Les fondements de l'IA agentique sur AWS

AWS Directives prescriptives

AWS Directives prescriptives: Les fondements de l'IA agentique sur AWS

Table of Contents

Les fondements de l'IA agentique sur AWS

Aaron Sempf, Amazon Web Services

Juillet 2025 (historique du document)

Dans un monde où les systèmes sont de plus en plus intelligents, distribués et autonomes, le concept d'agent, c'est-à-dire une entité capable de percevoir son environnement, de raisonner sur son état et d'agir avec intention, est devenu fondamental. Les agents ne sont pas simplement des programmes qui exécutent des instructions ; ce sont des entités orientées vers des objectifs et sensibles au contexte qui prennent des décisions au nom des utilisateurs, des systèmes ou des organisations. Leur émergence reflète un changement dans la façon dont vous concevez et concevez les logiciels : le passage de la logique procédurale et de l'automatisation réactive à des systèmes qui fonctionnent de manière autonome et ciblée.

À l'intersection de l'IA, des systèmes distribués et du génie logiciel se trouve un puissant paradigme connu sous le nom d'IA agentique. Cette nouvelle génération de systèmes intelligents est composée d'agents logiciels capables de comportements adaptatifs, de coordination complexe et de prise de décision déléguée.

Ce guide présente les principes qui définissent les agents logiciels modernes et décrit leur évolution vers l'IA agentique. Pour expliquer ce changement, le guide fournit le contexte conceptuel, puis retrace l'évolution des agents logiciels vers l'IA agentique :

- L'introduction aux agents logiciels définit les agents logiciels, les compare aux composants logiciels traditionnels et présente les caractéristiques essentielles qui différencient le comportement agentique de l'automatisation traditionnelle en s'appuyant sur des frameworks établis.
- L'objectif des agents logiciels consiste à examiner pourquoi les agents logiciels existent, quels rôles ils remplissent, quels problèmes ils résolvent, et comment ils permettent une délégation intelligente, réduisent la charge cognitive et favorisent le comportement adaptatif dans les environnements dynamiques.
- L'évolution des agents logiciels retrace les jalons intellectuels et technologiques qui ont façonné les agents logiciels, depuis les premiers concepts d'autonomie et de concurrence jusqu'à l'émergence de systèmes multi-agents et d'architectures d'agents formelles, aboutissant à la convergence avec l'IA générative.
- Les agents logiciels associés à l'IA agentic présentent l'IA agentic comme l'aboutissement de décennies de progrès qui combinent des modèles d'agents distribués avec des modèles de base,

des calculs sans serveur et des protocoles d'orchestration. Cette section décrit comment cette convergence permet de créer une nouvelle génération d'agents intelligents utilisant des outils qui fonctionnent de manière autonome, asynchronique et dotés d'une véritable agence à grande échelle.

Public visé

Ce guide est conçu pour les architectes, les développeurs et les leaders technologiques qui souhaitent comprendre l'histoire, les principaux concepts et l'évolution des agents logiciels vers l'IA agentique avant d'adopter cette technologie pour les solutions cloud modernes. AWS

Objectifs

L'adoption d'architectures agentiques aide les entreprises à :

- Accélérez le délai de rentabilisation : automatisez et adaptez le travail lié aux connaissances, et réduisez les efforts manuels et les temps de latence.
- Améliorez l'engagement des clients : fournissez des assistants intelligents dans tous les domaines.
- Réduisez les coûts opérationnels : automatisez les flux de décision qui nécessitaient auparavant une intervention humaine ou une supervision.
- Favorisez l'innovation et la différenciation : créez des produits intelligents qui s'adaptent, apprennent et sont compétitifs en temps réel.
- Modernisez les flux de travail existants : redéfinissez les scripts et les monolithes pour en faire des agents de raisonnement modulaires.

À propos de cette série de contenus

Ce guide fait partie d'un ensemble de publications qui fournissent des plans architecturaux et des conseils techniques pour la création d'agents logiciels pilotés par l'IA. AWS La série inclut les éléments suivants :

- Opérationnaliser l'IA agentique sur AWS
- Les fondements de l'IA agentique sur AWS (ce guide)
- Modèles et flux de travail d'IA agentic sur AWS
- Frameworks, protocoles et outils d'IA agentic sur AWS

- Création d'architectures sans serveur pour l'IA agentic sur AWS
- Création d'architectures multi-locataires pour l'IA agentic sur AWS

Pour plus d'informations sur cette série de contenus, consultez Agentic AI.

Présentation des agents logiciels

Le concept d'agents logiciels a évolué de manière significative depuis ses débuts dans les entités autonomes dans les années 1960 jusqu'à son exploration formelle au début des années 1990. Alors que les systèmes numériques deviennent de plus en plus complexes, qu'il s'agisse de scripts déterministes ou d'applications adaptatives et intelligentes, les agents logiciels sont devenus des éléments essentiels pour permettre un comportement autonome, sensible au contexte et axé sur les objectifs dans les systèmes informatiques. Dans le contexte des architectures natives du cloud et optimisées par l'IA, en particulier avec l'avènement de l'IA générative, des grands modèles linguistiques (LLMs) et des plateformes telles qu'Amazon Bedrock, les agents logiciels sont redéfinis selon de nouvelles perspectives de capacité et d'échelle.

Cette introduction s'inspire de l'ouvrage fondateur Software Agents : An Overview de Hyacinth S. Nwana (Nwana 1996). Il définit les agents logiciels, discute de leurs racines conceptuelles et étend la discussion dans un cadre contemporain afin de définir trois principes fondamentaux des agents logiciels modernes : autonomie, asynchronicité et agence. Ces principes distinguent les agents logiciels des autres types de services ou d'applications, et permettent à ces agents de fonctionner avec détermination, résilience et intelligence dans des environnements distribués en temps réel.

Dans cette section

De l'autonomie à l'intelligence distribuée

Avant que le terme agent logiciel ne soit généralisé, les premières recherches informatiques exploraient l'idée d'entités numériques autonomes, c'est-à-dire des systèmes capables d'agir de manière indépendante, de réagir aux entrées et de prendre des décisions en fonction de règles ou d'objectifs internes. Ces premières idées ont jeté les bases conceptuelles de ce qui allait devenir le paradigme de l'agent. (Pour une chronologie historique, consultez la section L'évolution des agents logiciels plus loin dans ce guide.)

Les premiers concepts d'autonomie

La notion de machines ou de programmes agissant indépendamment des opérateurs humains intrigue les concepteurs de systèmes depuis des décennies. Les premiers travaux sur la cybernétique, l'intelligence artificielle et les systèmes de contrôle ont examiné comment les logiciels pouvaient adopter un comportement autorégulé, réagir dynamiquement aux changements et fonctionner sans supervision humaine continue.

