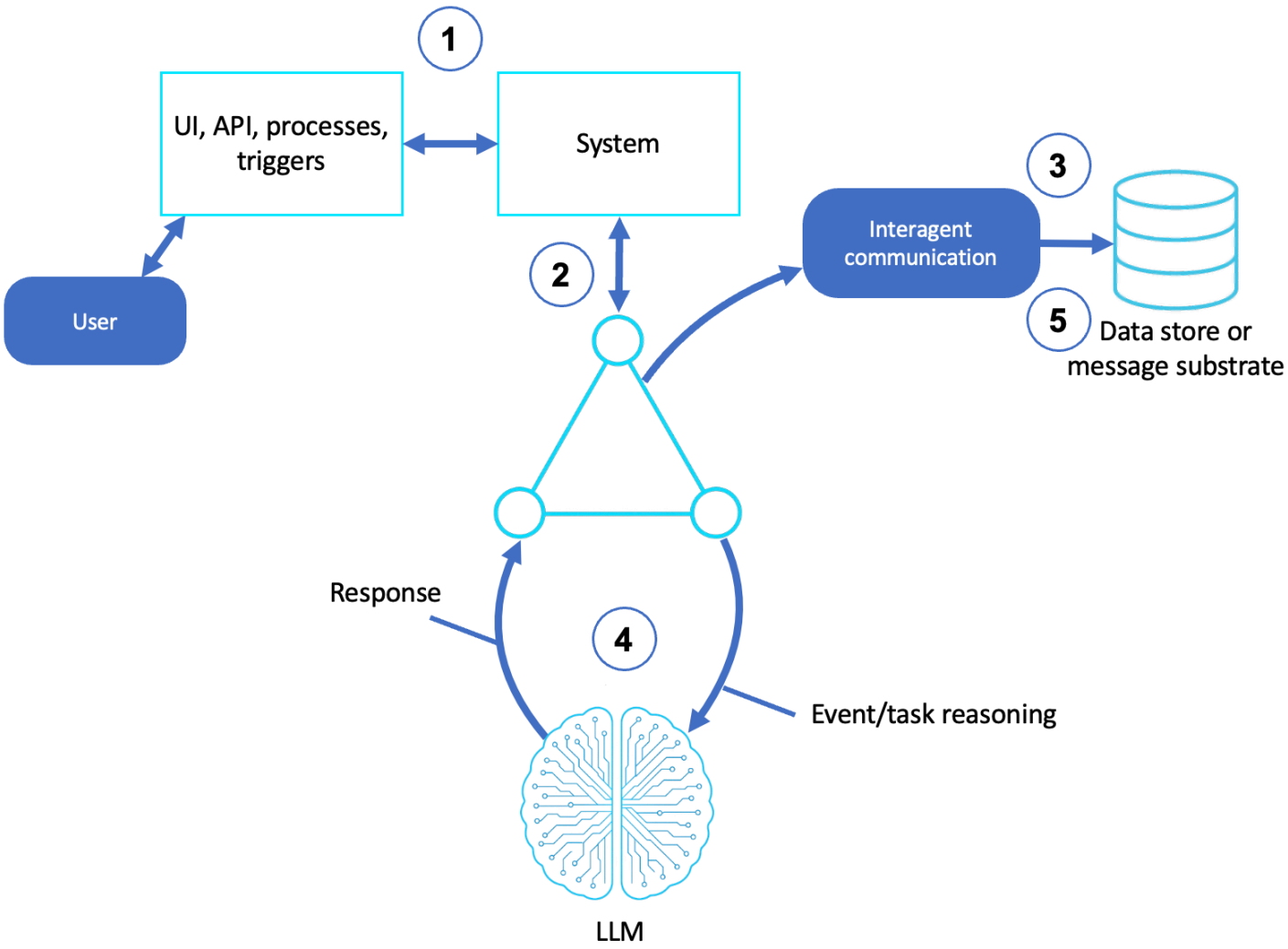
La collaborazione tra più agenti si riferisce a un modello in cui più agenti autonomi, ciascuno con un ruolo, una specializzazione o un obiettivo distinti, negoziano per risolvere compiti complessi. Questi agenti possono operare indipendentemente o con altri agenti condividendo informazioni, dividendo le responsabilità e ragionando collettivamente verso un obiettivo.

Questo modello è diverso dagli agenti del flusso di lavoro, che coordinano e delegano centralmente le attività agli agenti subordinati in un flusso strutturato. Al contrario, la collaborazione tra più agenti enfatizza la nostra coordinazione peer-to-peer emergente abilitando l'adattabilità, il parallelismo e la divisione della cognizione. La tabella seguente mette a confronto la collaborazione tra più agenti e gli agenti del flusso di lavoro:

Funzionalità	Agenti di workflow	Scopo
Controllo	Coordinatore centralizzato	Peer decentralizzati, distribuiti o basati sui ruoli
Interazione	Un agente delega e monitora l'esecuzione	Più agenti negoziano, condividono e si adattano
Progettazione	Sequenza predefinita di attività	Distribuzione delle attività emergente e flessibile
Coordinamento	Orchestrazione procedurale	Interazioni cooperative o competitive
Casi d’uso	Automazione dei processi aziendali	Ragionamento complesso , esplorazione e strategie emergenti

Architecture

Il diagramma seguente mostra la collaborazione tra più agenti:



Description

1. Avvia un'attività
   - Un utente o un sistema emette un obiettivo o un problema di alto livello.
   - Un agente «manager» o un contesto iniziale definisce l'obiettivo.
2. Assegna o scopre i ruoli
   - Gli agenti si autoassegnano (logica o ragionamento simbolico) o sono delegati (mediatori di eventi) ad altri ruoli, come pianificatore, ricercatore, esecutore, critico o spiegatore.
3. Comunica con altri agenti
   - Gli agenti comunicano tramite memoria condivisa, code di messaggistica o concatenamento di prompt.

- Possono discutere, interrogarsi o proporsi attività secondarie l'un l'altro.

4. Utilizza un ragionamento specializzato
   - Ogni agente utilizza il proprio modello o la propria logica di dominio per risolvere una parte del problema.
   - Gli agenti possono utilizzare LLMs prompt e memoria specifici per ruolo.
5. Coordina i risultati o gli obiettivi
   - Gli agenti sintetizzano i contributi in una risposta, un piano o un'azione finale.
   - (Facoltativo) Un agente supervisore può convalidare o riassumere l'output sintetizzato.

Funzionalità

- Agenti di livello paritario con ruoli o competenze specializzati
- Comportamento emergente attraverso la comunicazione o la negoziazione
- Elaborazione parallela di problemi complessi o sfaccettati
- Supporta la deliberazione, l'autocorrezione e l'iterazione riflessiva
- Modella le dinamiche sociali, la collaborazione scientifica o i ruoli dei team aziendali

Casi di utilizzo comune

- Team di ricerca autonomi (agente di ricerca, riepilogo e validatore)
- Sviluppo di software (pianificatore, programmatore e tester)
- Modellazione di scenari aziendali (finanze, politiche e conformità)
- Negoziazione, offerta o ragionamento multipartitico
- Attività multimodali (immagine, testo e logica)

Guida all'implementazione

È possibile creare un sistema multiagente utilizzando i seguenti strumenti e: Servizi AWS

Componente	Servizio AWS	Scopo

Hosting per agenti	Amazon Bedrock, Amazon SageMaker, AWS Lambda	Ospita singoli agenti basati su LLM
Livello di comunicazione	Amazon SQS, Amazon, EventBridge AWS AppFabric	Messaggistica e coordinam ento tra agenti
Memoria condivisa	Amazon DynamoDB, Amazon S3 o OpenSearch	Sistema di memoria o lavagna multiagente
Livello di orchestrazione	AWS Step Functions, condutture AWS Lambda	Logica di kickoff, timeout, fallback e ripetizione
Identificazione dell'agente	Agenti Amazon Bedrock (definiti dal ruolo) e API converse di AWS AppConfig Amazon Bedrock (agenti esterni ad Amazon Bedrock)	Richiamo di strumenti o agenti basati sul ruolo e applicazione dei limiti
Interazione emergente	Amazon EventBridge Pipeline o registri degli agenti	Abilita il routing o l'escalation dinamica delle attività

Riepilogo

La collaborazione tra più agenti distribuisce le attività di risoluzione dei problemi tra agenti modulari e basati sui ruoli. A differenza dell'orchestrazione del flusso di lavoro, i modelli di collaborazione utilizzano intelligenza, resilienza e scalabilità emergenti che rispecchiano il modo in cui gli esseri umani risolvono i problemi. È particolarmente utile per domini aperti, attività creative, ragionamento multimodale e ambienti che traggono vantaggio da prospettive diverse.

Conclusioni

I modelli discussi in precedenza illustrano gli approcci fondamentali alle implementazioni nel mondo reale dell'intelligenza artificiale agentica. Dal ragionamento di base all'intelligenza potenziata dalla memoria, ogni modello è configurato in modo univoco per la percezione, la cognizione e l'azione, basate su autonomia, asincronia e azione.

Questi modelli condividono vocabolari e modelli tecnici per la creazione di sistemi intelligenti e orientati agli obiettivi. Sia che un pattern sia incorporato in un'interfaccia utente, orchestrato tramite servizi cloud o coordinato tra team di agenti, ogni pattern è adattabile e modulare.

Da asporto

- Gli schemi degli agenti sono componibili: la maggior parte degli agenti del mondo reale combina due o più schemi (ad esempio, un agente vocale con ragionamento e memoria basati su strumenti).
- La progettazione degli agenti è contestuale: scegli i modelli in base alla superficie di interazione, alla complessità delle attività, alla tolleranza alla latenza e ai vincoli specifici del dominio.
- AWSl'implementazione nativa è realizzabile: con Amazon Bedrock SageMaker, Amazon e le architetture event-driven AWS Lambda AWS Step Functions, ogni modello di agente può essere fornito su larga scala.

Flussi di lavoro LLM

Nei modelli degli agenti, abbiamo esplorato i modelli comuni degli agenti di intelligenza artificiale, ciascuno basato su una serie di funzionalità modulari: percezione, azione, apprendimento e cognizione. Alla base del modulo cognitivo di molti modelli di agenti c'è un modello linguistico di grandi dimensioni (LLM) in grado di ragionare, pianificare e prendere decisioni. Tuttavia, invocare un LLM da solo non è sufficiente per produrre un comportamento intelligente e mirato.

Per eseguire attività complesse in modo affidabile, gli agenti devono incorporare il LLM in un flusso di lavoro strutturato, in cui le funzionalità del modello sono potenziate da strumenti, memoria, cicli di pianificazione e logica di coordinamento. Questi flussi di lavoro LLM consentono a un agente di suddividere gli obiettivi, indirizzare le sottoattività, chiamare servizi esterni, riflettere sui risultati e coordinarsi con altri agenti.

Questo capitolo introduce i modelli di progettazione di base per la creazione di moduli cognitivi basati su LLM robusti, estensibili e intelligenti, organizzati attorno a flussi di lavoro riutilizzabili.

In questa sezione

- Panoramica della cognizione aumentata con LLM
- Flusso di lavoro per il concatenamento rapido
- Flusso di lavoro per il routing
- Flusso di lavoro per la parallelizzazione
- Flusso di lavoro per l'orchestrazione
- Flusso di lavoro per valutatori e cicli Reflect-Refine
- Conclusioni

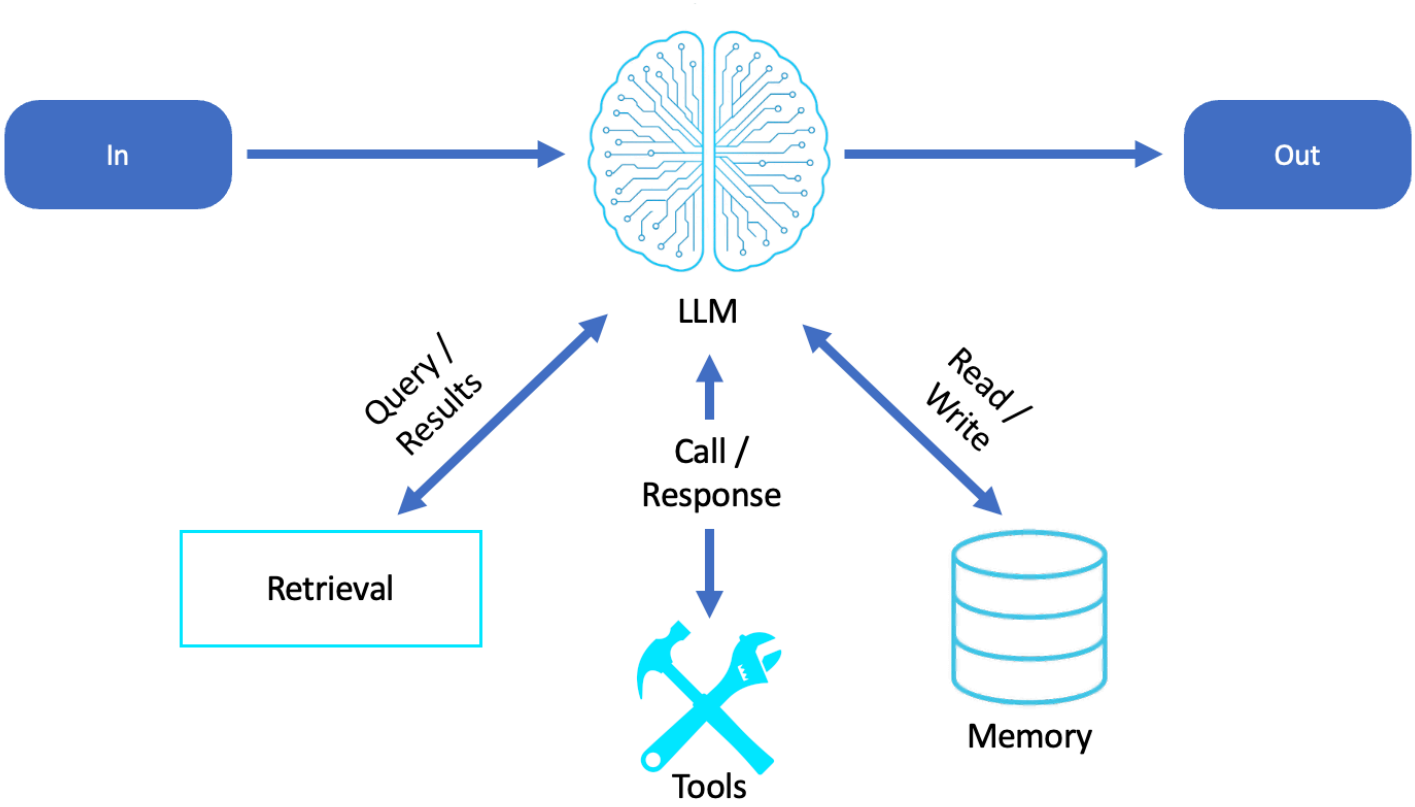
Panoramica della cognizione aumentata con LLM

Fondamentalmente, il modulo cognitivo di un agente software può essere visto come un LLM avvolto da potenziamenti. L'agente può utilizzare i seguenti elementi costitutivi per ragionare in modo efficace all'interno del suo ambiente:

- Richiesta: inquadra l'input utilizzando contesto, istruzioni, esempi e memoria

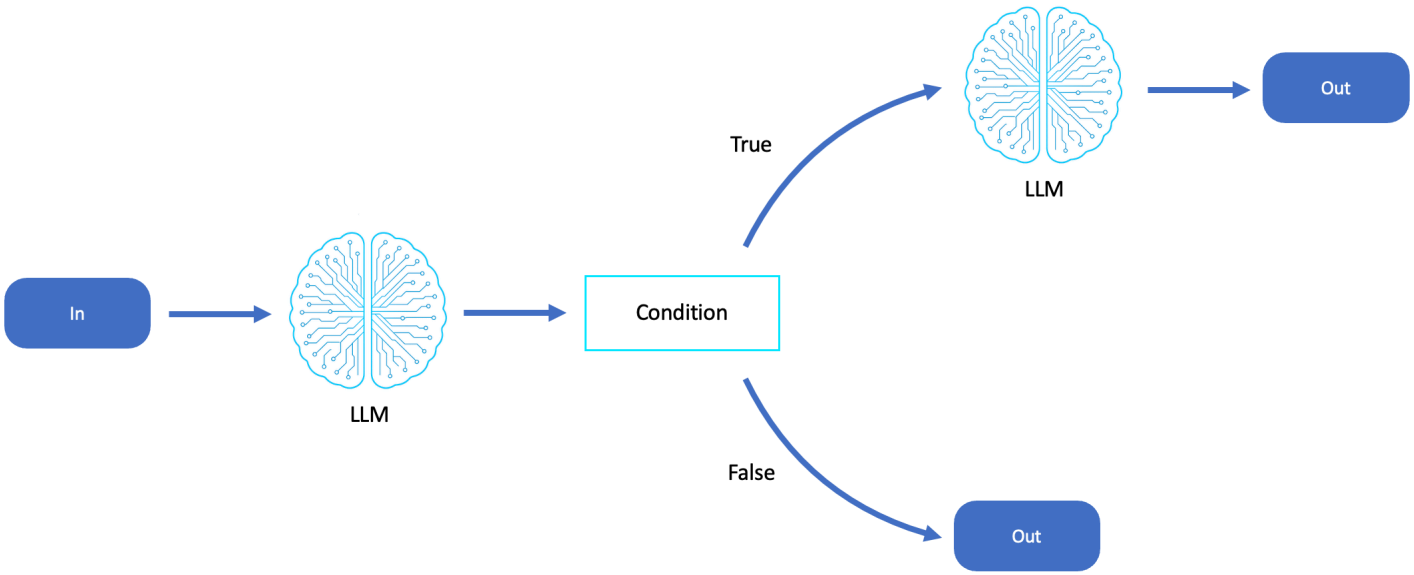
- Recupero: fornitura di conoscenze specifiche del dominio al prompt LLM tramite la ricerca vettoriale up-to-date o la memoria semantica, ad esempio tramite la generazione aumentata di recupero (RAG)
- Utilizzo dello strumento: consentire all'LLM di richiamare o richiamare funzioni per recuperare o agire in base alle informazioni APIs
- Memoria: incorporazione di uno stato persistente o basato sulla sessione nel ciclo di ragionamento, utilizzando database strutturati o riepiloghi contestuali

Questi miglioramenti sono composti da flussi di lavoro che definiscono il modo in cui l'LLM viene utilizzato nel tempo e tra le attività, trasformandolo da un motore stateless a un agente di ragionamento dinamico.