Ces idées ont fait de l'autonomie un attribut essentiel des systèmes intelligents et ont ouvert la voie à l'émergence de logiciels capables de décider et d'agir, au lieu de se contenter de réagir ou d'exécuter.

Le modèle d'acteur et l'exécution asynchrone

Dans les années 1970, le modèle d'acteur, introduit dans l'article A Universal Modular ACTOR Formalism for Artificial Intelligence (Hewitt et al. 1973), a fourni un cadre formel pour réfléchir au calcul décentralisé piloté par les messages. Dans ce modèle, les acteurs sont des entités indépendantes qui communiquent exclusivement en transmettant des messages asynchrones et mettent en place des systèmes évolutifs, concurrents et tolérants aux pannes.

Le modèle d'acteur a mis l'accent sur trois attributs clés qui continuent d'influencer la conception moderne des agents :

- Isolation de l'état et du comportement
- Interaction asynchrone entre entités
- Création dynamique et délégation de tâches

Ces attributs répondaient aux besoins des systèmes distribués et préfiguraient les caractéristiques opérationnelles des agents logiciels dans les environnements cloud natifs.

Intelligence distribuée et systèmes multi-agents

À mesure que les systèmes informatiques sont devenus plus interconnectés après les années 1960, les chercheurs ont exploré l'intelligence artificielle distribuée (DAI). Ce domaine s'est concentré sur la manière dont plusieurs entités autonomes pouvaient travailler en collaboration ou de manière compétitive au sein d'un système. La DAI a conduit au développement de systèmes multi-agents, dans lesquels chaque agent a des objectifs, une perception et un raisonnement locaux, mais opère également dans un environnement plus large et interconnecté.

Cette vision de l'intelligence distribuée, dans laquelle la prise de décision est décentralisée et où les comportements émergents découlent de l'interaction des agents, reste au cœur de la conception et de la construction des systèmes modernes basés sur les agents.

La typologie de Nwana et l'essor des agents logiciels

La formalisation du concept d'agent logiciel au milieu des années 1990 a marqué un tournant dans l'évolution des systèmes intelligents. Parmi les contributions les plus influentes à cette formalisation figure l'article phare de Hyacinth S. Nwana, Software Agents : An Overview (Nwana 1996), qui a fourni l'un des premiers cadres complets permettant de catégoriser et de comprendre les agents logiciels dans différentes dimensions.

Dans ce paper, Nwana passe en revue l'état de la recherche sur les agents logiciels et identifie une divergence croissante dans la façon dont les agents étaient définis et mis en œuvre. Le paper souligne la nécessité d'un cadre conceptuel commun et propose une typologie qui classe les agents en fonction de leurs principales capacités. Il passe en revue les systèmes d'agents représentatifs du monde universitaire et industriel, distingue les agents des programmes et objets traditionnels et décrit les défis et les opportunités de l'informatique basée sur les agents.

Nwana souligne que les agents logiciels ne sont pas un concept monolithique mais qu'ils existent dans un éventail de sophistication et de capacités. La typologie sert à clarifier ce paysage et à orienter la conception et les recherches futures.

Nwana définit un agent logiciel comme une entité logicielle qui fonctionne de manière continue et autonome dans un environnement particulier, souvent habité par d'autres agents et processus. Cette définition met l'accent sur deux caractéristiques principales :

- Continuité : L'agent agit de manière persistante dans le temps, sans nécessiter une intervention humaine constante.
- Autonomie : L'agent a la capacité de prendre des décisions et d'agir en conséquence de manière indépendante, en fonction de sa perception de l'environnement.

Cette définition, combinée à la typologie des agents de Nwana, met l'accent sur l'autorité déléguée (par le biais de l'autonomie) et la proactivité en tant que caractéristiques fondamentales des agents. Il fait la différence entre les agents et les sous-programmes ou services en mettant en évidence la capacité de l'agent à agir de manière indépendante pour le compte d'une autre entité et à initier un comportement dans la poursuite d'objectifs, au lieu de répondre uniquement à des commandes directes.

Typologie des agents de Nwana

Pour mieux différencier les différents types d'agents, Nwana introduit un système de classification basé sur six attributs clés :

- Autonomie : L'agent agit sans intervention directe d'humains ou d'autres personnes.
- Capacité sociale : L'agent interagit avec d'autres agents ou avec des humains en utilisant des mécanismes de communication.
- Réactivité : L'agent perçoit son environnement et réagit rapidement.
- Proactivité : l'agent adopte un comportement orienté vers un objectif en prenant l'initiative.
- Adaptabilité et apprentissage : l'agent améliore ses performances au fil du temps grâce à l'expérience.
- Mobilité : l'agent peut se déplacer entre différents environnements système ou réseaux.

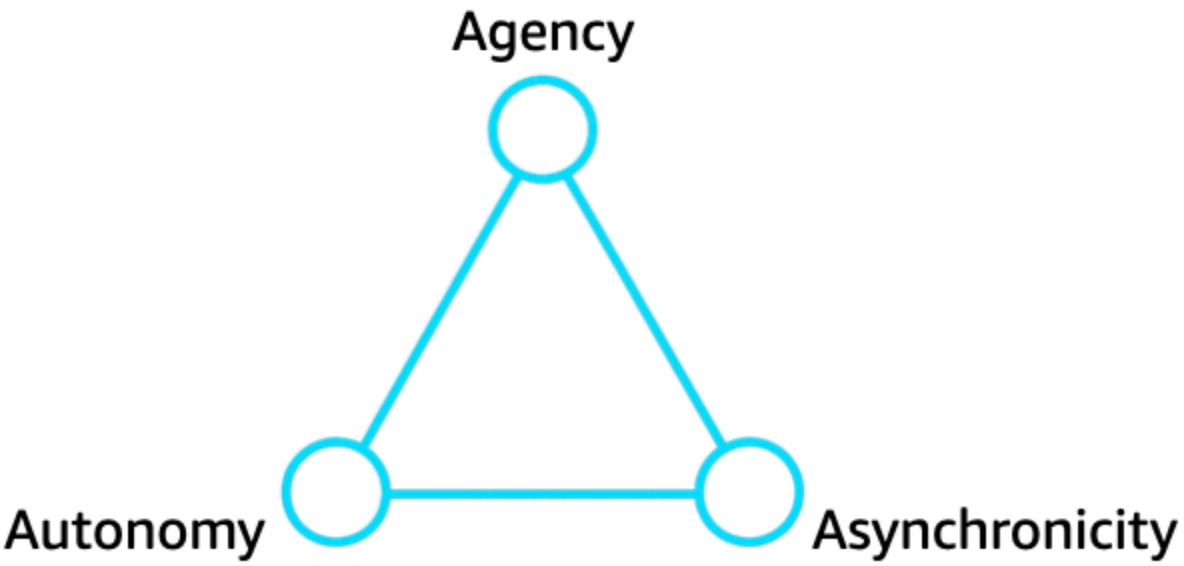
De la typologie aux principes agentiques modernes

Les travaux de Nwana ont servi à la fois de taxonomie et de lentille fondamentale à travers laquelle la communauté informatique a pu évaluer l'évolution des formes d'agence dans les logiciels. L'accent qu'il a mis sur l'autonomie, la proactivité et le concept d'agir au nom d'un utilisateur ou d'un système a jeté les bases de ce que nous considérons aujourd'hui comme un comportement agentique.

Bien que les technologies et les environnements aient évolué, notamment avec l'essor de l'IA générative, de l'infrastructure sans serveur et des cadres d'orchestration multi-agents, les connaissances fondamentales issues du travail de Nwana restent pertinentes. Ils constituent un pont essentiel entre les premières théories des agents et les trois piliers modernes des agents logiciels.

Les trois piliers des agents logiciels modernes

Dans le contexte des plateformes basées sur l'IA, des architectures de microservices et des systèmes pilotés par les événements actuels, les agents logiciels peuvent être définis selon trois principes interdépendants qui les distinguent des services standard ou des scripts d'automatisation : autonomie, asynchronicité et agence. Dans l'illustration suivante et dans les diagrammes suivants, le triangle représente ces trois piliers des agents logiciels modernes.



Autonomie

Les agents modernes fonctionnent de manière indépendante. Ils prennent des décisions en fonction de l'état interne et du contexte environnemental sans avoir besoin d'instructions humaines. Cela leur permet de réagir aux données en temps réel, de gérer leur propre cycle de vie et d'ajuster leur comportement en fonction des objectifs et des informations situationnelles.

L'autonomie est le fondement du comportement des agents. Il permet aux agents de fonctionner sans supervision continue ni flux de contrôle codés en dur.

Asynchronicité

Les agents sont fondamentalement asynchrones. Cela signifie qu'ils répondent aux événements, aux signaux et aux stimuli au fur et à mesure qu'ils se produisent, sans recourir au blocage des appels ou à des flux de travail linéaires. Cette caractéristique permet une communication évolutive et non bloquante, une réactivité dans les environnements distribués et un couplage souple entre les composants.

Grâce à l'asynchronicité, les agents peuvent participer à des systèmes en temps réel et se coordonner avec d'autres services ou agents de manière fluide et efficace.

L'agence comme principe directeur

L'autonomie et l'asynchronicité sont nécessaires, mais ces fonctionnalités ne suffisent pas à elles seules à faire d'un système un véritable agent logiciel. Le principal facteur de différenciation est l'agence, qui introduit :

- Comportement orienté vers un objectif : les agents poursuivent des objectifs et évaluent les progrès réalisés pour les atteindre.
- Prise de décision : les agents évaluent les options et choisissent les actions en fonction de règles, de modèles ou de politiques apprises.
- Intention déléguée : Les agents agissent pour le compte d'une personne, d'un système ou d'une organisation et ont un sens intrinsèque du but.
- Raisonnement contextuel : les agents incorporent de la mémoire ou des modèles de leur environnement pour guider le comportement de manière intelligente.

Un système autonome et asynchrone peut toujours être un service réactif. Ce qui en fait un agent logiciel, c'est sa capacité à agir avec intention et détermination, à être agentique.

Agence ayant un objectif

Les principes d'autonomie, d'asynchronicité et d'agentivité permettent aux systèmes de fonctionner de manière intelligente, adaptative et indépendante dans des environnements distribués. Ces principes sont ancrés dans des décennies d'évolution conceptuelle et architecturale et sous-tendent désormais bon nombre des systèmes d'IA les plus avancés développés aujourd'hui.

Dans cette nouvelle ère d'intelligence artificielle générative, d'orchestration axée sur les objectifs et de collaboration multi-agents, il est essentiel de comprendre ce qui fait qu'un agent logiciel est réellement agentique. Reconnaître l'agence comme la caractéristique déterminante nous aide à dépasser l'automatisation et à entrer dans le domaine de l'intelligence autonome et ciblée.

L'objectif des agents logiciels

À mesure que les systèmes modernes sont devenus de plus en plus complexes, distribués et intelligents, le rôle des agents logiciels a pris de l'importance dans des domaines allant des opérations autonomes aux technologies d'assistance aux utilisateurs. Mais quel est l'objectif sous-jacent des agents logiciels ? Pourquoi concevons-nous des systèmes qui vont au-delà des scripts, des services ou des modèles statiques pour déléguer des tâches à des entités capables de percevoir, de raisonner et d'agir ?

Cette section explore l'objectif fondamental des agents logiciels : permettre une délégation intelligente des tâches au sein d'environnements dynamiques, en mettant l'accent sur l'autonomie, l'adaptabilité et l'action ciblée. Il présente les fondements conceptuels des agents logiciels, retrace leur structure cognitive et décrit les problèmes du monde réel qu'ils sont particulièrement bien placés pour résoudre.

Dans cette section

Du modèle d'acteur à la cognition des agents

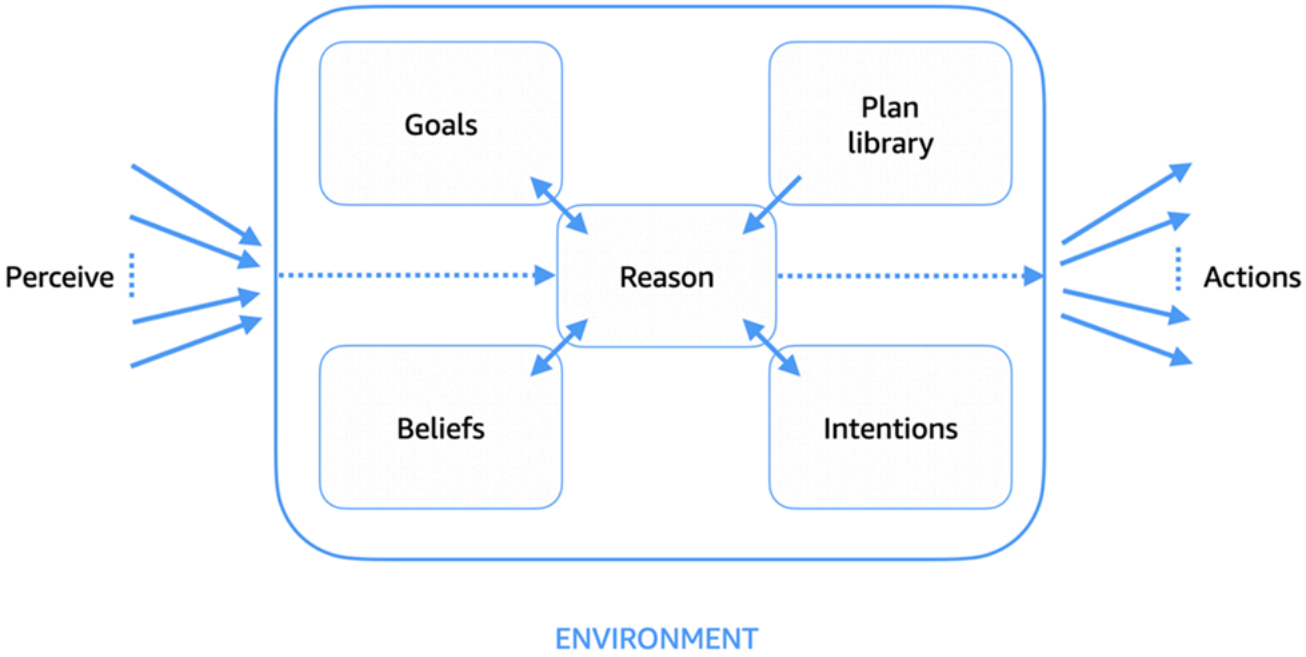
L'objectif et la structure des agents logiciels sont fondés sur des idées issues des premiers modèles de calcul, en particulier le modèle d'acteur introduit par Carl Hewitt dans les années 1970 (Hewitt et al. 1973).