Flusso di lavoro per il concatenamento rapido

Il prompt chaining scompone le attività complesse in una sequenza di passaggi, in cui ogni passaggio è una chiamata LLM discreta che elabora o si basa sull'output di quella precedente.



Il flusso di lavoro di concatenamento dei prompt è adatto per scenari in cui le attività possono essere suddivise logicamente in fasi di ragionamento sequenziali e in cui gli output intermedi determinano la fase successiva. Eccelle nei flussi di lavoro che richiedono un pensiero strutturato, una trasformazione progressiva o un'analisi su più livelli, come la revisione dei documenti, la generazione di codice, l'estrazione della conoscenza e il perfezionamento dei contenuti.

Descrizione

- La complessità dell'attività supera la finestra contestuale o la profondità di ragionamento di una singola chiamata LLM.
- I risultati di una fase (ad esempio, analisi, riepilogo o pianificazione) diventano input per una successiva fase decisionale o di generazione.
- Sono necessari trasparenza e controllo in tutte le fasi di ragionamento (ad esempio, risultati intermedi verificabili).
- Vuoi inserire una logica di convalida, filtraggio o arricchimento esterna tra i passaggi.
- È ideale per gli agenti che operano in cicli di ragionamento in stile pipeline, come agenti di ricerca, assistenti editoriali, sistemi di pianificazione e copiloti multistadio.

Funzionalità

- Catene lineari o ramificate di chiamate LLM
- Risultati intermedi passati come input strutturato o incorporati nei prompt di follow-up

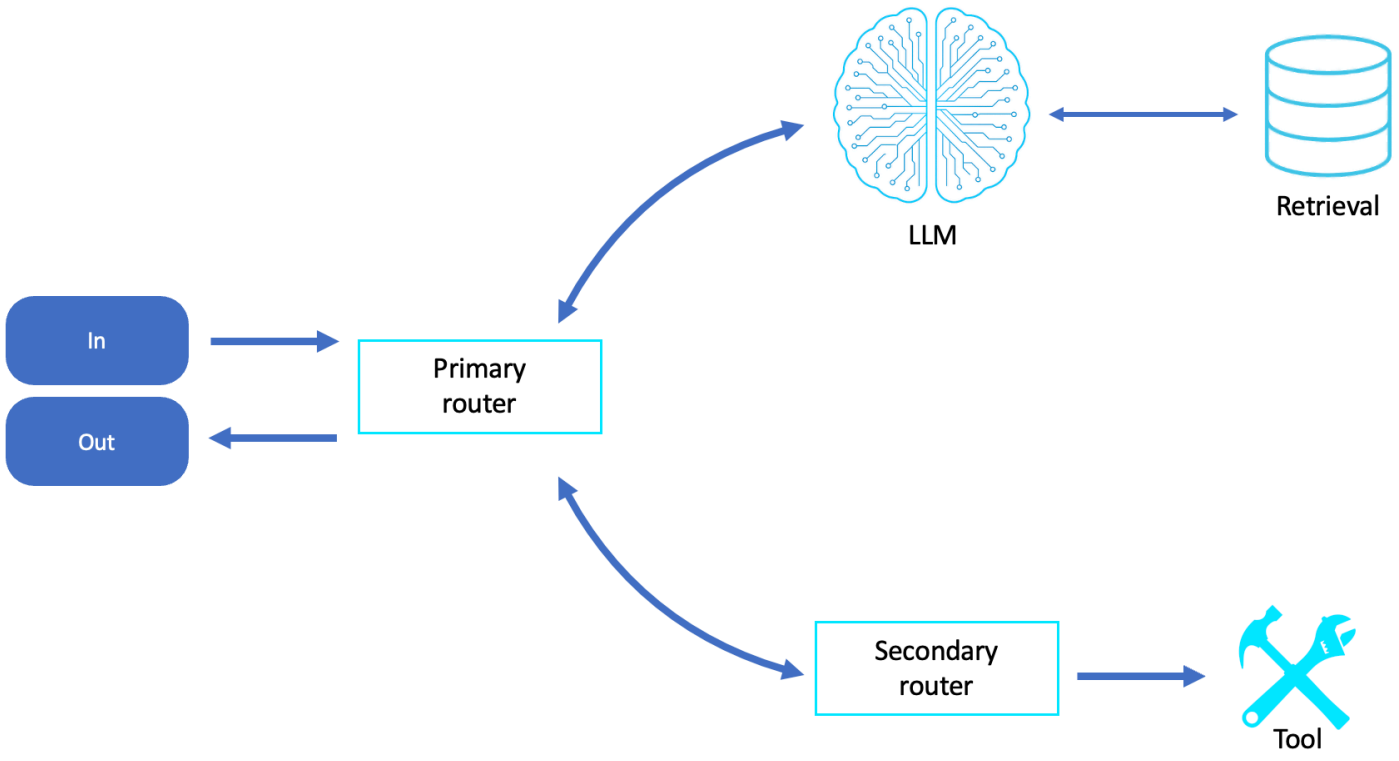
- Può essere orchestrato con o con corridori specifici per un agente AWS Step Functions AWS Lambda

Casi di utilizzo comune

- Attività di ragionamento in più fasi (ad esempio, «riepilogo, riscrittura critica»)
- Assistenti di ricerca che sintetizzano risultati stratificati (ad esempio, «cerca, estrai fatti, rispondi a una domanda»)
- Pipeline di generazione del codice («generate plan write code test code explain output»)

Flusso di lavoro per il routing

Nel modello di routing, un classificatore o un agente router utilizza un LLM per interpretare l'intento o la categoria di una query, quindi indirizza l'input a un'attività o un agente downstream specializzato.



Il flusso di lavoro di routing viene utilizzato in scenari in cui un agente deve classificare rapidamente l'intento di input, il tipo di attività o il dominio e quindi delegare la richiesta a un subagente, uno strumento o un flusso di lavoro specializzato. È particolarmente utile negli agenti di funzionalità,

come quelli che fungono da assistenti generali, nelle porte di accesso alle funzioni aziendali o nelle interfacce AI rivolte agli utenti che si estendono su più domini.

Il routing è particolarmente efficace quando:

- Classificazione delle richieste per una serie di attività (ad esempio ricerca, riepilogo, prenotazione, calcoli).
- Gli input devono essere preelaborati o normalizzati prima di accedere a flussi di lavoro più specializzati.
- Tipi di input diversi (ad esempio, immagini anziché testo, query strutturate e non strutturate) richiedono una gestione personalizzata.
- Un agente funge da centralino conversazionale, delegando attività ad agenti specializzati o microservizi.
- Questo flusso di lavoro è comune nei copiloti specifici del dominio, nei bot di assistenza clienti, nei router di servizi aziendali e negli agenti multimodali, dove il dispacciamento intelligente determina sia la qualità che l'efficienza del comportamento degli agenti.

Funzionalità

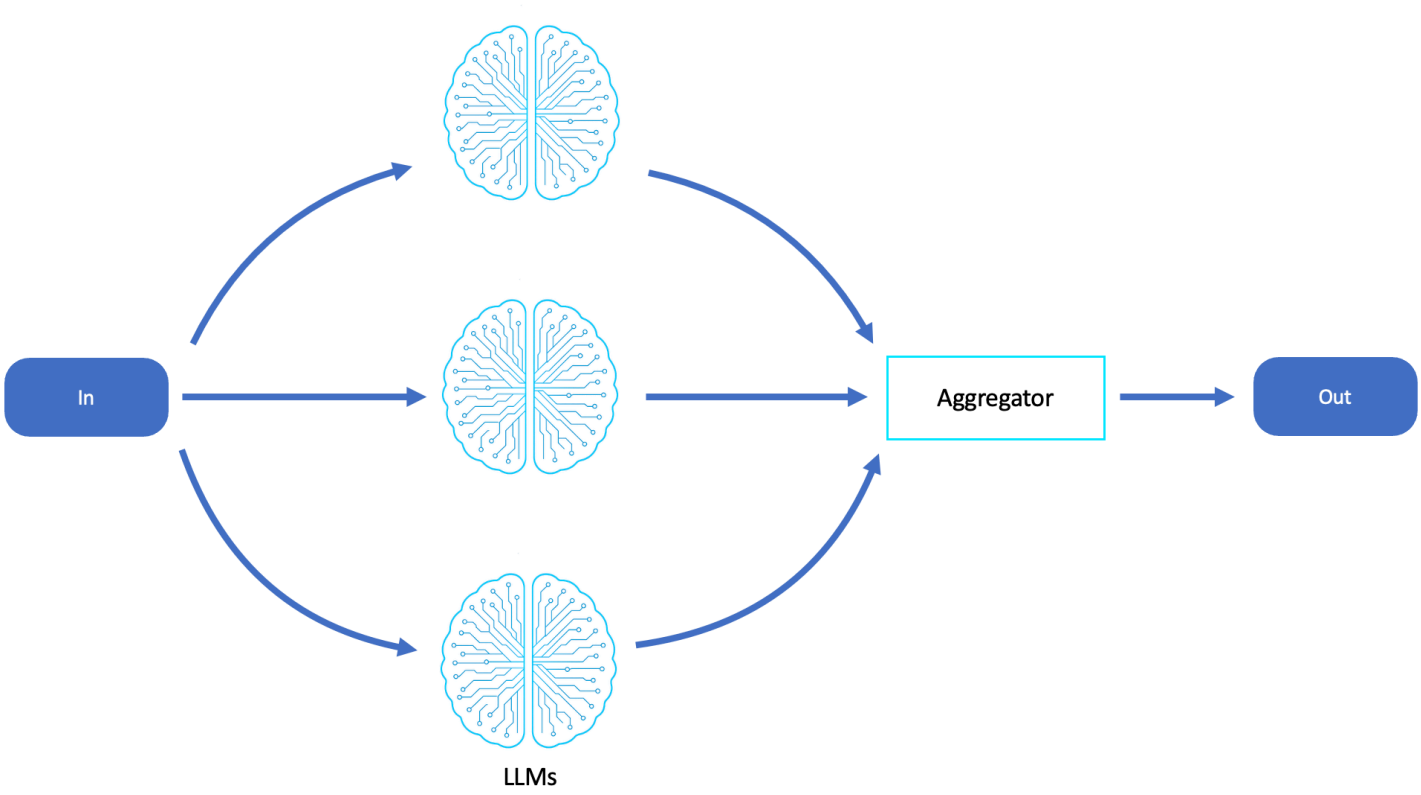
- Un LLM di primo passaggio funge da dispatcher
- I percorsi possono richiamare flussi di lavoro distinti o persino altri modelli di agenti
- Supporta l'espansione modulare delle funzionalità

Casi di utilizzo comune

- Assistenti multidominio («è una questione legale, medica o finanziaria?»)
- Alberi decisionali migliorati con il ragionamento LLM
- Selezione dinamica degli strumenti (ad esempio, ricerca e generazione di codice)

Flusso di lavoro per la parallelizzazione

Questo flusso di lavoro prevede la suddivisione di un'attività in sottoattività indipendenti che possono essere gestite contemporaneamente da più chiamate o agenti LLM. Gli output vengono quindi aggregati a livello di codice e sintetizzati in un risultato.



Il flusso di lavoro di parallelizzazione viene utilizzato quando un'attività può essere suddivisa in sottoattività indipendenti e non sequenziali che possono essere elaborate contemporaneamente, migliorando significativamente l'efficienza, la produttività e la scalabilità. È particolarmente efficace negli spazi problematici ad alto contenuto di dati, orientati ai batch o multiprospettici in cui l'agente deve analizzare o generare contenuti su più input.

La parallelizzazione è particolarmente efficace quando:

- Le sottoattività non dipendono dai reciproci risultati intermedi, pertanto possono essere eseguite in parallelo senza coordinamento.
- Un'attività prevede la ripetizione dello stesso processo di ragionamento su più elementi (ad esempio, riepilogando più documenti o valutando un elenco di opzioni).
- Molteplici ipotesi o prospettive vengono esplorate in parallelo per promuovere la diversità, la creatività o la solidità.
- È necessario ridurre la latenza per le richieste ad alto volume o ad alta frequenza tramite l'esecuzione simultanea di LLM.
- Questo flusso di lavoro viene comunemente utilizzato negli agenti di elaborazione di documenti, nei motori di sondaggi o di confronto, nei riepiloghi di batch, nei brainstorming multiagente e nelle

attività di classificazione o etichettatura scalabili, specialmente laddove il ragionamento rapido e parallelo rappresenta un vantaggio in termini di prestazioni.

Funzionalità

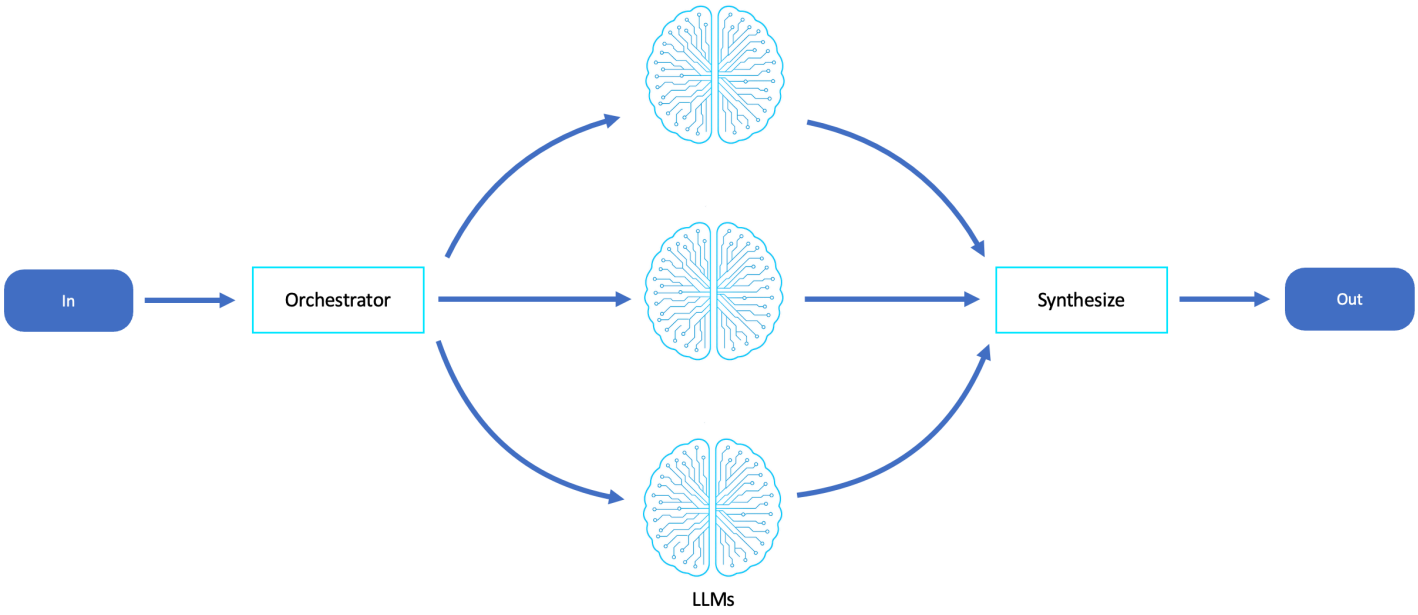
- Esecuzione parallela di attività LLM (utilizzando o uno stato della mappa) AWS Lambda AWS Fargate AWS Step Functions
- Richiede l'allineamento, la convalida o la deduplicazione dei risultati in fase di sintesi
- Ideale per i loop di agenti stateless

Casi di utilizzo comune

- Analisi parallela di più documenti o prospettive
- Generazione di bozze, riepiloghi o piani diversi
- Accelerazione della produttività tra i lavori in batch

Flusso di lavoro per l'orchestrazione

Un agente di orchestrazione centrale utilizza un LLM per pianificare, scomporre e delegare le sottoattività ad agenti o modelli di lavoro specializzati, ciascuno con un ruolo o un'esperienza di dominio specifici. Ciò rispecchia le strutture dei team umani e supporta il comportamento emergente tra più agenti.