Le modèle d'acteur traite le calcul comme un ensemble d'entités indépendantes s'exécutant simultanément appelées acteurs. Chaque acteur encapsule son propre état, interagit uniquement par transmission de messages asynchrones et peut créer de nouveaux acteurs et déléguer des tâches.

Ce modèle a fourni le fondement conceptuel du raisonnement décentralisé, de la réactivité et de l'isolement, éléments qui sous-tendent l'architecture comportementale des agents logiciels modernes.

La fonction de l'agent : percevoir, raisonner, agir

Au cœur de chaque agent logiciel se trouve un cycle cognitif souvent décrit comme la boucle de perception, de raison et d'action. Ce processus est illustré dans le diagramme suivant. Il définit la manière dont les agents fonctionnent de manière autonome dans des environnements dynamiques.



- Percevoir : les agents collectent des informations (par exemple, des événements, des entrées de capteurs ou des signaux d'API) à partir de l'environnement et mettent à jour leur état interne ou leurs croyances.
- Raison : Les agents analysent les croyances, les objectifs et les connaissances contextuelles actuels à l'aide d'une bibliothèque de plans ou d'un système logique. Ce processus peut impliquer la priorisation des objectifs, la résolution des conflits ou la sélection des intentions.
- Agir : les agents sélectionnent et exécutent les actions qui les rapprochent de la réalisation des objectifs qui leur ont été délégués.

Cette architecture permet aux agents de fonctionner au-delà d'une programmation rigide et permet un comportement flexible, sensible au contexte et orienté vers les objectifs. Il constitue le cadre mental qui guide les objectifs généraux des agents logiciels.

Collaboration autonome et intentionnalité

L'objectif des agents logiciels est d'apporter l'autonomie, la connaissance du contexte et la délégation intelligente à l'informatique moderne. Parce que les agents sont construits sur les principes du modèle d'acteur et incarnés dans le cycle de perception, de raison et d'action, ils permettent des systèmes qui sont non seulement réactifs, mais proactifs et déterminés.

Les agents permettent aux logiciels de décider, de s'adapter et d'agir dans des environnements complexes. Ils représentent les utilisateurs, interprètent les objectifs et mettent en œuvre les tâches à la vitesse de la machine. À mesure que nous entrons dans l'ère de l'IA agentique, les agents logiciels deviennent l'interface opérationnelle entre l'intention humaine et l'action numérique intelligente.

Intention de délégation

Contrairement aux composants logiciels traditionnels, les agents logiciels existent pour agir pour le compte d'un autre utilisateur, d'un autre système ou d'un service de niveau supérieur. Ils ont une intention déléguée, ce qui signifie qu'ils :

- Opérer indépendamment après l'initiation.
- Faites des choix conformes aux objectifs du délégué.
- Gérez les incertitudes et les compromis lors de l'exécution.

Les agents comblent le fossé entre les instructions et les résultats, ce qui permet aux utilisateurs d'exprimer leur intention à un niveau d'abstraction plus élevé au lieu d'exiger des instructions explicites.

Fonctionnement dans des environnements dynamiques et imprévisibles

Les agents logiciels sont conçus pour les environnements dans lesquels les conditions changent constamment, où les données arrivent en temps réel et où le contrôle et le contexte sont distribués.

Contrairement aux programmes statiques qui nécessitent des entrées précises ou une exécution synchrone, les agents s'adaptent à leur environnement et réagissent de manière dynamique. Il s'agit d'une fonctionnalité vitale dans les infrastructures cloud natives, l'informatique de pointe, les réseaux Internet des objets (IoT) et les systèmes de prise de décision en temps réel.

Réduire la charge cognitive humaine

L'un des principaux objectifs des agents logiciels est de réduire la charge cognitive et opérationnelle qui pèse sur les humains. Les agents peuvent :

- Surveillez en permanence les systèmes et les flux de travail.
- Détectez et répondez à des conditions prédéfinies ou émergentes.
- Automatisez les décisions répétitives et volumineuses.
- Réagissez aux changements environnementaux avec une latence minimale.

Lorsque la prise de décision passe des utilisateurs aux agents, les systèmes deviennent plus réactifs, résilients et centrés sur l'humain, et peuvent s'adapter en temps réel aux nouvelles informations ou aux perturbations. Cela permet un délai de réaction plus rapide ainsi qu'une plus grande continuité opérationnelle dans les environnements complexes ou à grande échelle. Il en résulte un changement d'orientation vers l'humain, passant de la prise de décisions au niveau micro à la supervision stratégique et à la résolution créative des problèmes.

Permettre l'intelligence distribuée

La capacité des agents logiciels à fonctionner individuellement ou collectivement permet de concevoir des systèmes multi-agents (MAS) coordonnés entre les environnements ou les organisations. Ces systèmes peuvent répartir les tâches de manière intelligente et négocier, coopérer ou concourir pour atteindre des objectifs composites.

Par exemple, dans un système de chaîne d'approvisionnement mondial, des agents individuels gèrent les usines, les expéditions, les entrepôts et les livraisons sur le dernier kilomètre. Chaque agent fonctionne avec une autonomie locale : les agents d'usine optimisent la production en fonction des contraintes de ressources, les agents d'entrepôt ajustent les flux d'inventaire en temps réel et les agents de livraison réacheminent les expéditions en fonction du trafic et de la disponibilité des clients.