Il flusso di lavoro di orchestrazione è ideale per scenari complessi, gerarchici o multidisciplinari, che richiedono una scomposizione strutturata e un'esecuzione specializzata. È particolarmente adatto per attività che richiedono la divisione del lavoro, in cui i diversi sottocomponenti di un'attività vengono gestiti al meglio da agenti con capacità, conoscenze o set di strumenti distinti.

Questo flusso di lavoro è particolarmente efficace quando:

- Le attività possono essere suddivise in sottoattività che variano per ambito, tipo o ragionamento (ad esempio, pianificazione, ricerca, implementazione e test).
- Un LLM o un meta-agente deve coordinare altri agenti, monitorare i progressi e sintetizzare i risultati.
- Desiderate modularizzare le responsabilità degli agenti, abilitando la scalabilità, il riutilizzo e l'ottimizzazione specializzata.
- Il sistema richiede un comportamento basato sui ruoli, che imita il modo in cui i team umani (ad esempio, project manager, sviluppatori e revisori) operano in collaborazione.

L'orchestrazione è ideale per agenti di pianificazione a più turni, copiloti di sviluppo software, agenti di processo aziendali ed esecutori di progetto autonomi. È particolarmente utile quando si implementano sistemi multiagente che richiedono una suddivisione centralizzata delle attività ma una logica di esecuzione distribuita, che consente l'estensibilità e un comportamento più spiegabile tra i livelli degli agenti.

Funzionalità

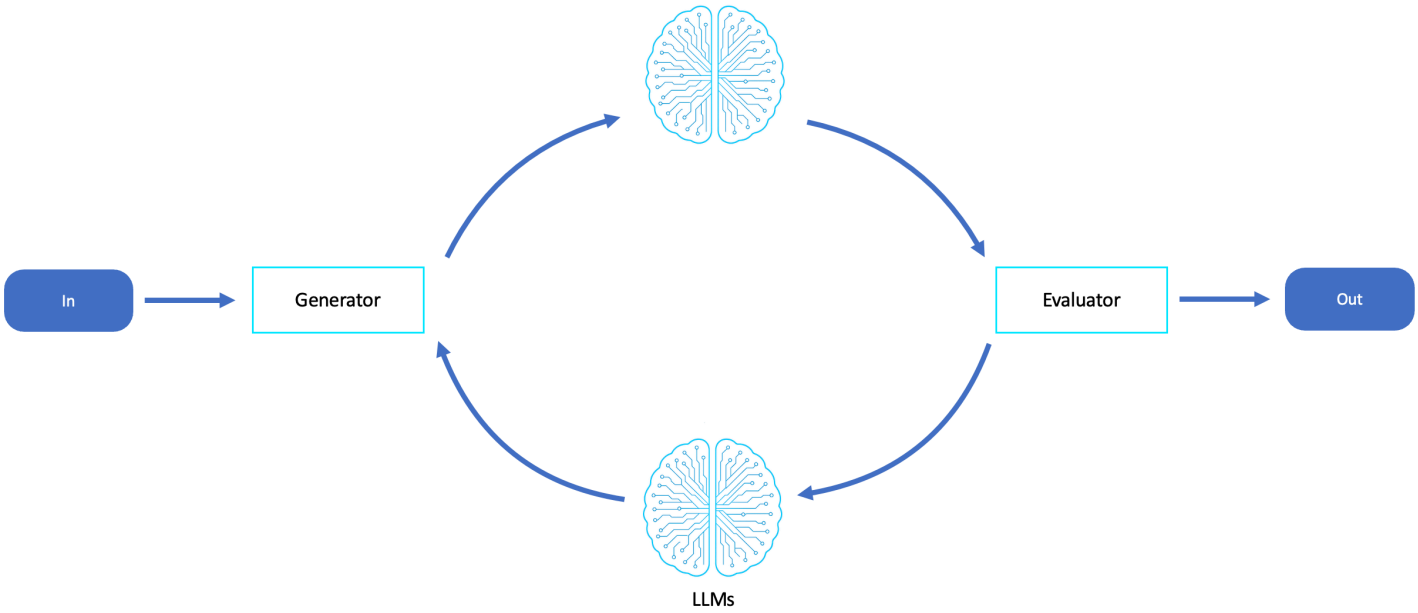
- Orchestrator esegue il meta-ragionamento degli obiettivi
- Gli agenti di lavoro possono includere l'accesso agli strumenti, la memoria o richieste specifiche del dominio
- Può essere gerarchico (ovvero delega di attività a più livelli)

Casi di utilizzo comune

- Responsabili di progetto, ricercatori coordinatori, scrittori e agenti di garanzia della qualità
- Copiloti di codifica che combinano pianificazione, esecuzione e test
- Agenti che supervisionano le toolchain o i modelli di accesso alle API

Flusso di lavoro per valutatori e cicli Reflect-Refine

Questo flusso di lavoro fornisce un ciclo di feedback in cui un LLM genera un risultato e un altro valuta o critica il risultato. Ciò promuove l'autoriflessione, l'ottimizzazione e i miglioramenti iterativi.



Il flusso di lavoro del valutatore è ideale per scenari in cui la qualità, la precisione e l'allineamento dell'output sono importanti e in cui la generazione a passaggio singolo è inaffidabile o insufficiente. Questo flusso di lavoro eccelle quando gli agenti devono autocriticare, iterare e perfezionare i propri

risultati, per soddisfare uno standard di correttezza più elevato o per esplorare alternative migliori basate sul feedback.

Questo flusso di lavoro è particolarmente efficace quando:

- L'output include metriche di qualità soggettive (ad esempio stile, tono e leggibilità) o criteri oggettivi (ad esempio, correttezza, sicurezza e prestazioni).
- L'agente deve ragionare sulla base di compromessi, valutare i vincoli o ottimizzare per raggiungere un obiettivo.
- Sono necessarie ridondanza e garanzia di qualità integrate, specialmente in domini regolamentati, rivolti ai clienti o creativi.
- Human-in-the-loop la revisione è costosa o non disponibile e si desidera una convalida autonoma.

Questo flusso di lavoro viene utilizzato per la generazione di contenuti, la sintesi e la revisione del codice, l'applicazione delle policy, il controllo dell'allineamento, l'ottimizzazione delle istruzioni e la postelaborazione RAG. È utile anche per gli agenti che si migliorano da soli, dove il feedback continuo aiuta a modellare risposte migliori nel tempo per creare cicli decisionali affidabili e autonomi.

Casi di utilizzo comune

- Agenti della squadra rossa rispetto agli agenti della squadra blu
- Agenti che generano, valutano e rivedono codice o piani
- Garanzia della qualità, rilevamento delle allucinazioni e applicazione dello stile

Funzionalità

- Supporta la generazione e la valutazione disaccoppiate utilizzando diversi modelli (ad esempio, Claude per la generazione e Mistral per la valutazione)
- Il feedback è strutturato e utilizzato per richiedere risultati rivisti
- Supporta più iterazioni o soglie di convergenza

Conclusioni

LLMs forniscono il nucleo cognitivo dei moderni agenti software, ma l'invocazione di modelli grezzi non è sufficiente per ottenere un'intelligenza mirata, solida e controllabile. Per passare dalla

generazione di output al ragionamento strutturato e al comportamento allineato agli obiettivi, è LLMs necessario incorporarlo in modelli di flusso di lavoro intenzionali che definiscono il modo in cui i modelli elaborano gli input, gestiscono i contesti e coordinano le azioni.

I flussi di lavoro LLM introducono le basi per creare il modulo cognitivo di un agente:

- Il concatenamento rapido suddivide il ragionamento complesso in fasi modulari e verificabili.
- Il routing consente una classificazione intelligente delle attività e una delega mirata.
- La parallelizzazione accelera la produttività e promuove ragionamenti diversificati.
- L'orchestrazione degli agenti struttura la collaborazione tra più agenti attraverso la scomposizione delle attività e l'esecuzione basata sui ruoli.
- Evaluator (reflect-refine loop) consente il miglioramento automatico, il controllo della qualità e il controllo dell'allineamento.

Ogni flusso di lavoro rappresenta uno schema componibile che può essere adattato alle esigenze dell'operatore, alla complessità dell'attività e alle aspettative dell'utente. Questi flussi di lavoro non si escludono a vicenda. Sono elementi costitutivi che vengono spesso combinati in architetture ibride che supportano il ragionamento dinamico, il coordinamento multiagente e l'affidabilità di livello aziendale.

Passando al prossimo capitolo sui modelli di flusso di lavoro agentici, questi flussi di lavoro LLM riappariranno come strutture integrate all'interno di sistemi più grandi, che supportano la delega degli obiettivi, l'orchestrazione degli strumenti, i cicli decisionali e l'autonomia del ciclo di vita. La padronanza di questi flussi di lavoro LLM è essenziale per progettare agenti software che non si limitino a prevedere il testo, ma che ragionino, si adattino e agiscano in modo mirato.

Modelli di flusso di lavoro agentici

I modelli di flusso di lavoro Agentic integrano agenti software modulari con flussi di lavoro LLM (Structured Large Language Model), consentendo ragionamenti e azioni autonomi. Sebbene ispirati alle tradizionali architetture serverless e basate sugli eventi, questi modelli spostano la logica di base dal codice statico agli agenti basati su LLM, offrendo una maggiore adattabilità e un processo decisionale contestuale. Questa evoluzione trasforma le architetture cloud convenzionali da sistemi deterministici a sistemi capaci di interpretazione dinamica e aumento intelligente, pur mantenendo i principi fondamentali di scalabilità e reattività.

In questa sezione

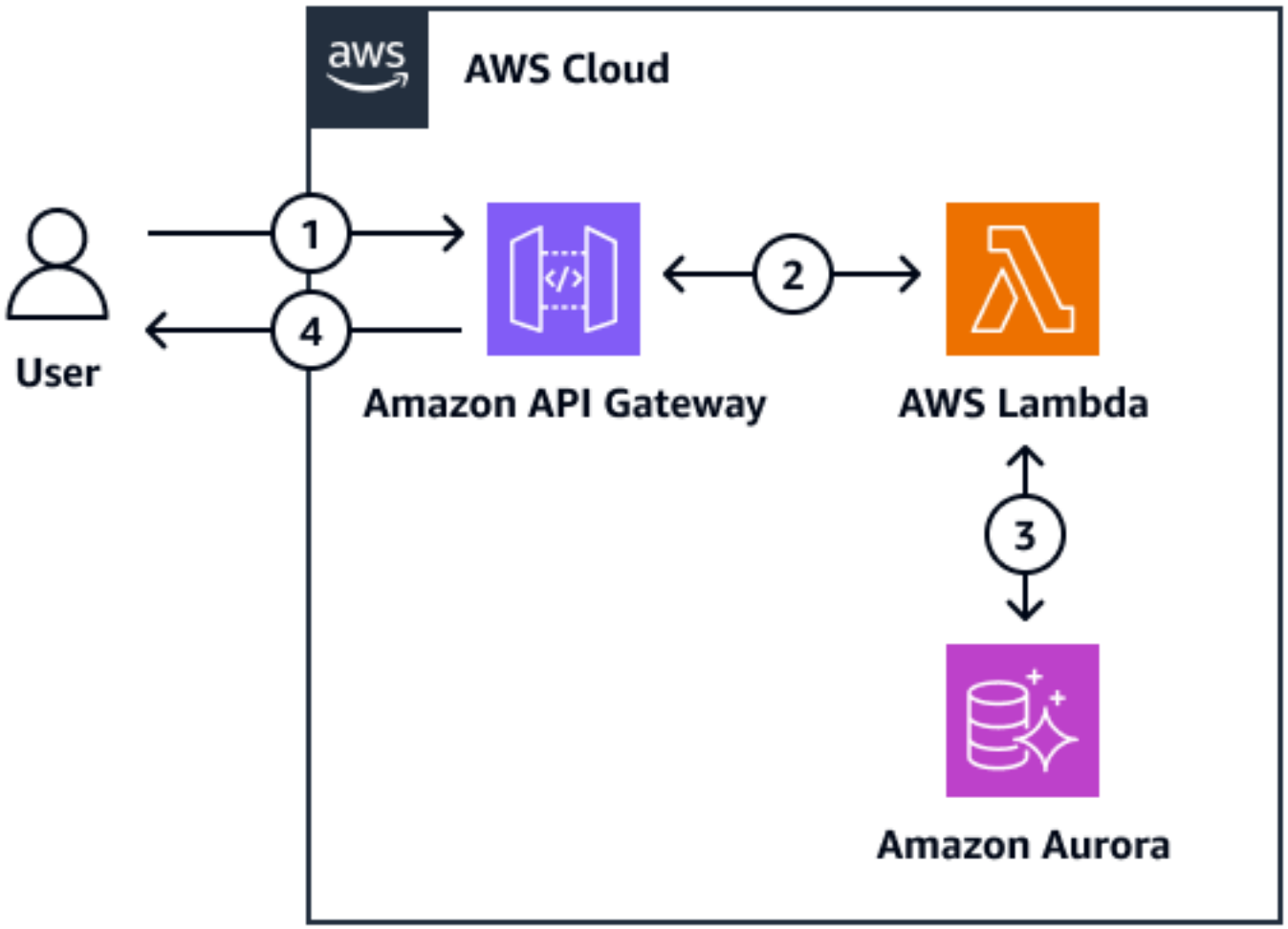
- Dai sistemi basati sugli eventi a quelli basati sulla cognizione aumentata
- Rapido concatenamento degli schemi della saga
- Schemi di invio dinamici di routing
- Parallelizzazione e schemi di dispersione
- Schemi di orchestrazione Saga
- Evaluator Reflect-Refine Loop Patterns
- Progettazione di flussi di lavoro agentici su AWS
- Conclusioni

Dai sistemi basati sugli eventi a quelli basati sulla cognizione aumentata

Le moderne architetture cloud, in particolare quelle basate su principi serverless e basati sugli eventi, si sono tradizionalmente basate su modelli come routing, fan-out e arricchimento per creare sistemi reattivi e scalabili. I sistemi di intelligenza artificiale Agentic si basano su queste basi e le riformulano attorno al ragionamento e alla flessibilità cognitiva basati su LLM. Questo approccio consente funzionalità di automazione e risoluzione dei problemi più sofisticate, rivoluzionando potenzialmente il modo in cui le attività complesse vengono gestite negli ambienti cloud.

Architettura basata su eventi

Il diagramma seguente mostra un tipico sistema distribuito:

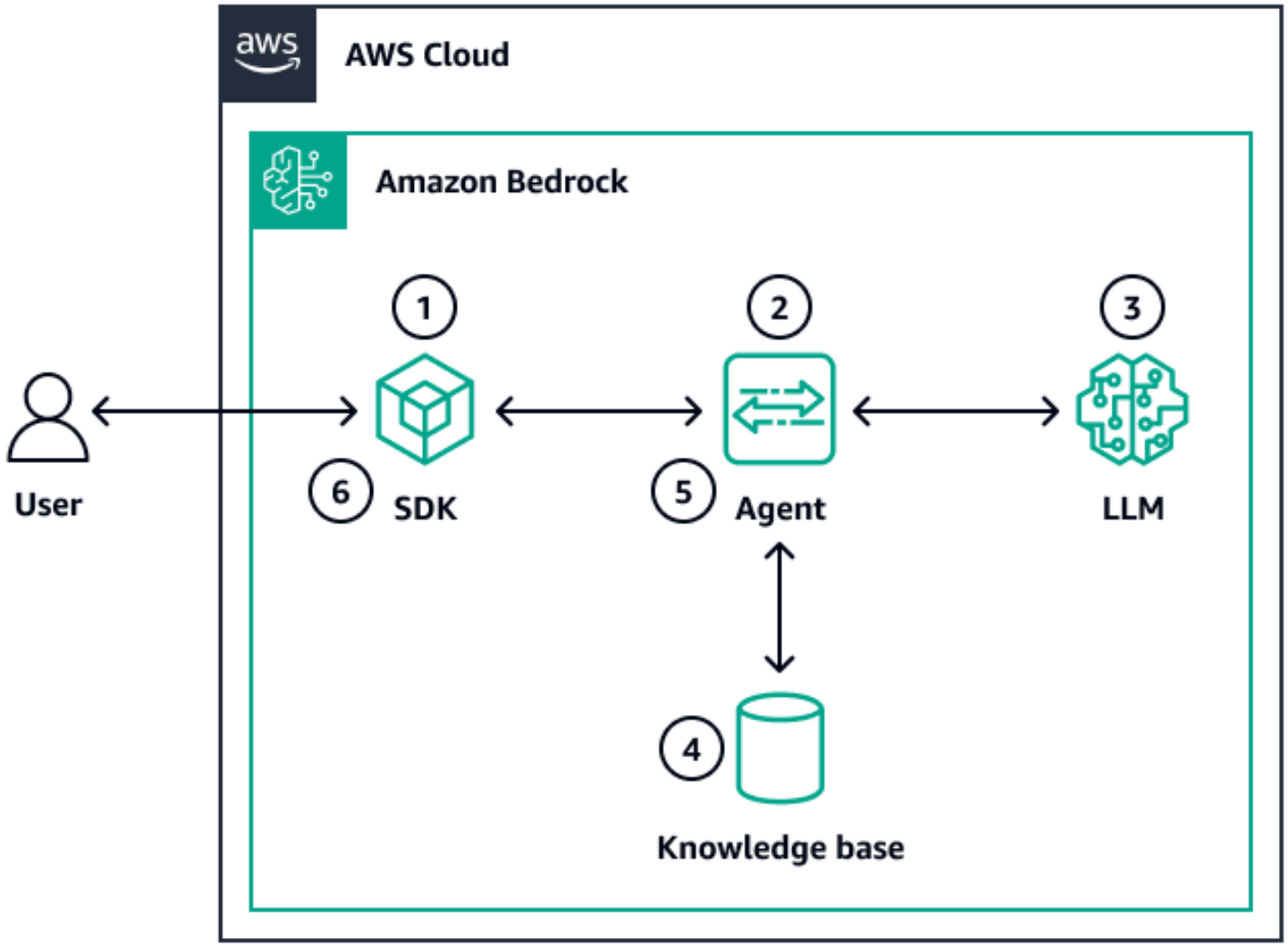


1. Un utente invia una richiesta ad Amazon API Gateway.
2. Amazon API Gateway indirizza la richiesta a una AWS Lambda funzione.
3. AWS Lambda esegue l'arricchimento dei dati interrogando un database Amazon Aurora
4. Amazon API Gateway restituisce il payload arricchito al chiamante.

Questa struttura è affidabile e scalabile, ma è fondamentalmente statica. Le regole aziendali e i percorsi logici devono essere codificati in modo esplicito e l'adattamento ai contesti mutevoli o alle informazioni incomplete è limitato.

Flussi di lavoro basati sulla cognizione

Le architetture agentiche aggiungono l'aumento cognitivo a un sistema basato sugli eventi. Il diagramma seguente mostra un equivalente agentico:



1. Un utente invia una richiesta tramite una chiamata SDK o API.
2. Un agente Amazon Bedrock riceve la richiesta.
3. L'agente interpreta la query richiamando un LLM
4. L'agente esegue l'arricchimento semantico effettuando ricerche nella knowledge base di Amazon Bedrock o in altre fonti di dati esterne.
5. L'LLM sintetizza una risposta ricca di contesto e allineata agli obiettivi.
6. Il sistema restituisce una risposta sintetizzata all'utente.

In questo flusso, l'LLM utilizza la logica, comprende l'intento, recupera e combina il contesto pertinente e quindi decide come rispondere al meglio. Questo modello rispecchia il modello di arricchimento tradizionale, in cui i messaggi vengono arricchiti con dati esterni prima di essere

inoltrati ulteriormente. Nei sistemi agentici, tuttavia, questo arricchimento non è una ricerca statica. L'arricchimento è invece dinamico, guidato semanticamente e guidato da uno scopo.

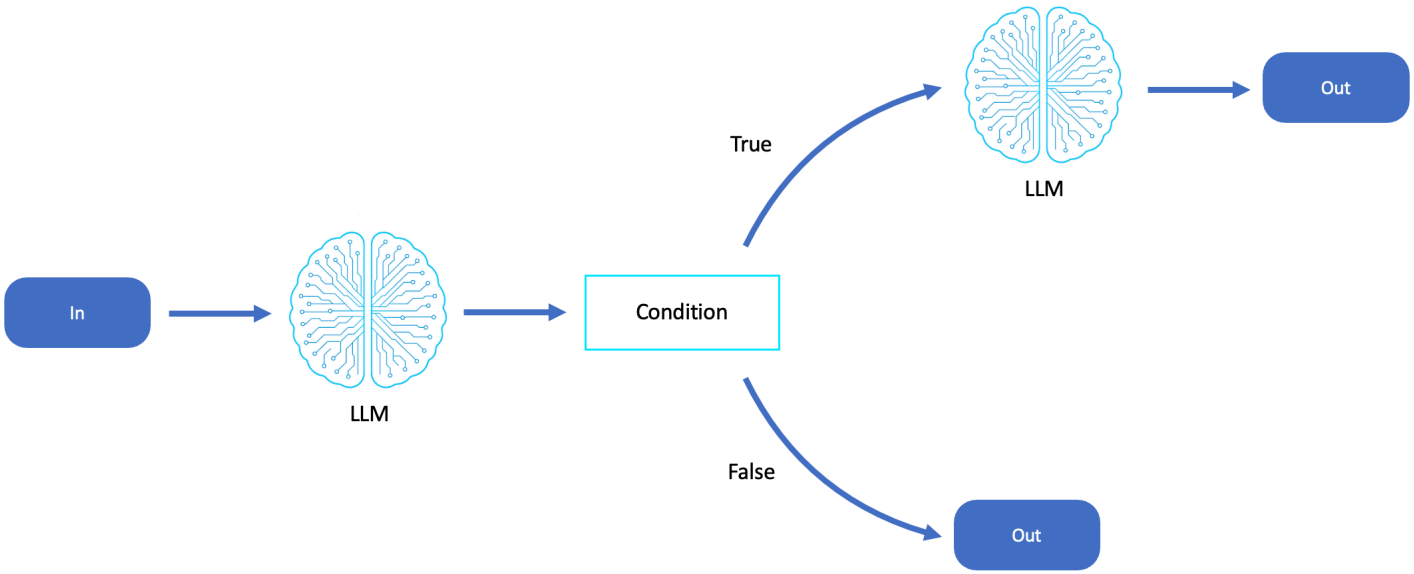
Approfondimenti fondamentali

Ogni flusso di lavoro LLM può essere mappato su un modello di flusso di lavoro agentico, che rispecchia ed evolve gli stili di architettura tradizionali basati sugli eventi. Un elemento fondamentale dei flussi di lavoro agentici è la capacità di ampliare il contesto di un LLM con dati, strumenti e memoria. Questo crea un ciclo di ragionamento informato, adattivo e allineato alle intenzioni dell'utente. Laddove i sistemi tradizionali arricchiscono i messaggi con dati di ricerca, i sistemi agentici consentono al software di agire meno come script e più come collaboratori intelligenti.

Rapido concatenamento degli schemi della saga

Reimmaginando il prompt chaining LLM come una saga basata sugli eventi, sblocchiamo un nuovo modello operativo: i flussi di lavoro diventano distribuiti, recuperabili e coordinati semanticamente tra agenti autonomi. Ogni fase di pronta risposta viene riformulata come un'attività atomica, emessa come evento, consumata da un agente dedicato e arricchita con metadati contestuali.

Il diagramma seguente è un esempio di concatenamento di prompt LLM:

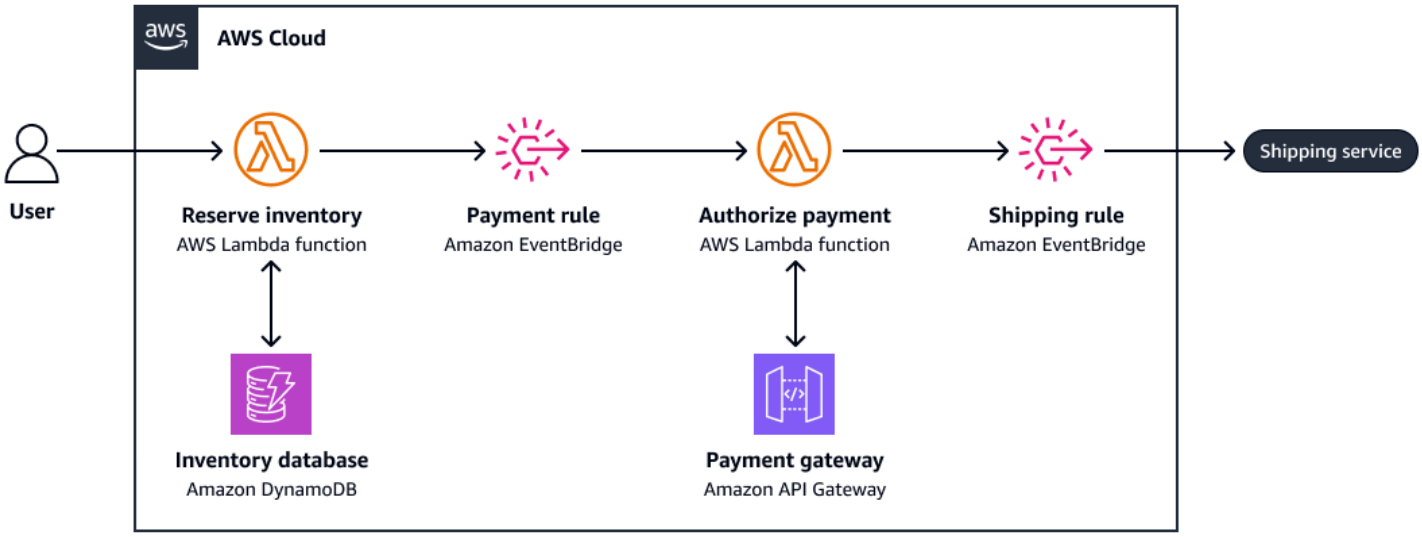


Coreografia saga

Il modello coreografico della saga è un approccio di implementazione in sistemi distribuiti che non ha un coordinatore centrale. Invece, ogni servizio o componente pubblica eventi che attivano l'azione

successiva del flusso di lavoro. Questo modello è ampiamente utilizzato nei sistemi distribuiti per la gestione delle transazioni tra più servizi. In una saga, il sistema esegue una serie di transazioni locali coordinate. Se una fallisce, il sistema attiva azioni di compensazione per mantenere la coerenza.

Il diagramma seguente è un esempio di coreografia da saga:



1. Inventario di riserva
2. Autorizza il pagamento
3. Crea un ordine di spedizione

Se la fase 3 fallisce, il sistema richiama azioni di compensazione (ad esempio, annullare un pagamento o rilasciare l'inventario).

Questo modello è particolarmente utile nelle architetture basate sugli eventi, in cui i servizi sono collegati in modo flessibile e gli stati devono essere risolti in modo coerente nel tempo, anche in presenza di guasti parziali.

Schema di concatenamento rapido

Il concatenamento rapido ricorda lo schema della saga sia nella struttura che nello scopo. Esegue una serie di passaggi di ragionamento che vengono creati in sequenza preservando il contesto e consentendo rollback e revisioni.

Coreografia dell'agente

1. LLM interpreta una query utente complessa e genera un'ipotesi

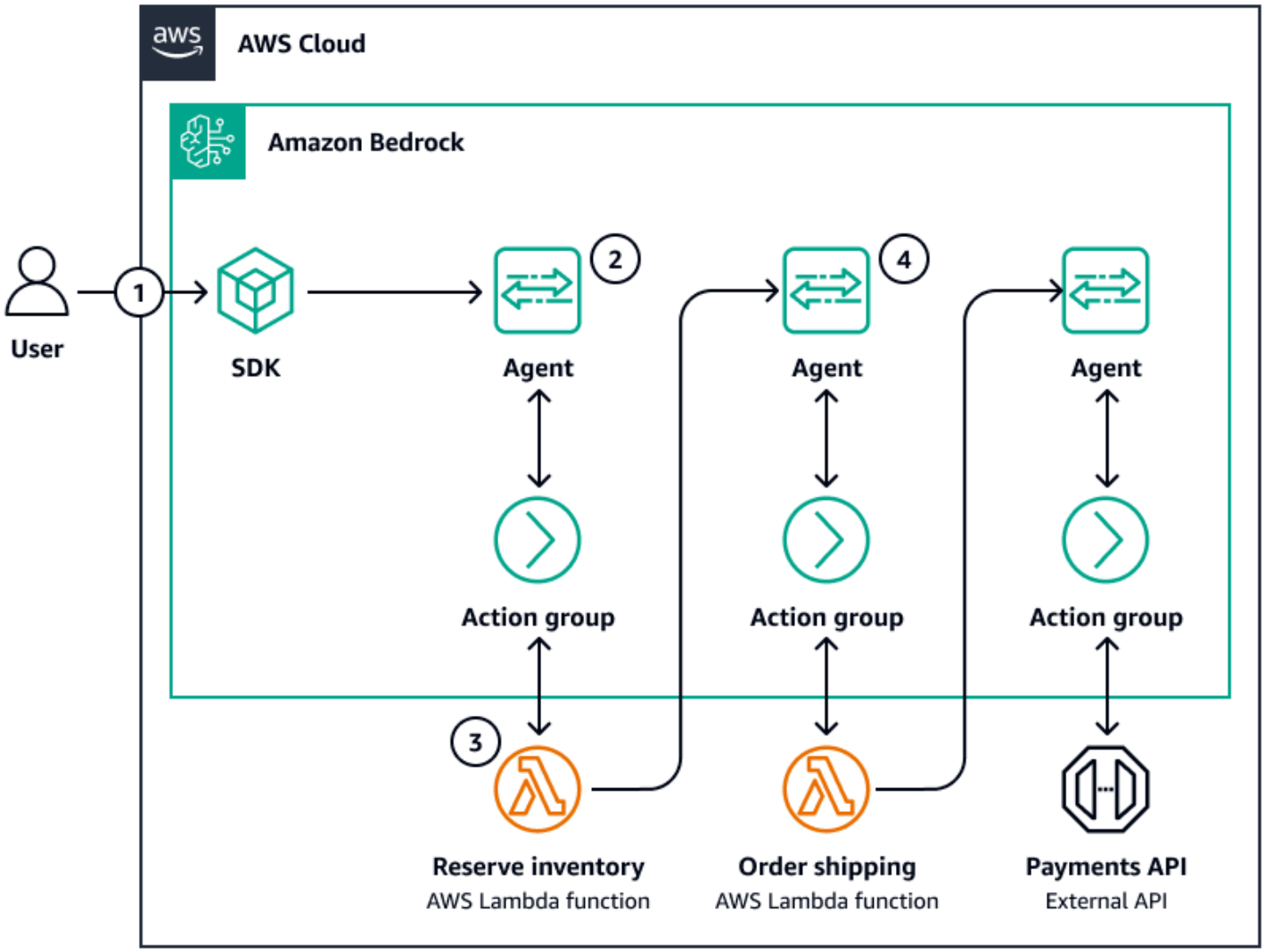
2. LLM elabora un piano per risolvere il problema
3. LLM esegue una sottoattività (ad esempio, utilizzando una chiamata a uno strumento o recuperando conoscenze)
4. LLM perfeziona l'output o rivisita un passaggio precedente se ritiene un risultato insoddisfacente

Se un risultato intermedio è difettoso, il sistema può eseguire una delle seguenti operazioni:

- Riprova i passaggi utilizzando un approccio diverso
- Torna a un prompt precedente e ripianifica
- Utilizzate un ciclo di valutazione (ad esempio, dal pattern valutatore-ottimizzatore) per rilevare e correggere gli errori

Analogamente al modello saga, il prompt chaining consente meccanismi di avanzamento e rollback parziali. Ciò avviene attraverso il perfezionamento iterativo e la correzione diretta da LLM anziché attraverso la compensazione delle transazioni del database.