Ces agents communiquent et coordonnent de manière dynamique, et s'adaptent aux perturbations telles que les retards dans les ports ou les pannes de camions sans contrôle centralisé. L'intelligence globale du système émerge de ces interactions et permet une logistique résiliente et optimisée qui dépasse les capacités d'un seul composant.

Dans ce modèle, les agents agissent comme des nœuds dans un tissu de renseignement plus large. Ils forment des systèmes émergents capables de résoudre des problèmes qu'aucun composant ne pourrait résoudre seul.

Agir avec détermination, pas seulement pour réagir

L'automatisation seule ne suffit pas dans les systèmes complexes. L'objectif principal d'un agent logiciel est d'agir avec détermination, d'évaluer les objectifs, d'évaluer le contexte et de faire des choix éclairés. Cela signifie que les agents logiciels poursuivent des objectifs au lieu de répondre uniquement aux déclencheurs. Ils peuvent réviser leurs croyances et leurs intentions en fonction de leur expérience ou de leurs commentaires. Dans ce contexte, les croyances font référence à la représentation interne de l'environnement par l'agent (par exemple, « le colis X est dans l'entrepôt A »), en fonction de ses perceptions (entrées et capteurs). Les intentions font référence aux plans choisis par l'agent pour atteindre un objectif (par exemple, « utiliser l'itinéraire de livraison B et informer le destinataire »). Les agents peuvent également intensifier, différer ou adapter les actions selon les besoins.

C'est cette intentionnalité qui fait des agents logiciels non seulement des exécuteurs réactifs, mais des collaborateurs autonomes dans les systèmes intelligents.

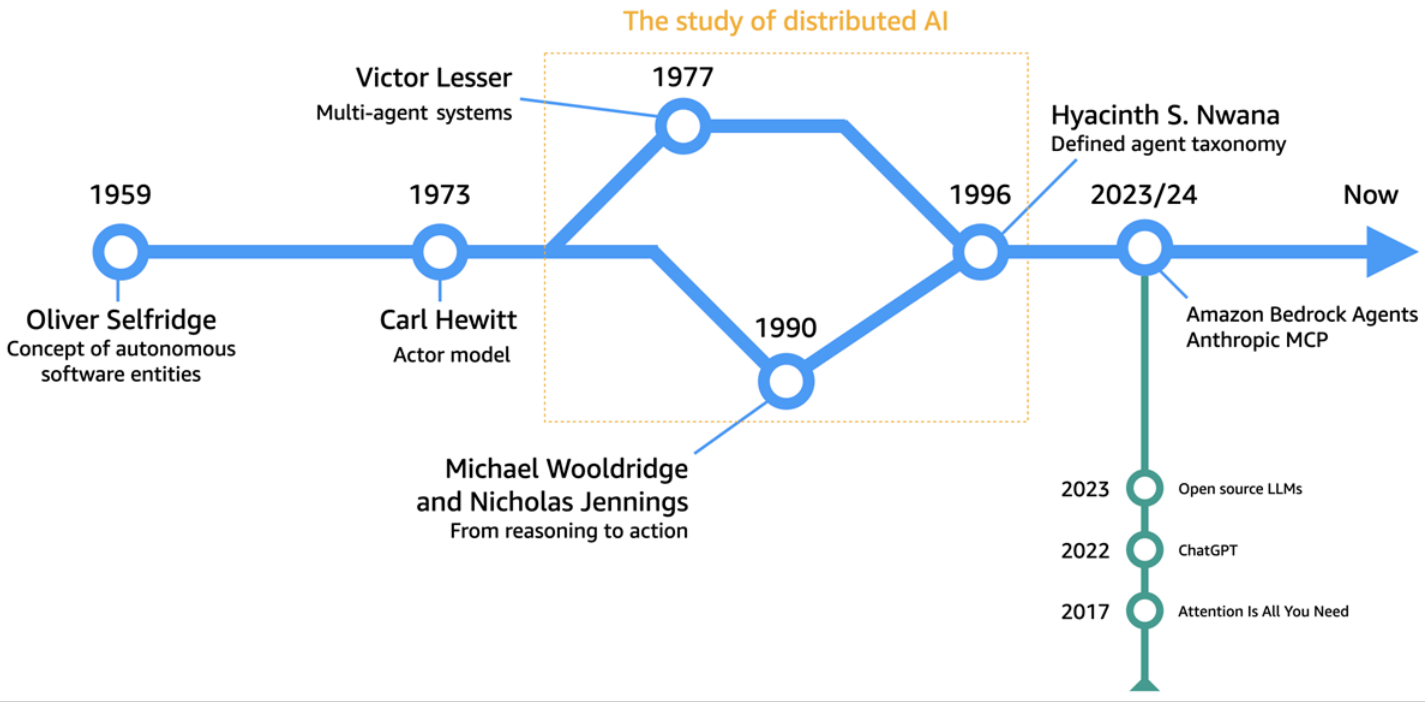
L'évolution des agents logiciels

Le passage de simples systèmes automatisés à des agents logiciels intelligents, autonomes et orientés vers des objectifs reflète des décennies d'évolution dans les domaines de l'informatique, de l'intelligence artificielle et des systèmes distribués.

Cette évolution a été suivie par l'essor de l'apprentissage automatique, qui a fait passer le paradigme des règles élaborées à la main à la reconnaissance statistique des formes. Ces systèmes pouvaient tirer des leçons des données et permettre des avancées en matière de perception, de classification et de prise de décision.

Les grands modèles linguistiques (LLMs) représentent une convergence d'échelle, d'architecture et d'apprentissage non supervisé. LLMs peut raisonner, générer et adapter des tâches avec peu ou pas de formation spécifique. En LLMs combinant une infrastructure native cloud évolutive et des architectures composables, nous concrétisons désormais la vision complète de l'IA agentique : des agents logiciels intelligents capables de fonctionner de manière autonome, consciente du contexte et adaptables à l'échelle de l'entreprise.

Cette section explore l'histoire des agents logiciels, de la théorie fondamentale à la pratique moderne, comme l'illustre le schéma suivant. Il met en évidence la convergence de l'intelligence artificielle distribuée (DAI) et de l'IA générative basée sur des transformateurs, et identifie les principaux jalons qui ont façonné l'émergence de l'IA agentique.



Dans cette section

Fondements des agents logiciels

1959 — Oliver Selfridge : la naissance de l'autonomie dans le logiciel

Les agents logiciels trouvent leur origine dans Oliver Selfridge, qui a introduit le concept d'entités logicielles autonomes (démons), c'est-à-dire des programmes capables de percevoir leur environnement et d'agir de manière indépendante (Selfridge 1959). Ses premiers travaux sur la perception et l'apprentissage des machines ont jeté les bases philosophiques des futures notions selon lesquelles les agents seraient des systèmes indépendants et intelligents.