Il diagramma seguente è un esempio di coreografia tra agenti:



1. Un utente invia una richiesta tramite un SDK.
2. Un agente Amazon Bedrock orchestra il ragionamento attraverso quanto segue:
   - Interpretazione (LLM)
   - Pianificazione (LLM)
   - Esecuzione tramite uno strumento o una base di conoscenze
   - Costruzione della risposta
3. Se uno strumento si guasta o restituisce dati insufficienti, l'agente può ripianificare o riformulare dinamicamente l'attività.
4. La memoria (ad esempio, un archivio vettoriale a breve termine) può preservare il proprio stato attraverso i passaggi

Da asporto

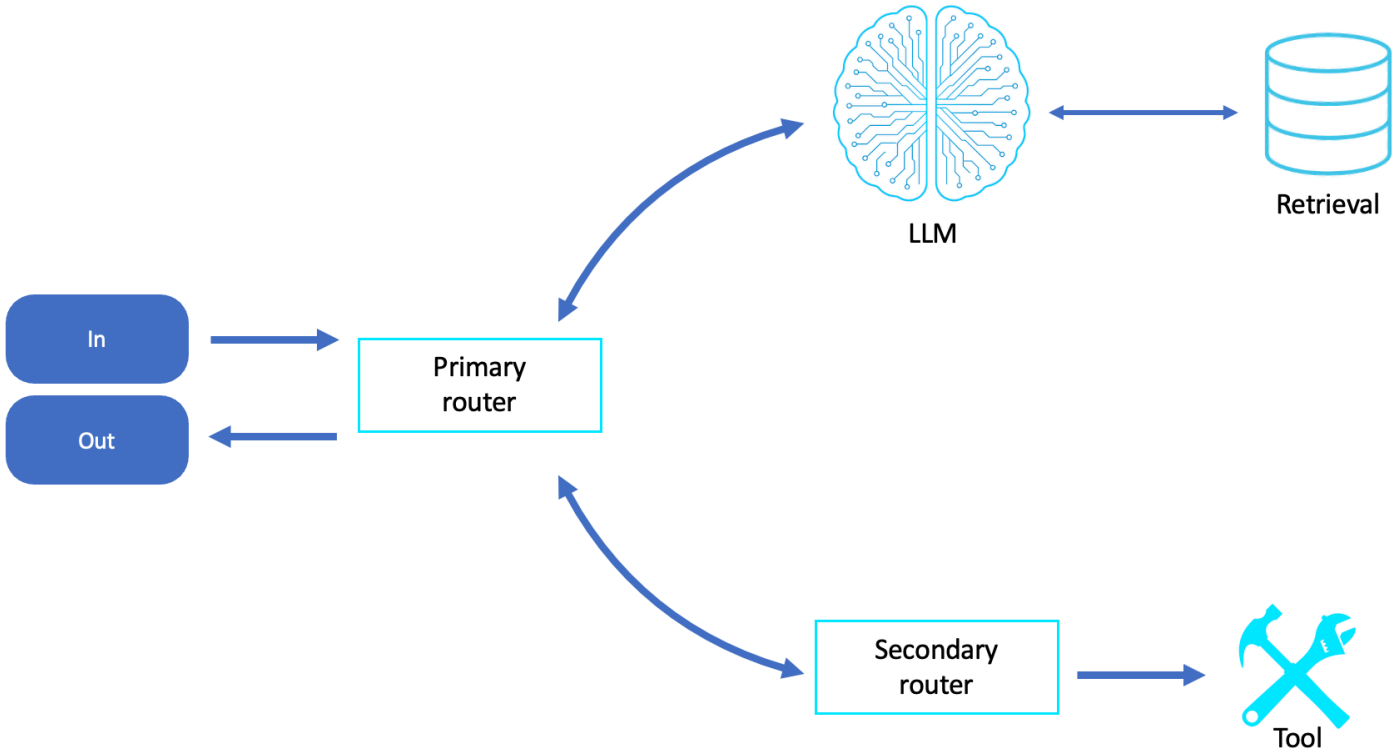
Laddove Saga Pattern gestisce le chiamate di servizio distribuite con una logica di compensazione, il prompt chaining gestisce le attività di ragionamento con sequenziamento riflessivo e ripianificazione adattiva. Entrambi i sistemi consentono progressi incrementali, punti decisionali decentralizzati e ripristino degli errori, e tutto ciò avviene attraverso un ragionamento informato anziché un rigido rollback.

Il concatenamento rapido introduce il ragionamento transazionale, che è l'equivalente cognitivo delle saghe. Cioè, ogni «pensiero» viene rivalutato, rivisto o abbandonato come parte di un dialogo più ampio mirato.

Schemi di invio dinamici di routing

Nei moderni sistemi agentici, in cui le attività spaziano dall'analisi dei documenti alla generazione autonoma di software, la capacità di indirizzare dinamicamente le richieste all'agente o modello di linguaggio di grandi dimensioni (LLM) più capace diventa fondamentale. La logica di routing statica, spesso incorporata negli script di orchestrazione o nei livelli API, non ha l'adattabilità richiesta per ambienti in tempo reale, multimodello e con più capacità. Per risolvere questo problema, i flussi di lavoro di routing LLM possono essere trasformati in un'architettura basata sugli eventi che sfrutta uno schema di invio dinamico, trasformando le chiamate LLM in eventi indirizzati in modo intelligente e sensibili al contesto.

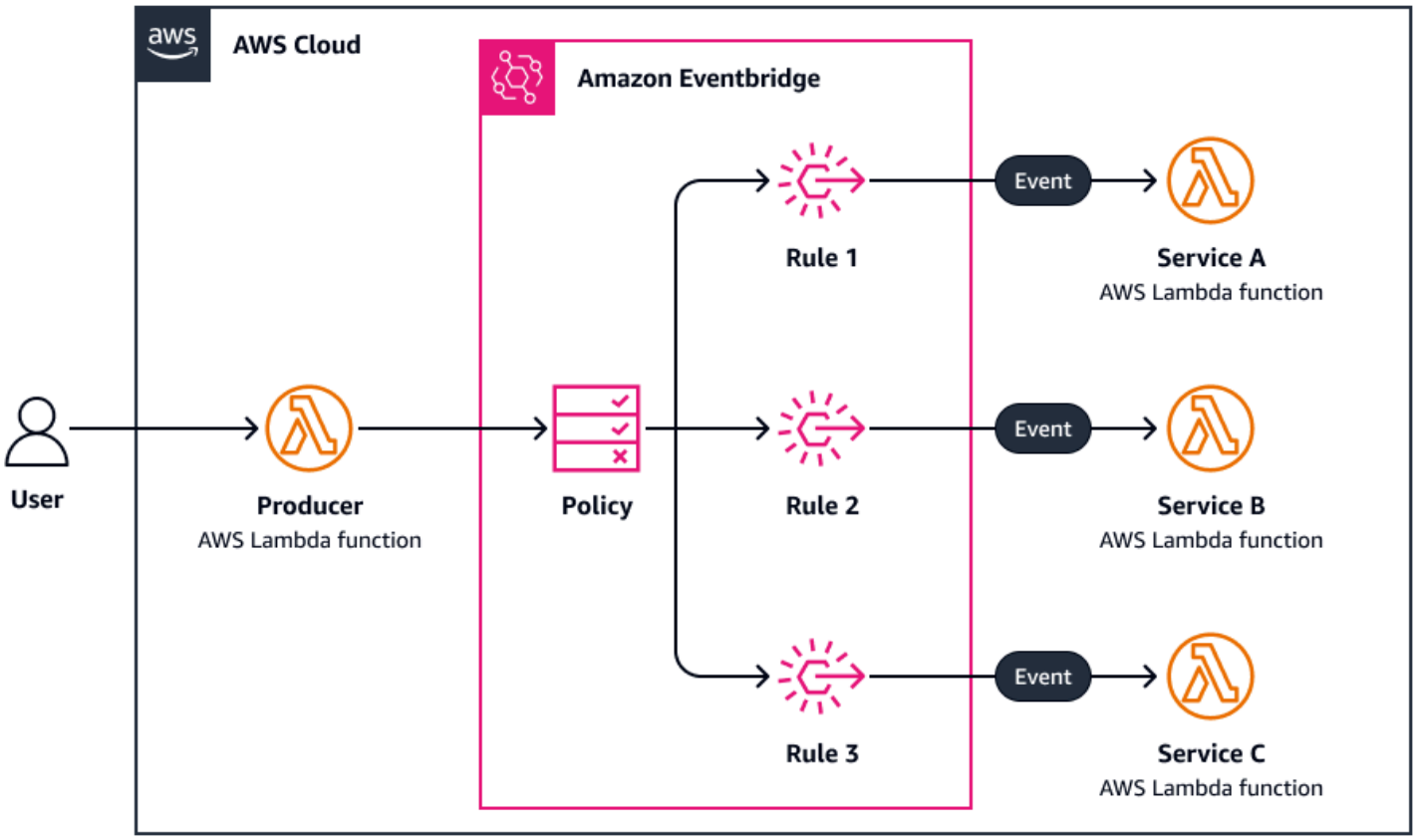
Il diagramma seguente è un esempio di routing LLM:



Invio dinamico

Nei sistemi distribuiti tradizionali, il pattern di invio dinamico seleziona e richiama servizi specifici in fase di esecuzione in base agli attributi degli eventi in entrata, come il tipo di evento, l'origine e il payload. Questo viene comunemente implementato utilizzando Amazon EventBridge, che può valutare e indirizzare gli eventi in entrata verso destinazioni appropriate (ad esempio AWS Step Functions, AWS Lambda funzioni o attività di Amazon Elastic Container Service).

Il diagramma seguente è un esempio di invio dinamico:



1. Un'applicazione emette un evento (ad esempio, {"type»: «orderCreated», «priority»: «high"}).
2. Amazon EventBridge valuta l'evento in base alle sue regole di routing.
3. In base agli attributi di un evento, il sistema invia dinamicamente a quanto segue:
   - `HighPriorityOrderProcessor`(servizio A)
   - `StandardOrderProcessor`(servizio B)
   - `UpdateOrderProcessor`(servizio C)

Questo modello supporta l'accoppiamento libero, la specializzazione basata sul dominio e l'estensibilità del runtime. Ciò consente ai sistemi di rispondere in modo intelligente ai mutevoli requisiti e alla semantica degli eventi.

Routing basato su LLM

Nei sistemi agentici, il routing esegue anche la delega dinamica delle attività, ma invece delle EventBridge regole di Amazon o dei filtri dei metadati, l'LLM classifica e interpreta l'intento dell'utente

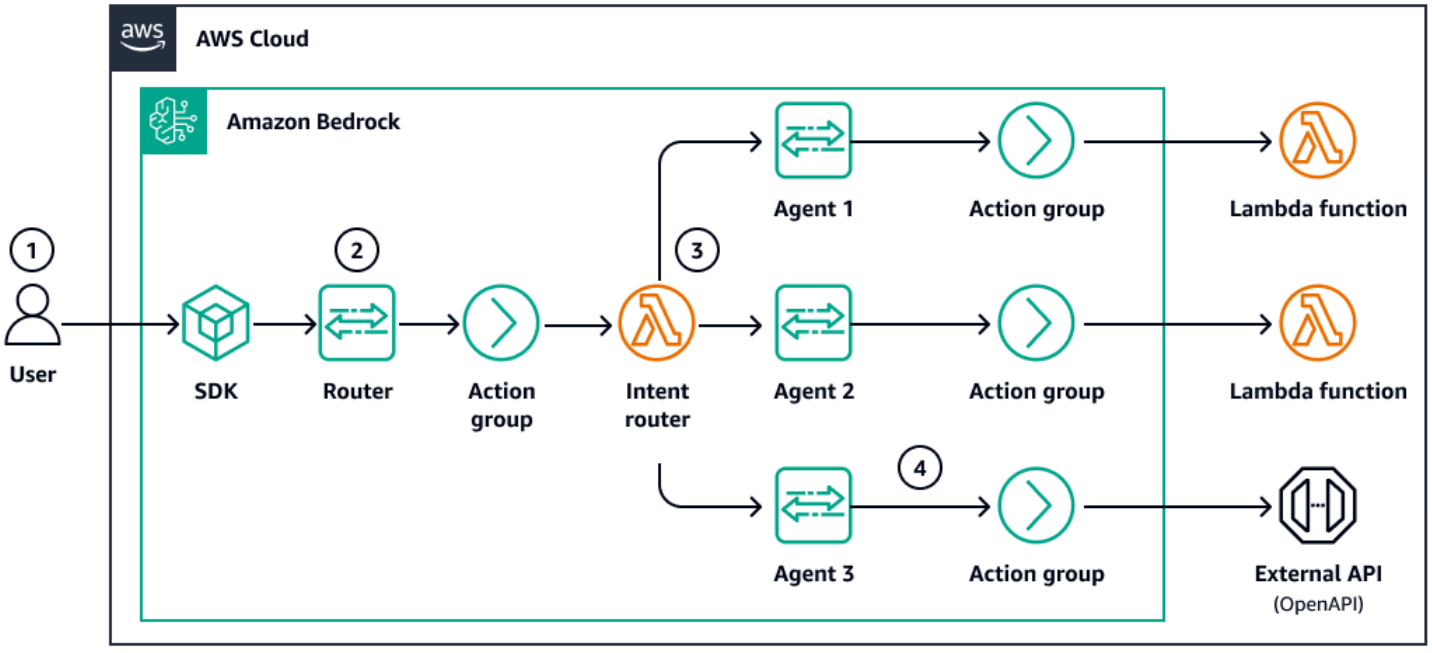
attraverso il linguaggio naturale. Il risultato è una forma di dispacciamento flessibile, semantica e adattiva.

Router agente

Questa architettura consente un ampio invio basato sugli intenti senza schemi o tipi di eventi predefiniti, ideale per input non strutturati e query complesse.

1. Un utente invia la richiesta «Potete aiutarmi a rivedere le condizioni del mio contratto?»
2. L'LLM lo interpreta come un'attività documentale legale.
3. L'agente indirizza l'attività a uno o più dei seguenti:
   - Modello di richiesta di revisione del contratto
   - Subagente per il ragionamento legale
   - Strumento di analisi dei documenti

Il diagramma seguente è un esempio di router agente:



1. Un utente invia una richiesta in linguaggio naturale tramite un SDK.
2. Un agente Amazon Bedrock utilizza un LLM per classificare l'attività (ad esempio legale, tecnica o di pianificazione).

3. L'agente indirizza dinamicamente l'attività attraverso un gruppo di azioni per richiamare l'agente richiesto:
   - Agente specifico del dominio
   - Catena di utensili specializzata
   - Configurazione personalizzata del prompt
4. Il gestore selezionato elabora l'operazione e restituisce una risposta personalizzata.

Da asporto

Laddove la distribuzione dinamica tradizionale utilizza le EventBridge regole di Amazon per il routing basate su attributi di eventi strutturati, il routing agentico utilizza il routing per classificare e LLMs indirizzare semanticamente le attività in base al significato e all'intento. Ciò amplia la flessibilità del sistema abilitando quanto segue:

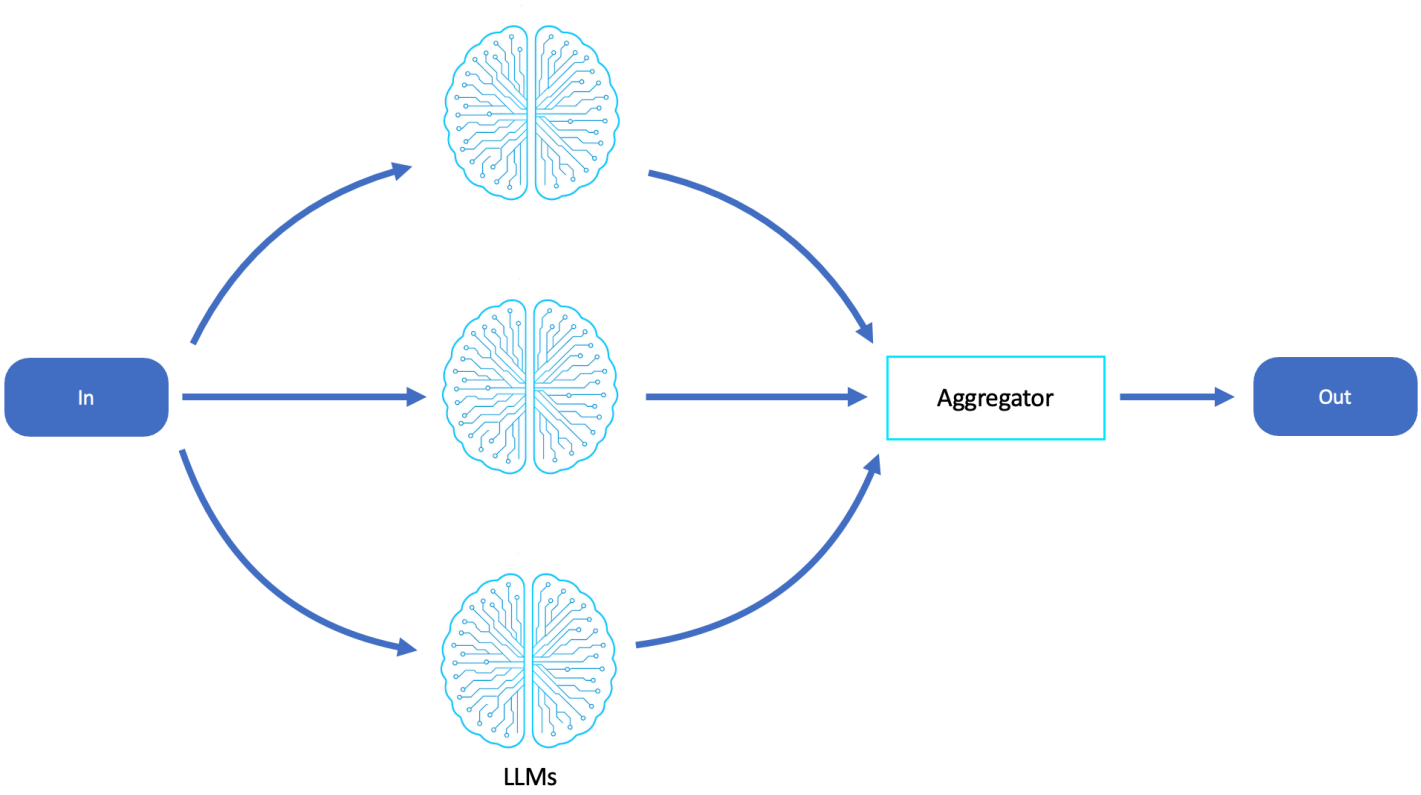
- Comprensione più ampia degli input
- Fallback intelligente e selezione degli strumenti
- Estensibilità naturale attraverso nuovi ruoli di agente o stili di prompt

L'agentic routing sostituisce le regole rigide con il dispatching cognitivo dinamico, che consente ai sistemi di evolversi con il linguaggio anziché con il codice.