1973 — Carl Hewitt : le modèle d'acteur

Le modèle d'acteur de Carl Hewitt (Hewitt et al. 1973), qui est un modèle informatique formel qui décrit les agents comme des entités indépendantes et concurrentes, a constitué une avancée décisive. Dans ce modèle, les agents peuvent encapsuler leur propre état et leur propre comportement, communiquer en utilisant la transmission de messages asynchrones, créer dynamiquement d'autres acteurs et leur déléguer des tâches.

Le modèle d'acteur a fourni à la fois le fondement théorique et le paradigme architectural des systèmes distribués basés sur des agents. Ce modèle a préfiguré les implémentations de simultanéité modernes telles que le langage de programmation Erlang et le framework Akka.

Faire mûrir le domaine : du raisonnement à l'action

1977 — Victor Lesser : systèmes multi-agents

À la fin des années 1970, l'intelligence artificielle distribuée (DAI) est apparue. Il a été défendu par Victor Lesser, largement reconnu pour avoir été le pionnier des systèmes multi-agents (MAS). Ses travaux se sont concentrés sur la manière dont les entités logicielles indépendantes pouvaient coopérer, se coordonner et négocier (voir la section Ressources). Cette évolution a donné naissance

à des systèmes capables de résoudre collectivement des problèmes complexes, ce qui constitue une avancée essentielle dans le développement de l'intelligence distribuée.

Années 90 — Michael Wooldridge et Nicholas Jennings : le spectre des agents

Dans les années 1990, le domaine du renseignement distribué avait mûri grâce aux contributions de chercheurs tels que Michael Wooldridge et Nicholas Jennings. Ces chercheurs ont classé les agents selon un spectre allant du réactif au délibératif, des systèmes non cognitifs aux agents de raisonnement axés sur les objectifs (Wooldridge et Jennings 1995). Leurs travaux ont souligné que les agents n'étaient plus des idées abstraites mais étaient appliqués dans un large éventail de domaines pratiques, de la robotique aux logiciels d'entreprise.

Ces chercheurs ont également introduit un changement d'orientation : du raisonnement centralisé à l'action distribuée. Les agents n'étaient plus simplement des penseurs, ils étaient des acteurs qui opéraient dans des environnements en temps réel avec autonomie et détermination.

1996 — Hyacinth S. Nwana : formalisation du concept d'agent

En 1996, Hyacinth S. Nwana a publié l'influent article Software Agents : An Overview, qui fournit la classification des agents la plus complète à ce jour. Sa typologie incluait des attributs tels que l'autonomie, la capacité sociale, la réactivité, la proactivité, l'apprentissage et la mobilité, et faisait la distinction entre les agents logiciels et les constructions logicielles traditionnelles.

Nwana a également proposé une définition désormais largement acceptée, paraphrasée : un agent logiciel est un programme informatique basé sur un logiciel qui agit pour le compte d'un utilisateur ou d'un autre programme dans une relation d'agence, qui découle de la notion de délégation.

Cette formalisation a contribué à faire passer les agents logiciels des constructions théoriques aux applications du monde réel. Cela a donné naissance à une génération de systèmes basés sur des agents dans des domaines tels que les télécommunications, l'automatisation des flux de travail et les assistants intelligents.

Les travaux de Nwana se situent au point de convergence des premières recherches sur l'IA distribuée et des architectures opérationnelles des agents modernes. Il s'agit d'un pont crucial entre la théorie cognitive des agents et leur déploiement pratique dans les systèmes actuels.

Une chronologie parallèle : l'essor des grands modèles linguistiques

Alors que les frameworks d'agents évoluaient, une révolution parallèle et convergente se produisait dans le traitement du langage naturel et l'apprentissage automatique :

- 2017 — transformers : The paper Attention Is All You Need (Vaswani et al. 2017) a introduit l'architecture des transformateurs, qui a considérablement amélioré la façon dont les machines traitent et génèrent le langage.
- 2022 — ChatGPT : OpenAI a publié une interface basée sur le chat pour GPT-3.5 appelée ChatGPT, qui permettait une conversation naturelle et interactive avec un système d'IA à usage général.
- 2023 — open source LLMs : les versions de Llama, Falçon et Mistral ont rendu de puissants modèles largement accessibles et ont accéléré le développement de frameworks d'agents dans les environnements open source et d'entreprise.

Ces innovations ont transformé les modèles linguistiques en moteurs de raisonnement capables d'analyser le contexte, de planifier des actions et de chaîner les réponses, et LLMs sont devenus des outils essentiels pour les agents logiciels intelligents.

Les chronologies convergent : l'émergence de l'IA agentique

2023-2024 — plateformes d'agents adaptées aux entreprises

La convergence des architectures d'agents logiciels distribuées et des architectures basées sur des transformateurs LLMs a abouti à l'essor de l'IA agentique.

- Amazon Bedrock Agents a introduit une méthode entièrement gérée pour créer des agents logiciels axés sur des objectifs et utilisant des outils en utilisant les modèles de base d'Amazon Bedrock.
- Le Model Context Protocol (MCP) d'Anthropic a défini une méthode permettant aux grands modèles de langage d'accéder à des outils, des environnements et de la mémoire externes et d'interagir avec ceux-ci. C'est essentiel pour un comportement contextuel, persistant et autonome.

Ces deux étapes représentent la synthèse de l'agence et du renseignement. Les agents n'étaient plus limités à des flux de travail statiques ou à une automatisation rigide. Ils pouvaient désormais

raisonner en plusieurs étapes, se coordonner avec les outils APIs, maintenir l'état contextuel, apprendre et s'adapter au fil du temps.

Janvier-juin 2025 : capacités d'entreprise étendues

Au cours du premier semestre 2025, le paysage de l'IA agentique s'est considérablement développé grâce aux nouvelles capacités des entreprises. En février 2025, Anthropic a publié Claude 3.7 Sonnet, le premier modèle de raisonnement hybride sur le marché, et la spécification MCP a été largement adoptée.

Des assistants de codage basés sur l'IA tels qu'Amazon Q Developer, Cursor et le MCP WindSurf intégré pour standardiser la génération de code, l'analyse des référentiels et les flux de travail de développement. La version de mars 2025 de MCP a introduit d'importantes fonctionnalités adaptées aux entreprises, notamment l'intégration de la sécurité OAuth 2.1, des types de ressources étendus pour un accès diversifié aux données et des options de connectivité améliorées grâce au protocole HTTP streamable. Sur cette base, elle AWS a annoncé en mai 2025 son adhésion au comité directeur du MCP et sa contribution aux nouvelles capacités de agent-to-agent communication. Cela renforce encore la position du protocole en tant que norme industrielle pour l'interopérabilité de l'IA agentique.