Parallelizzazione e schemi di dispersione

Molte attività avanzate di ragionamento e generazione, come il riepilogo di documenti di grandi dimensioni, la valutazione di più percorsi di soluzione o il confronto di diverse prospettive, traggono vantaggio dall'esecuzione parallela dei prompt. I flussi di lavoro sequenziali tradizionali non sono all'altezza quando sono richieste scalabilità, reattività e tolleranza agli errori. Per ovviare a questo problema, la parallelizzazione basata su LLM può essere reinventata utilizzando un modello di scatter-gather basato sugli eventi, in cui le attività vengono assegnate dinamicamente ad agenti autonomi e i risultati sintetizzati in modo intelligente.

Il diagramma seguente è un esempio di flusso di lavoro di parallelizzazione LLM:



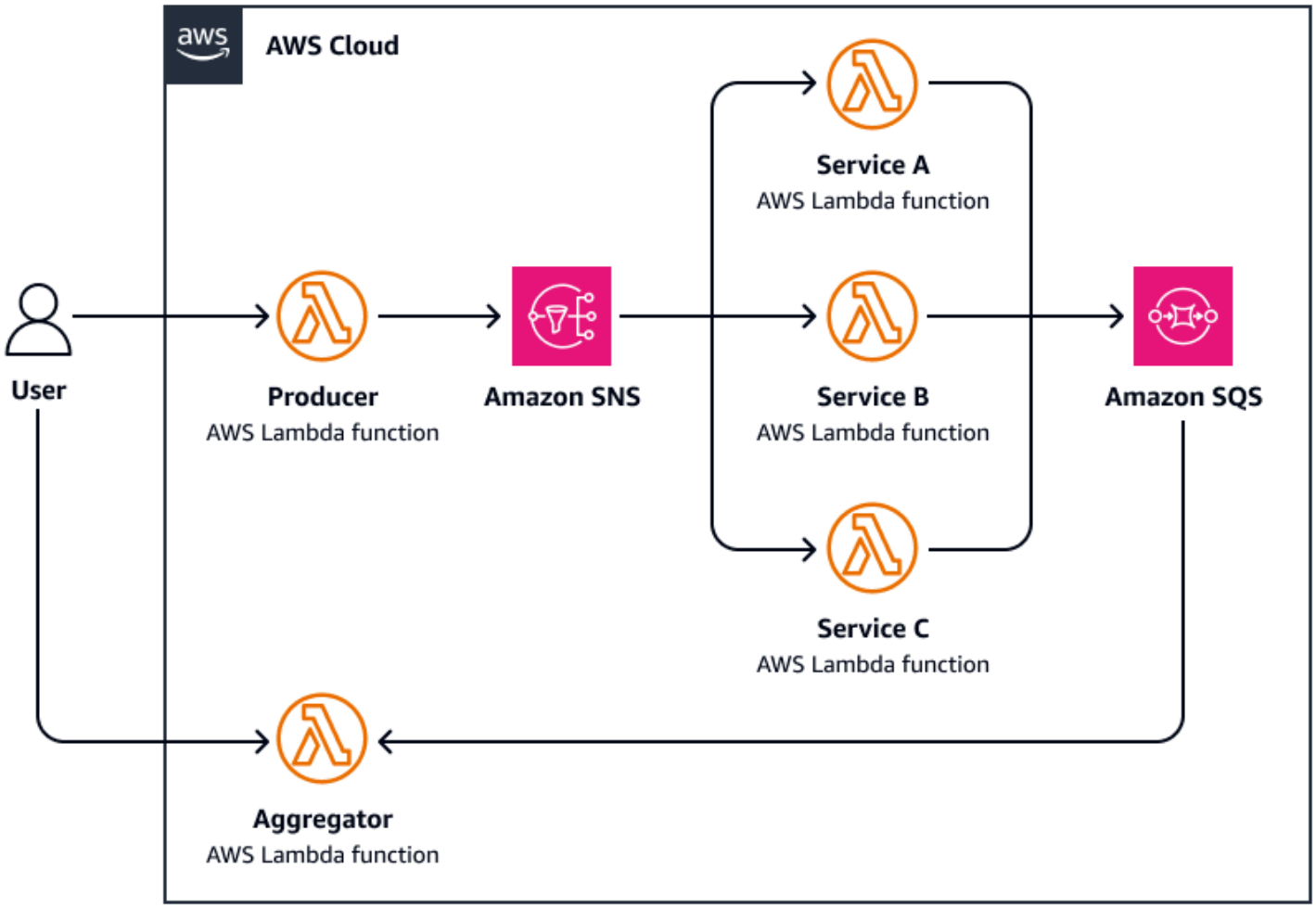
Scatter-gather

Nei sistemi distribuiti, un pattern scatter-gather invia le attività a più servizi o unità di elaborazione in parallelo, attende le loro risposte e quindi aggrega i risultati in un output consolidato. A differenza del fan-out, lo scatter-gather è coordinato perché prevede risposte e di solito applica la logica per combinare, confrontare e selezionare i risultati.

Le implementazioni comuni per la parallelizzazione e lo scatter-gather includono quanto segue:

- AWS Step Functions mappare uno stato per l'esecuzione parallela di attività
- AWS Lambda con concorrenza, coordinando i risultati di più funzioni richiamate
- Amazon EventBridge con flussi di lavoro di correlazione IDs e aggregazione
- Pattern di controller personalizzato per gestire il fan-out e raccogliere risultati utilizzando Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB o code

Il diagramma seguente è un esempio di scatter-gather:



1. Un utente invia una richiesta a una funzione di coordinamento centrale che suddivide l'attività pubblicando messaggi paralleli su un argomento di Amazon Simple Notification Service (Amazon SNS).
2. Ogni messaggio include i metadati delle attività e viene indirizzato a un operatore specializzato. AWS Lambda
3. Ogni lavoratore elabora AWS Lambda in modo indipendente la sottoattività assegnata (ad esempio, interrogazione di un'API esterna, elaborazione di un documento e analisi dei dati).
4. I risultati vengono scritti su un livello di storage comune, come Amazon Simple Queue Service (Amazon SQS).
5. La funzione di aggregazione attende il completamento di tutte le risposte, quindi esegue le seguenti operazioni:
   - Raccoglie e aggrega i risultati (ad esempio, unisce i riepiloghi, seleziona le migliori corrispondenze)

- Invia una risposta finale o attiva un flusso di lavoro a valle

I casi d'uso più comuni dei pattern scatter-gather includono i seguenti:

- Ricerca federata
- Motori di confronto dei prezzi
- Analisi aggregata dei dati
- Inferenza multimodello

Parallelizzazione basata su LLM (cognizione dispersa e raccolta)

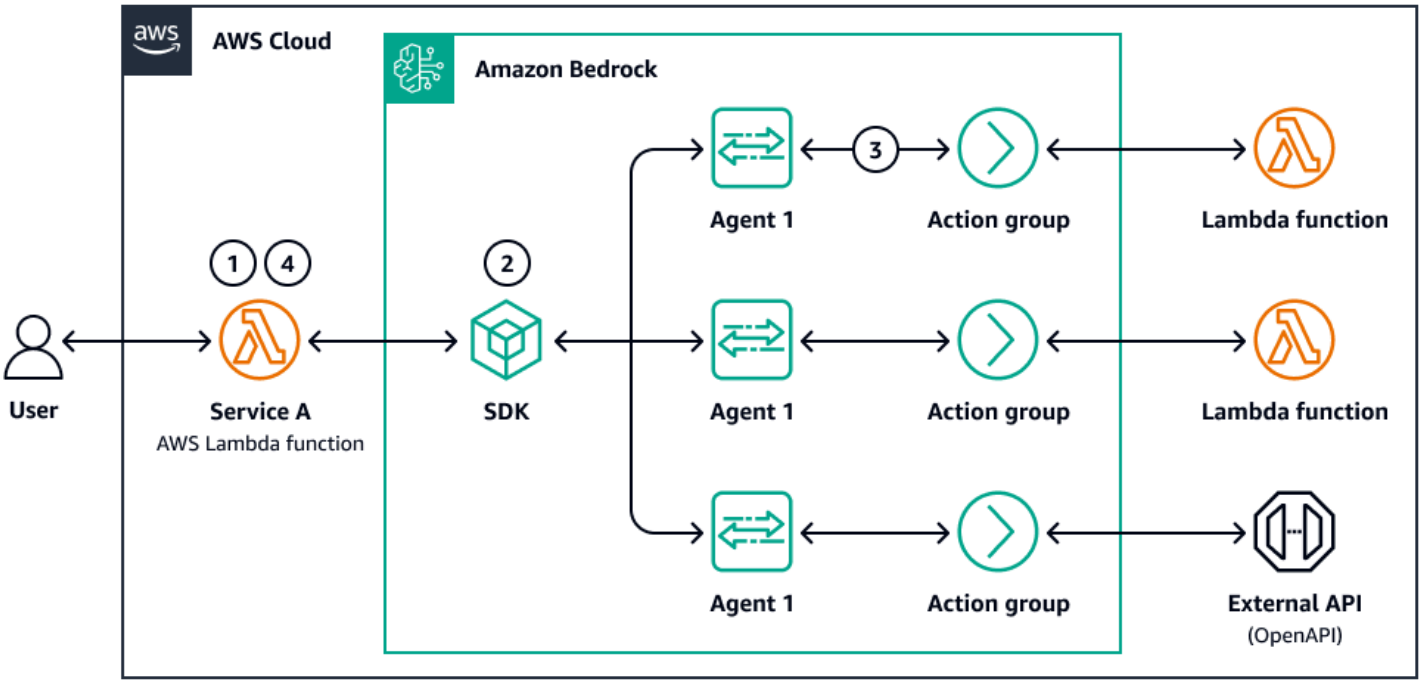
Nei sistemi agentici, la parallelizzazione rispecchia da vicino lo scatter-gather distribuendo le sottoattività su più chiamate o agenti LLM, ciascuno dei quali affronta in modo indipendente una parte del problema. I risultati restituiti vengono raccolti e sintetizzati da un processo di aggregazione, che spesso è un altro LLM o un altro agente di controllo.

Parallelizzazione degli agenti

1. Un agente invia una richiesta «Riepiloga le informazioni raccolte in questi 10 report».
2. Suddivide i report in 10 attività di riepilogo LLM parallele.
3. Quando restituisce tutti i riepiloghi, l'agente esegue le seguenti operazioni:
   - Aggrega i riepiloghi in un briefing unificato
   - Identifica temi o contraddizioni
   - Invia l'output sintetizzato all'utente

Questo flusso di lavoro agentico consente un ragionamento parallelo scalabile, modulare e adattivo. È ideale per i casi d'uso che richiedono un throughput cognitivo elevato.

Il diagramma seguente è un esempio di parallelizzazione degli agenti:



1. Un utente invia una query o un set di documenti composto da più parti.
2. Un controller AWS Lambda o una funzione Step distribuisce le sottoattività. Ogni attività richiama una chiamata o un subagente Amazon Bedrock LLM con il proprio prompt.
3. Quando le chiamate e le sottoattività sono complete, i risultati vengono archiviati (ad esempio, in Amazon S3 o nell'archivio di memoria) e una fase di aggregazione unisce, confronta o filtra gli output.
4. Il sistema restituisce la risposta finale all'utente o all'agente downstream.

Questo sistema dispone di un ciclo di ragionamento distribuito con tracciabilità, tolleranza ai guasti e logica opzionale di ponderazione o selezione dei risultati.

Take-away

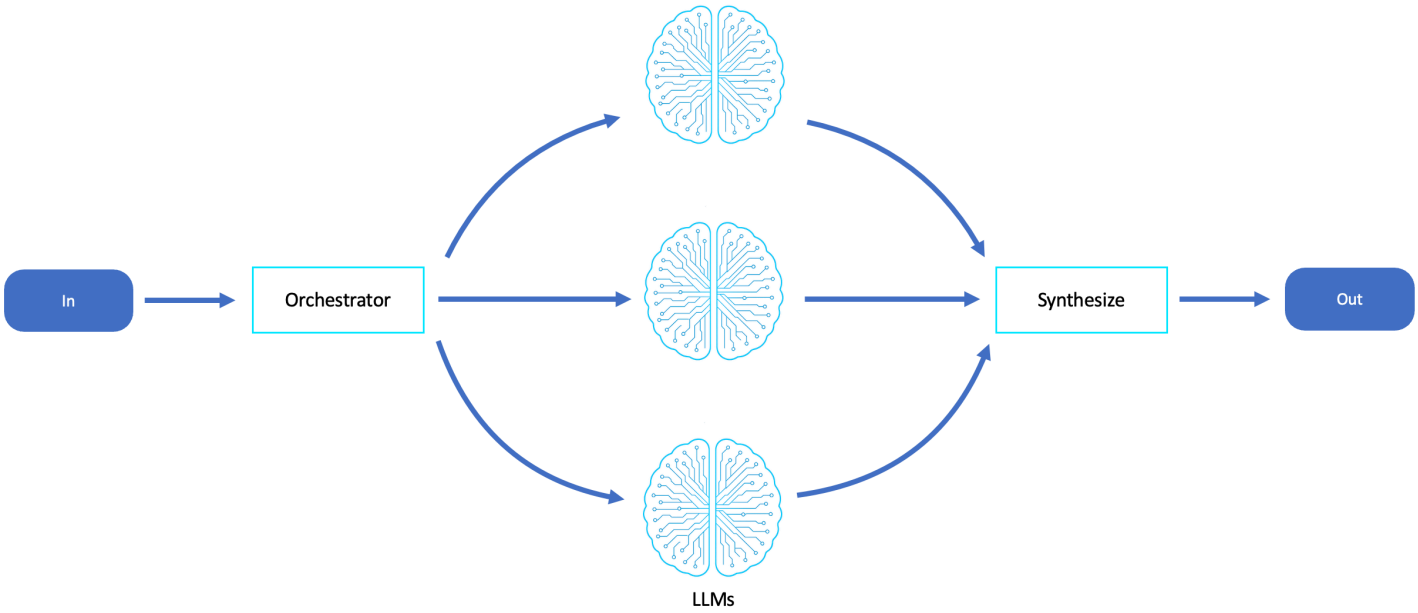
La parallelizzazione agentica utilizza modelli scatter-gather per distribuire le attività LLM, abilitando l'elaborazione parallela e la sintesi intelligente dei risultati.

Schemi di orchestrazione Saga

Man mano che i flussi di lavoro guidati da tali flussi di lavoro LLMs diventano sempre più complessi e comprendono catene di richieste, fasi di elaborazione dei dati, invocazioni di strumenti e

collaborazione tra agenti, la necessità di un'orchestrazione intelligente diventa essenziale. Anziché fare affidamento su script strettamente associati o flussi di esecuzione statici predeterminati, questi flussi di lavoro possono essere implementati come modelli di orchestrazione basati sugli eventi, consentendo ai sistemi basati su LLM di coordinare, monitorare e adattare dinamicamente le attività in più fasi tra agenti autonomi.

Il diagramma seguente è un esempio di orchestratore:



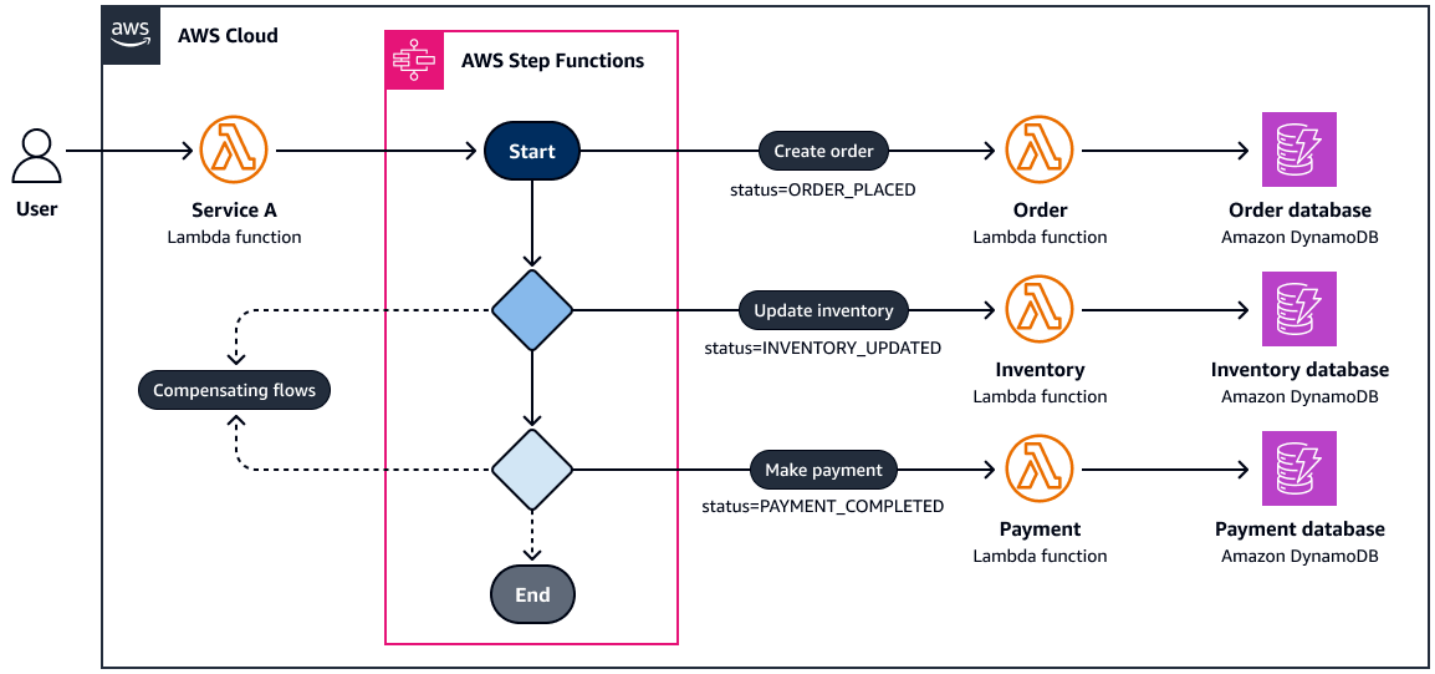
Orchestrazione di eventi

Nei sistemi distribuiti tradizionali, l'orchestrazione degli eventi si riferisce a uno schema in cui un coordinatore centrale gestisce un flusso di lavoro complesso indirizzando esplicitamente il flusso di controllo su più servizi o attività. A differenza della coreografia degli eventi (in cui ogni servizio reagisce in modo indipendente), l'orchestrazione fornisce logica, visibilità e controllo centralizzati sull'intero processo.

Questa operazione viene in genere implementata utilizzando i seguenti strumenti:

- AWS Step Functions— Definizione ed esecuzione di flussi di lavoro basati sullo stato
- AWS Lambda— Esegui attività discrete all'interno del flusso orchestrato
- Amazon SQS o Amazon EventBridge: attiva passaggi o risposte asincroni

Il diagramma seguente è un esempio di orchestrazione di saga:



Un AWS Step Functions flusso di lavoro gestisce un processo di ordinazione dei clienti:

1. Crea ordine (AWS Lambda)
2. Aggiorna l'inventario (AWS Lambda)
3. Effettua il pagamento (AWS Lambda)

L'orchestratore coordina ogni fase gestendo nuovi tentativi, branch paralleli, timeout e guasti.

Sistema di agenti basato sui ruoli (orchestratore)

Nei sistemi agentici, il pattern di orchestrazione rispecchia l'orchestrazione degli eventi ma distribuisce la logica tra più agenti di ragionamento, ciascuno con un ruolo o una specializzazione definiti. Un agente di orchestrazione centrale interpreta l'attività complessiva, la scompone in sottoattività e le delega agli agenti di lavoro, ciascuno ottimizzato per un particolare dominio (ad esempio, ricerca, codifica, riepilogo, revisione).

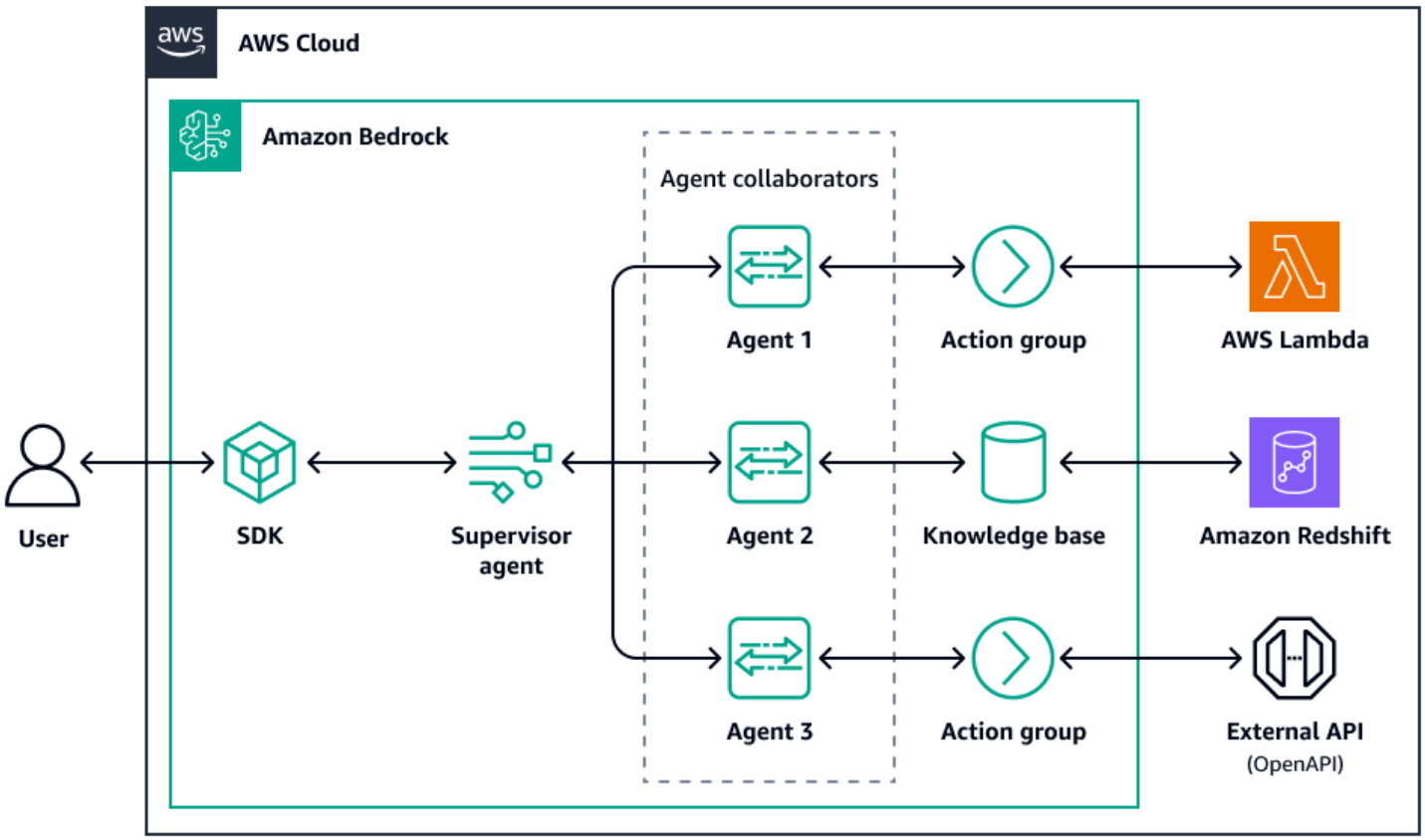
Supervisore

1. Un utente invia la domanda «Crea un brief del progetto e riepiloga i primi 5 concorrenti».
2. L'agente orchestrator esegue le seguenti operazioni:
   - Assegna a un agente di ricerca il compito di trovare i dati dei concorrenti

- Invia i risultati non elaborati a un agente di riepilogo
- Trasmette i risultati a un agente addetto alla redazione di brevi istruzioni
- Compila l'output finale per l'utente

Ogni agente opera in modo indipendente, ma l'orchestratore coordina le attività. È come una funzione Lambda che gestisce le attività del flusso di lavoro.

Il diagramma seguente mostra un esempio di supervisore:



1. Un utente invia un'attività a un agente supervisore di Amazon Bedrock.
2. L'agente supervisore analizza la richiesta in sottoattività per ogni collaboratore agente.
3. Ogni sottoattività viene assegnata a un agente collaboratore con istruzioni o toolchain specifici per il ruolo.
4. Gli agenti di lavoro chiamano strumenti esterni APIs o utilizzano un gruppo di azione.
5. Ogni agente di lavoro restituisce l'output in un formato strutturato.
6. Quando tutti i lavoratori restituiscono i risultati, il supervisore valuta, sintetizza e restituisce la risposta finale.

Questa struttura consente modularità, adattabilità e introspezione in flussi di lavoro complessi con agenti in più fasi.

Cose da asporto

Laddove l'orchestrazione degli eventi utilizza il controllo centralizzato (ad esempio AWS Step Functions) per dirigere l'esecuzione del servizio, i sistemi di agenti basati sui ruoli utilizzano un agente di orchestrazione basato su LLM per ragionare sull'obiettivo, delegare le attività secondarie agli agenti di lavoro e sintetizzare l'output finale.

In entrambi i paradigmi, l'orchestratore esegue le seguenti operazioni:

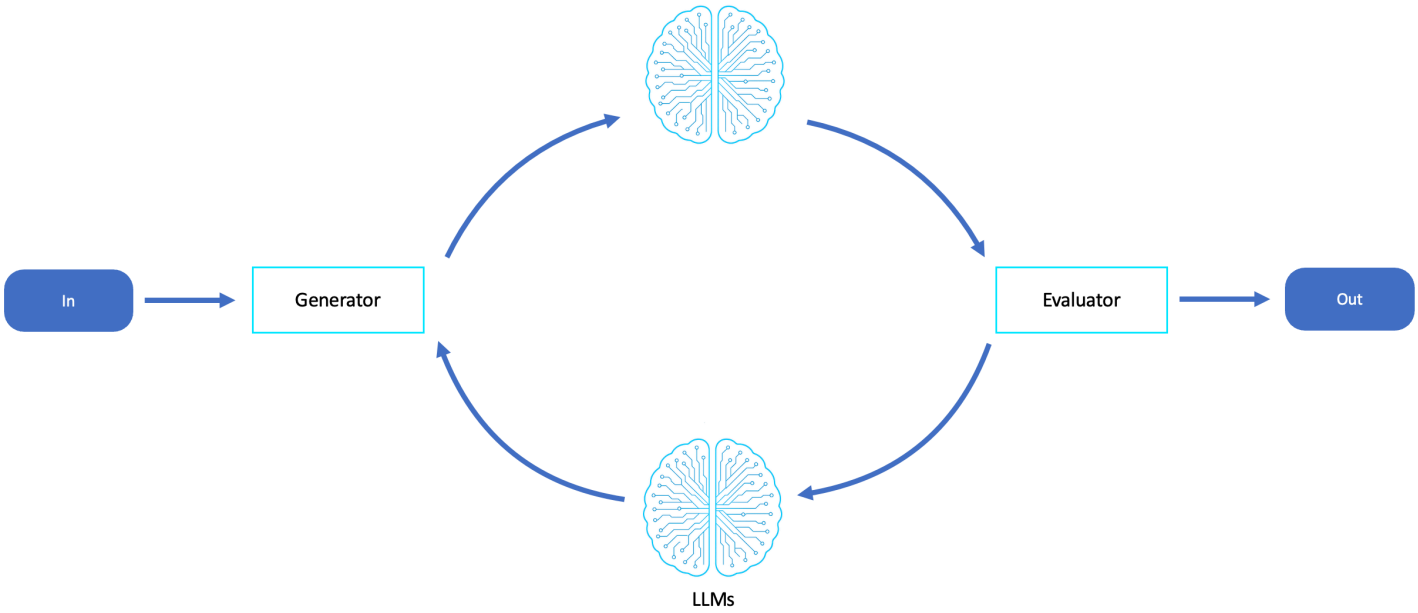
- Mantiene il contesto e il flusso di esecuzione
- Gestisce la ramificazione, il sequenziamento e la gestione degli errori
- Produce un risultato unificato a partire da componenti distribuiti

L'orchestrazione agentica, tuttavia, aggiunge ragionamento, adattabilità e delega semantica. Ciò lo rende adatto a compiti aperti, ambigui e in continua evoluzione.

Evaluator Reflect-Refine Loop Patterns

Attività come la generazione di codice, il riepilogo o il processo decisionale autonomo traggono grandi vantaggi dal feedback in fase di esecuzione, che consente al sistema di evolversi attraverso l'osservazione e il perfezionamento. Per renderlo operativo, il ciclo reflect-refine può essere implementato come un circuito di controllo del feedback basato sugli eventi, un modello ispirato all'ingegneria dei sistemi, adattato per flussi di lavoro autonomi e intelligenti.

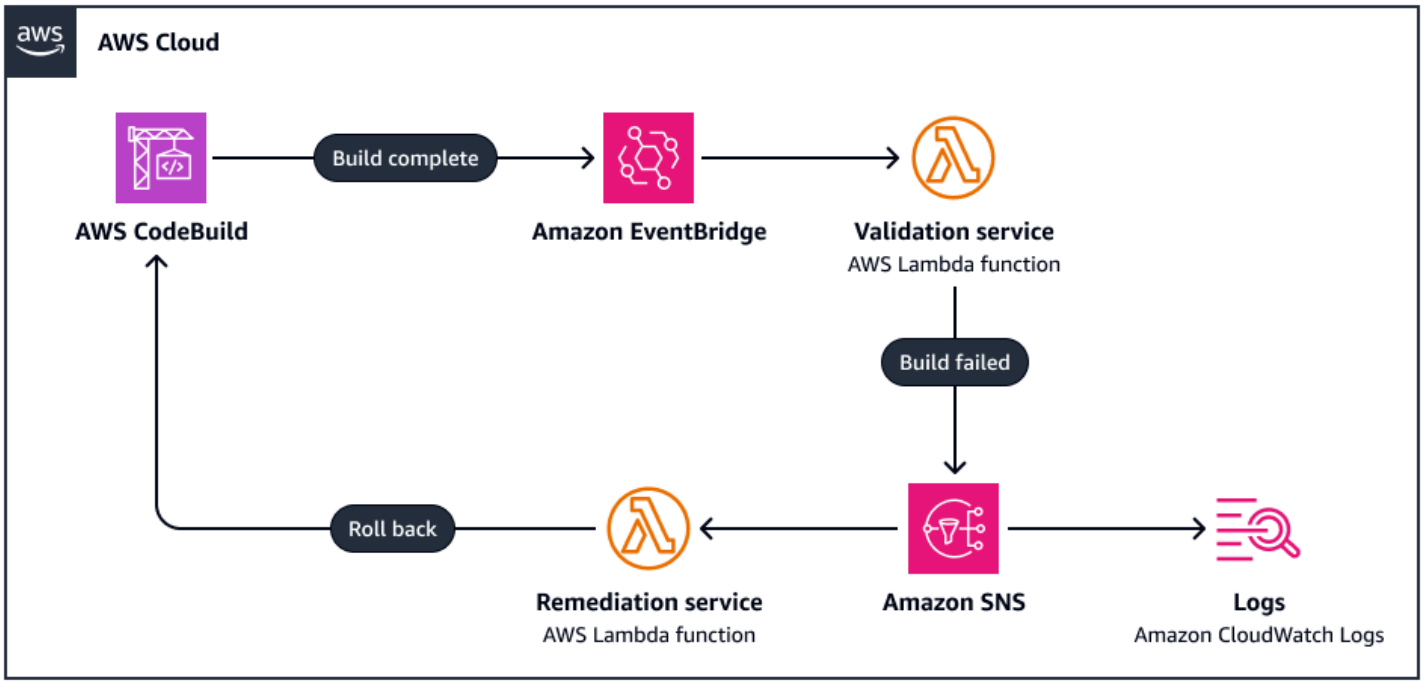
Il diagramma seguente è un esempio di ciclo di feedback reflect-refine di Evaluator:



Circuito di controllo del feedback

Un circuito di controllo del feedback è uno schema che monitora i propri output e comportamenti, li valuta in base a criteri definiti o allo stato desiderato e quindi regola le proprie azioni di conseguenza. Questa architettura si ispira alla teoria del controllo ed è fondamentale in domini come le pipeline di automazione, integrazione continua e distribuzione continua (CI/CD) e le operazioni di apprendimento automatico.