En mai 2025, nous avons AWS renforcé les options offertes aux clients pour créer des flux de travail d'IA agentiques en open source avec le framework Strands Agents. Ce framework indépendant du fournisseur et indépendant du modèle permet aux développeurs d'utiliser des modèles de base sur toutes les plateformes tout en maintenant une intégration approfondie des services. AWS Comme le souligne le blog AWS Open Source, Strands Agents suit une philosophie de conception axée sur le modèle qui place les modèles de base au cœur de l'intelligence des agents. Cela permet aux clients de créer et de déployer plus facilement des agents d'IA sophistiqués pour leurs cas d'utilisation spécifiques.

Emergence — IA magnétique

L'évolution des agents logiciels, des premières idées d'autonomie à l'orchestration moderne basée sur le LLM, a été longue et progressive. Ce qui a commencé avec la vision d'Oliver Selfridge de percevoir les programmes s'est transformé en un écosystème robuste d'agents logiciels intelligents, sensibles au contexte et axés sur les objectifs, capables de collaborer, de s'adapter et de raisonner.

La convergence de l'intelligence artificielle distribuée (DAI) et de l'IA générative basée sur des transformateurs marque le début d'une nouvelle ère dans laquelle les agents logiciels ne sont plus seulement des outils, mais des acteurs autonomes dans les systèmes intelligents.

L'IA agentic représente la prochaine évolution des systèmes logiciels. Il fournit une classe d'agents intelligents autonomes, asynchrones et agentiques, capables d'agir avec une intention déléguée et de fonctionner de manière ciblée dans des environnements dynamiques et distribués. Agentic AI unifie les éléments suivants :

- La lignée architecturale des systèmes multi-agents et le modèle d'acteur
- Le modèle cognitif de percevoir, raisonner, agir
- La puissance génératrice LLMs des transformateurs
- La flexibilité opérationnelle de l'informatique native dans le cloud et sans serveur

Des agents logiciels à l'intelligence artificielle

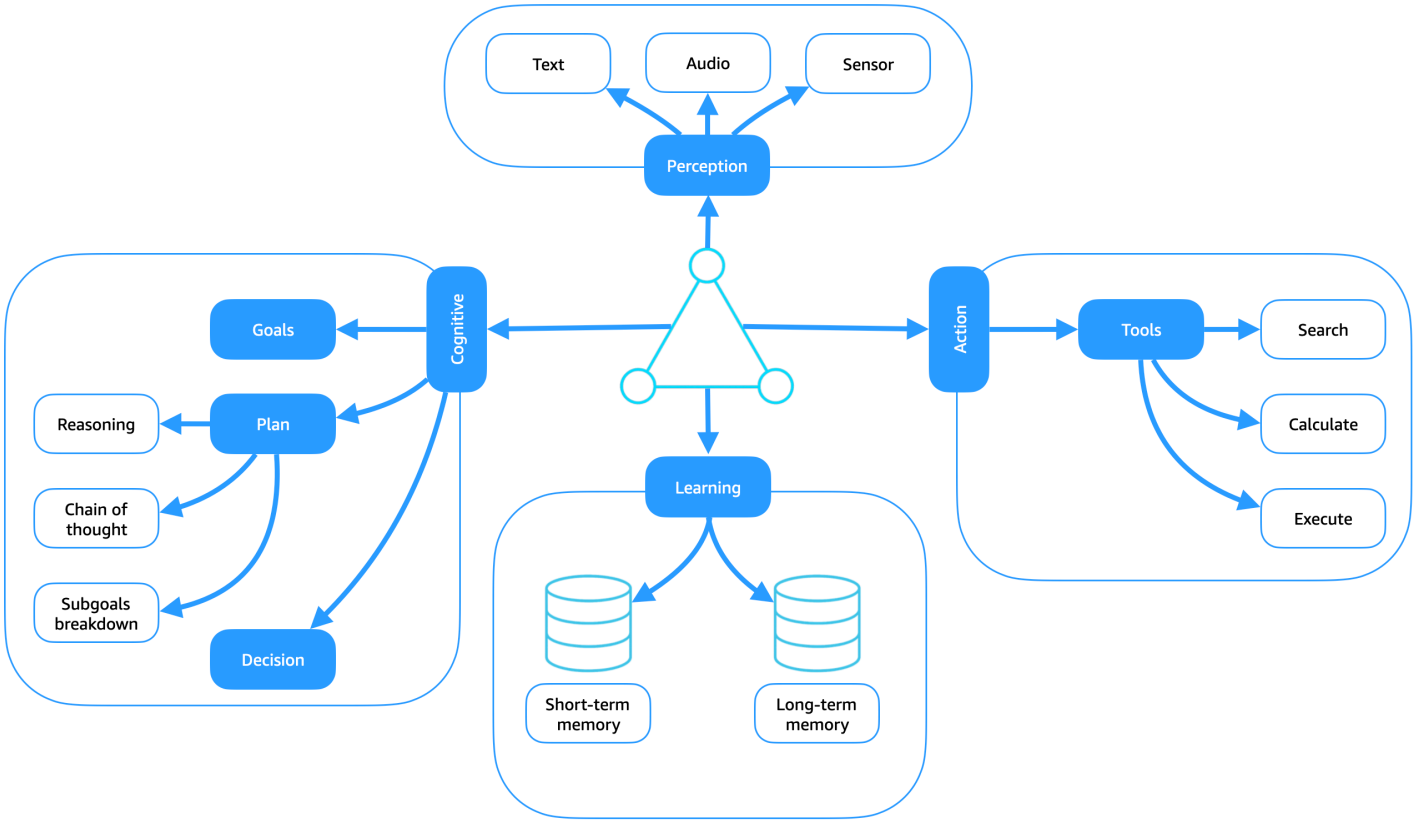
Les agents logiciels sont des entités numériques autonomes conçues pour percevoir leur environnement, raisonner sur leurs objectifs et agir en conséquence. Contrairement aux logiciels traditionnels qui suivent une logique fixe, les agents adaptent leur comportement en fonction des entrées contextuelles et des cadres de décision. Cela les rend idéaux pour les environnements dynamiques et distribués tels que les systèmes natifs du cloud, la robotique, l'automatisation intelligente et, désormais, l'orchestration générative de l'IA.

Cette section présente les éléments de base des agents logiciels et explique comment ces composants interagissent au sein des architectures traditionnelles sur la base du modèle de perception, de raison et d'action. Il explique comment l'IA générative, en particulier les grands modèles linguistiques (LLMs), a transformé la façon dont les agents logiciels raisonnent et planifient. Cela marque un passage fondamental des systèmes basés sur des règles à l'intelligence apprise pilotée par les données de l'IA agentique.