Il diagramma seguente è un esempio di circuito di controllo del feedback:



1. Una pipeline di distribuzione emette un evento BuildComplete.
2. L'evento attiva un processo di test o valutazione automatizzato che convalida la build.
3. Se la convalida fallisce (ad esempio, a causa di test non riusciti, problemi di sicurezza o violazione delle politiche), il sistema:
   - Emette un evento BuildComplete
   - Registra il problema o invia una notifica
   - Attiva un'azione riparativa o correttiva, ad esempio il rollback, l'applicazione di una patch o un nuovo tentativo

Il ciclo continua finché non produce un risultato o un'escalation accettabile o fino a quando non si verifica un timeout. Questo modello viene comunemente utilizzato per quanto segue:

- EventBridge Regole di Amazon per indirizzare gli eventi alle attività di valutazione o riparazione
- AWS Step Functions per una logica di ripetizione iterativa e una ramificazione dei risultati della valutazione
- Amazon Simple Notification Service (Amazon SNS) o allarmi Amazon per i trigger di CloudWatch feedback e gli avvisi
- AWS Lambda funzioni o lavoratori containerizzati per applicare azioni correttive

Circuito di controllo del feedback (valutatore)

Il flusso di lavoro di un valutatore è un ciclo di feedback cognitivo alimentato dai nostri agenti di LLMs ragionamento. Il processo è costituito dai seguenti elementi:

1. Un agente generatore o LLM produce un output (ad esempio, un piano, una risposta o una bozza).
2. Un agente valutatore esamina il risultato utilizzando un prompt di critica o una rubrica di valutazione.
3. In base al feedback, l'agente originale o un nuovo agente di ottimizzazione rivede l'output.

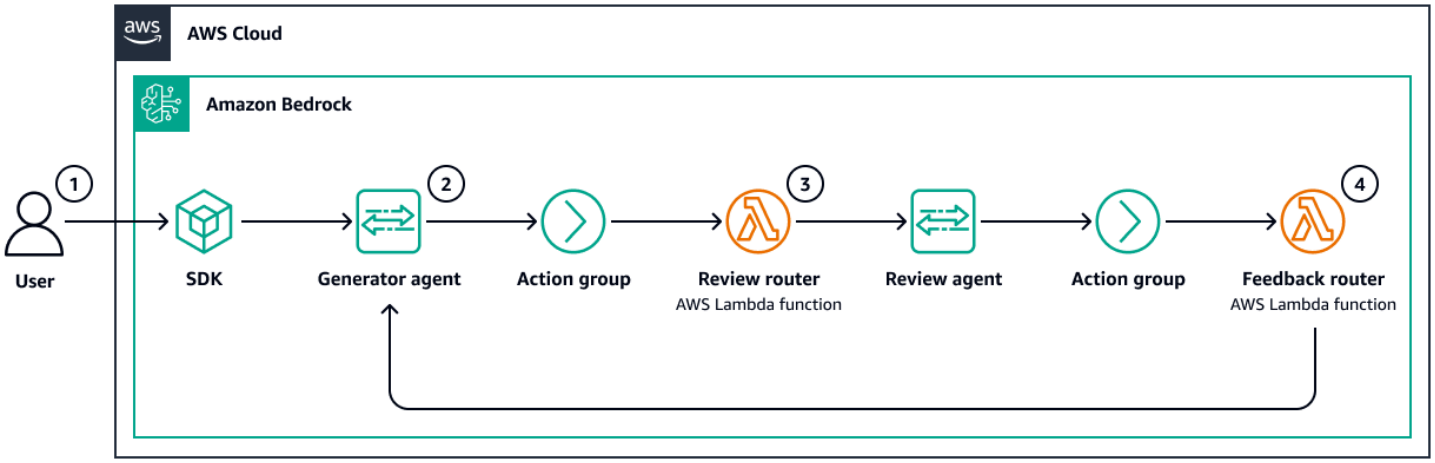
Il ciclo si ripete finché il risultato non soddisfa una serie di criteri, non viene approvato o raggiunge un limite di tentativi.

Valutatore

1. Un utente chiede a un agente di scrivere un riepilogo delle politiche.
2. L'agente generatore lo redige.
3. Un agente valutatore verifica la copertura, il tono e la correttezza legale.
4. Se la risposta è inadeguata, viene perfezionata e inviata nuovamente fino alla convergenza del ciclo di feedback.

Ciò consente l'autovalutazione, il perfezionamento iterativo e il controllo adattivo dell'output, il tutto senza l'intervento umano.

Il diagramma seguente è un esempio di circuito di controllo del feedback (valutatore):



1. Un utente esegue un'attività (ad esempio, redigere una strategia aziendale).
2. Un agente Amazon Bedrock genera una bozza iniziale utilizzando un LLM.
3. Un secondo agente (o un prompt di follow-up) esegue una valutazione strutturata (ad esempio, «valuta questo risultato in base a chiarezza, completezza e tono»).
4. Se la valutazione scende al di sotto di una soglia, la risposta viene modificata da:
   - Reinvocare il generatore con una critica incorporata
   - Invio del feedback a un agente di raffinazione specializzato
   - Iterazione fino al raggiungimento di una risposta accettabile

Componenti opzionali come AWS Lambda i controller AWS Step Functions possono gestire soglie di feedback, nuovi tentativi e strategie di fallback.

Cose da asporto

Laddove i tradizionali circuiti di controllo del feedback utilizzano eventi, metriche e logica di correzione per convalidare e regolare il comportamento del sistema, i loop di valutazione agentica utilizzano agenti di ragionamento per valutare, riflettere e rivedere l'output in modo dinamico.

In entrambi i paradigmi:

- L'output viene valutato dopo essere stato generato
- Le azioni correttive o di perfezionamento vengono attivate in base al feedback
- Il sistema si adatta continuamente alla qualità o all'obiettivo prefissato

La versione agentica trasforma la convalida statica in riflessione semantica, permettendo agli agenti di automiglioramento di valutare la propria efficacia.

Progettazione di flussi di lavoro agentici su AWS

Ogni modello in questa guida può essere creato utilizzando. Servizi AWS Gli agenti Amazon Bedrock forniscono canali di orchestrazione, accesso ai dati e interazione.

Componente	Servizio AWS	Scopo

Ragionamento LLM	Amazon Bedrock	Logica degli agenti, pianifica zione, utilizzo degli strumenti
Esecuzione degli strumenti	AWS Lambda, Amazon ECS, Amazon SageMaker	Ospita strumenti esterni per agenti
Memoria e RAG	Base di conoscenze Amazon Bedrock, Amazon S3, OpenSearch	Memoria persistente e semantica
Orchestrazione	AWS Step Functions	Coordinamento di attività e agenti in più fasi
Routing degli eventi	Amazon EventBridge, Amazon SQS	Messaggistica interagente disaccoppiata
Interfaccia utente	Amazon API Gateway AWS AppSync, SDK	Punti di ingresso per applicazi oni o sistemi
Monitoraggio	Amazon CloudWatch AWS X-Ray, AWS Distro per OpenTelemetry	Osservabilità e introspezione degli agenti

Conclusioni

I modelli di flusso di lavoro agentici sono la prossima fase evolutiva delle architetture basate sugli eventi, in cui la logica aziendale non è definita staticamente ma ragionata dinamicamente attraverso l'utilizzo di una cognizione migliorata del modello di linguaggio (LLM). Combinando le tradizionali primitive native del cloud con flussi di lavoro LLM e modelli di progettazione degli agenti, le organizzazioni possono creare sistemi adattivi, intelligenti e modulari che rispondono con uno scopo e imparano dall'esperienza.

In questi modelli, Amazon Bedrock è la porta di accesso alla cognizione agentica, che consente agli agenti basati su LLM di accedere ai flussi di lavoro degli eventi, interagire con strumenti e memoria e fornire risultati strutturati, tracciabili e allineati.

Durante la progettazione e l'implementazione di sistemi agentici, questi modelli di flusso di lavoro forniscono modelli per la creazione di architetture AI autonome e componibili. Questi sistemi si basano su best practice serverless e potenziati con modelli di base intelligenti.

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 tuo database Oracle locale a Oracle su un'istanza EC2 in. Cloud AWS
- Trasferire (eseguire il rehosting a livello hypervisor): trasferisci l'infrastruttura sul cloud senza acquistare nuovo hardware, riscrivere le applicazioni o modificare le operazioni esistenti. Esegui la migrazione dei server da una piattaforma locale a un servizio cloud per la stessa piattaforma. Esempio: migra un'applicazione 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 frequentemente per un problema ricorrente in cui la soluzione è controproducente, inefficace o meno efficace di un'alternativa.

controllo delle applicazioni

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

portfolio di applicazioni

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

intelligenza artificiale (IA)

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

operazioni di intelligenza artificiale (AIOps)

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

crittografia asimmetrica

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

atomicità, consistenza, isolamento, durabilità (ACID)

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

Controllo degli accessi basato su attributi (ABAC)

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

fonte di dati autorevole

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

Zona di disponibilità

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

AWS Cloud Adoption Framework (CAF)AWS

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

AWS Workload Qualification Framework (WQF)AWS

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

B

bot difettoso

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

BCP

Vedi la pianificazione della continuità operativa.

grafico comportamentale

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

sistema big-endian

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

Classificazione binaria

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

filtro Bloom

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

implementazione blu/verde

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

bot

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

botnet

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

ramo

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

accesso break-glass

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

strategia brownfield

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

cache del buffer

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

capacità di business

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

pianificazione della continuità operativa (BCP)

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

C

CAF

Vedi Cloud Adoption AWS Framework.

implementazione canaria

Il rilascio lento e incrementale di una versione agli utenti finali. Quando sei sicuro, 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, come dati in tempo reale o dati di traduzione recenti. Questi dati richiedono in genere un livello o una classe di storage ad alte prestazioni per fornire risposte rapide alle query.

hotfix

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

periodo di hypercare

Subito dopo la conversione, il periodo di tempo in cui un team di migrazione gestisce e monitora le applicazioni migrate nel cloud per risolvere eventuali problemi. In genere, questo periodo dura

da 1 a 4 giorni. Al termine del periodo di hypercare, il team addetto alla migrazione in genere trasferisce la responsabilità delle applicazioni al team addetto alle operazioni cloud.

I

IaC

Vedi l'infrastruttura come codice.

Policy basata su identità

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

applicazione inattiva

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

IIoT

Vedi Industrial Internet of Things.

infrastruttura immutabile

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

VPC in ingresso (ingress)

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

migrazione incrementale

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

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

Industria 4.0

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

infrastruttura

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

infrastruttura come codice (IaC)

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

IIoInternet delle cose industriale (T)

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

VPC di ispezione

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

Internet of Things (IoT)

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

interpretabilità

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

IoT

Vedi Internet of Things.

libreria di informazioni IT (ITIL)

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

gestione dei servizi IT (ITSM)

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

ITIL

Vedi la libreria di informazioni IT.

ITSM

Vedi Gestione dei servizi IT.

L

controllo degli accessi basato su etichette (LBAC)

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

zona di destinazione

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

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

modello linguistico di grandi dimensioni (LLM)

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

migrazione su larga scala

Una migrazione di 300 o più server.

BIANCO

Vedi controllo degli accessi basato su etichette.

Privilegio minimo

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

eseguire il rehosting (lift and shift)

Vedi 7 R.

sistema little-endian

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

LLM

Vedi modello linguistico di grandi dimensioni.

ambienti inferiori

Vedi ambiente.

M

machine learning (ML)

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

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

ramo principale

Vedi filiale.

malware

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

servizi gestiti

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

sistema di esecuzione della produzione (MES)

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

MAP

Vedi Migration Acceleration Program.

meccanismo

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

account membro

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

MEH

Vedi sistema di esecuzione della produzione.

Message Queuing Telemetry Transport (MQTT)

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

microservizio

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

architettura di microservizi

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

Programma di accelerazione della migrazione (MAP)

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

migrazione su larga scala

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

fabbrica di migrazione

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

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

metadati di migrazione

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

modello di migrazione

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

Valutazione del portfolio di migrazione (MPA)

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

valutazione della preparazione alla migrazione (MRA)

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

strategia di migrazione

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

ML

Vedi machine learning.

modernizzazione

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

valutazione della preparazione alla modernizzazione

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

applicazioni monolitiche (monoliti)

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

MAPPA

Vedi Migration Portfolio Assessment.

MQTT

Vedi Message Queuing Telemetry Transport.

classificazione multiclasse

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

infrastruttura mutabile

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

O

OAC

Vedi Origin Access Control.

QUERCIA

Vedi Origin Access Identity.

OCM

Vedi gestione delle modifiche organizzative.

migrazione offline

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

OI

Vedi l'integrazione delle operazioni.

OLA

Vedi accordo a livello operativo.

migrazione online

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

OPC-UA

Vedi Open Process Communications - Unified Architecture.

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

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

accordo a livello operativo (OLA)

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

revisione della prontezza operativa (ORR)

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

tecnologia operativa (OT)

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

integrazione delle operazioni (OI)

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

trail organizzativo

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

gestione del cambiamento organizzativo (OCM)

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

persone, a causa della velocità di cambiamento richiesta nei progetti di adozione del cloud. Per ulteriori informazioni, consultare la Guida OCM.

controllo dell'accesso all'origine (OAC)

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

identità di accesso origine (OAI)

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

ORR

Vedi la revisione della prontezza operativa.

NON

Vedi la tecnologia operativa.

VPC in uscita (egress)

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

P

limite delle autorizzazioni

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

informazioni di identificazione personale (PII)

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

Informazioni che consentono l'identificazione personale degli utenti

Visualizza le informazioni di identificazione personale.

playbook

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

PLC

Vedi controllore logico programmabile.

PLM

Vedi la gestione del ciclo di vita del prodotto.

policy

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

persistenza poliglotta

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

valutazione del portfolio

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

predicate

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

predicato pushdown

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

controllo preventivo

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

principale

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

privacy fin dalla progettazione

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

zone ospitate private

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

controllo proattivo

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

gestione del ciclo di vita del prodotto (PLM)

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

Ambiente di produzione

Vedi ambiente.

controllore logico programmabile (PLC)

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

concatenamento rapido

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

pseudonimizzazione

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

publish/subscribe (pub/sub)

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

Q

Piano di query

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

regressione del piano di query

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

R

Matrice RACI

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

RAG

Vedi Retrieval Augmented Generation.

ransomware

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

Matrice RASCI

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

RCAC

Vedi controllo dell'accesso a righe e colonne.

replica di lettura

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

riprogettare

Vedi 7 Rs.

obiettivo del punto di ripristino (RPO)

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

obiettivo del tempo di ripristino (RTO)

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

rifattorizzare

Vedi 7 R.

Region

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

regressione

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

riospitare

Vedi 7 R.

rilascio

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

trasferisco

Vedi 7 Rs.

ripiattaforma

Vedi 7 Rs.

riacquisto

Vedi 7 Rs.

resilienza

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

policy basata su risorse

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

matrice di assegnazione di responsabilità (RACI)

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

controllo reattivo

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

retain

Vedi 7 R.

andare in pensione

Vedi 7 Rs.

Retrieval Augmented Generation (RAG)

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

rotazione

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

controllo dell'accesso a righe e colonne (RCAC)

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

RPO

Vedi obiettivo del punto di ripristino.

VERSO

Vedi obiettivo del tempo di ripristino.

runbook

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

S

SAML 2.0

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

SCADA

Vedi controllo di supervisione e acquisizione dati.

SCP

Vedi la politica di controllo del servizio.

Secret

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

sicurezza fin dalla progettazione

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

controllo di sicurezza

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

rafforzamento della sicurezza

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

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

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

automazione della risposta alla sicurezza

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

Crittografia lato server

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

Policy di controllo dei servizi (SCP)

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

endpoint del servizio

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

accordo sul livello di servizio (SLA)

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

indicatore del livello di servizio (SLI)

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

obiettivo a livello di servizio (SLO)

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

Modello di responsabilità condivisa

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

SIEM

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

punto di errore singolo (SPOF)

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

SLAM

Vedi il contratto sul livello di servizio.

SLI

Vedi l'indicatore del livello di servizio.

LENTA

Vedi obiettivo del livello di servizio.

split-and-seed modello

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

SPOF

Vedi punto di errore singolo.

schema a stella

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

modello del fico strangolatore

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

sottorete

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

controllo di supervisione e acquisizione dati (SCADA)

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

crittografia simmetrica

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

test sintetici

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

prompt di sistema

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

T

tag

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

variabile di destinazione

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

elenco di attività

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

ambiente di test

Vedi ambiente.

training

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

Transit Gateway

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

flusso di lavoro basato su trunk

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

Accesso attendibile

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

regolazione

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

team da due pizze

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

U

incertezza

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

compiti indifferenziati

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

ambienti superiori

Vedi ambiente.

V

vacuum

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

controllo delle versioni

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

Peering VPC

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

vulnerabilità

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

W

cache calda

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

dati caldi

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

funzione finestra

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

Carico di lavoro

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

flusso di lavoro

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

VERME

Vedi scrivere una volta, leggere molti.

WQF

Vedi AWS Workload Qualification Framework.

scrivi una volta, leggi molte (WORM)

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

Z

exploit zero-day

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

vulnerabilità zero-day

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

prompt zero-shot

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

applicazione zombie

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

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