Dans cette section

Éléments de base des agents logiciels

Le schéma suivant présente les principaux modules fonctionnels présents dans la plupart des agents intelligents. Chaque composant contribue à la capacité de l'agent à fonctionner de manière autonome dans des environnements complexes.



Dans le contexte de la boucle de perception, de raison et d'action, la capacité de raisonnement d'un agent est répartie entre ses modules cognitifs et d'apprentissage. Grâce à l'intégration de la mémoire et de l'apprentissage, l'agent développe un raisonnement adaptatif fondé sur l'expérience passée. Lorsque l'agent agit dans son environnement, il crée une boucle de rétroaction émergente : chaque action influence les perceptions futures, et l'expérience qui en résulte est incorporée dans la mémoire et les modèles internes par le biais du module d'apprentissage. Cette boucle continue de perception, de raisonnement et d'action permet à l'agent de s'améliorer au fil du temps et complète le cycle complet de perception, de raison et d'action.

Module de perception

Le module de perception permet à l'agent d'interagir avec son environnement par le biais de diverses modalités de saisie telles que le texte, le son et les capteurs. Ces entrées constituent les données brutes sur lesquelles reposent tous les raisonnements et actions. Les entrées de texte peuvent inclure des instructions en langage naturel, des commandes structurées ou des documents. Les entrées audio incluent des instructions vocales ou des sons environnementaux. Les entrées des capteurs incluent des données physiques telles que des flux visuels, des signaux de mouvement ou des coordonnées GPS. La fonction principale de la perception est d'extraire des caractéristiques et des représentations significatives de ces données brutes. Cela permet à l'agent de construire

une compréhension précise et exploitable de son contexte actuel. Le processus peut impliquer l'extraction de caractéristiques, la reconnaissance d'objets ou d'événements et l'interprétation sémantique, et constitue la première étape critique de la boucle de perception, de raison et d'action. Une perception efficace garantit que le raisonnement et la prise de décisions en aval sont fondés sur une connaissance pertinente de up-to-date la situation.

Module cognitif

Le module cognitif sert de noyau délibératif à l'agent logiciel. Il est chargé d'interpréter les perceptions, de former l'intention et de guider un comportement déterminé grâce à une planification et à une prise de décision axées sur les objectifs. Ce module transforme les entrées en processus de raisonnement structurés, ce qui permet à l'agent d'agir intentionnellement plutôt que de manière réactive. Ces processus sont gérés par le biais de trois sous-modules clés : objectifs, planification et prise de décision.

Sous-module Objectifs

Le sous-module des objectifs définit l'intention et la direction de l'agent. Les objectifs peuvent être explicites (par exemple, « naviguer vers un emplacement » ou « soumettre un rapport ») ou implicites (par exemple, « maximiser l'engagement des utilisateurs » ou « minimiser le temps de latence »). Ils sont au cœur du cycle de raisonnement de l'agent et fournissent un état cible pour sa planification et ses décisions.

L'agent évalue en permanence les progrès réalisés par rapport à ses objectifs et peut redéfinir les priorités ou régénérer les objectifs en fonction de nouvelles perceptions ou de nouveaux apprentissages. Cette connaissance des objectifs permet à l'agent de s'adapter aux environnements dynamiques.

Sous-module de planification

Le sous-module de planification élabore des stratégies pour atteindre les objectifs actuels de l'agent. Il génère des séquences d'actions, décompose les tâches de manière hiérarchique et sélectionne des plans prédéfinis ou générés dynamiquement.

Pour fonctionner efficacement dans des environnements non déterministes ou changeants, la planification n'est pas statique. Les agents modernes peuvent générer des chain-of-thought séquences, introduire des sous-objectifs en tant qu'étapes intermédiaires et réviser les plans en temps réel lorsque les conditions changent.

Ce sous-module est étroitement lié à la mémoire et à l'apprentissage, et permet à l'agent d'affiner sa planification au fil du temps en fonction des résultats antérieurs.

Sous-module de prise de décision

Le sous-module de prise de décision évalue les plans et les actions disponibles afin de sélectionner l'étape suivante la plus appropriée. Il intègre les informations issues de la perception, du plan actuel, des objectifs de l'agent et du contexte environnemental.

La prise de décision tient compte de :

- Compromis entre des objectifs contradictoires
- Seuils de confiance (par exemple, incertitude de perception)
- Conséquences des actions
- L'expérience acquise par l'agent

Selon l'architecture, les agents peuvent s'appuyer sur le raisonnement symbolique, l'heuristique, l'apprentissage par renforcement ou les modèles linguistiques (LLMs) pour prendre des décisions éclairées. Ce processus garantit que le comportement de l'agent reste adapté au contexte, aligné sur les objectifs et adaptatif.

Module d'action

Le module d'action est chargé d'exécuter les décisions sélectionnées par l'agent et de s'interfacer avec le monde externe ou les systèmes internes pour produire des effets significatifs. Il représente la phase d'acte de la boucle de perception, de raison, d'acte, où l'intention est transformée en comportement.

Lorsque le module cognitif sélectionne une action, il coordonne l'exécution par le biais de sous-modules spécialisés, où chaque sous-module s'aligne sur l'environnement intégré de l'agent :

- Activation physique : pour les agents intégrés à des systèmes robotiques ou à des appareils IoT, ce sous-module traduit les décisions en mouvements physiques réels ou en instructions matérielles.

  Exemples : diriger un robot, déclencher une vanne, activer un capteur.
- Interaction intégrée : ce sous-module gère les actions non physiques mais visibles de l'extérieur, telles que l'interaction avec des systèmes logiciels, des plateformes ou. APIs

Exemples : envoi d'une commande à un service cloud, mise à jour d'une base de données, envoi d'un rapport en appelant une API.

- Invocation d'outils : les agents étendent souvent leurs capacités en utilisant des outils spécialisés pour accomplir des sous-tâches telles que les suivantes :
  - Recherche : interrogation de sources de connaissances structurées ou non structurées
  - Résumé : compression des entrées de texte volumineuses en aperçus de haut niveau
  - Calcul : exécution de calculs logiques, numériques ou symboliques

  L'invocation d'outils permet de composer des comportements complexes grâce à des compétences modulaires et appelables.

Module d'apprentissage

Le module d'apprentissage permet aux agents de s'adapter, de généraliser et de s'améliorer au fil du temps en fonction de leur expérience. Il soutient le processus de raisonnement en affinant continuellement les modèles internes, les stratégies et les politiques de décision de l'agent en utilisant le feedback issu de la perception et de l'action.

Ce module fonctionne en coordination avec la mémoire à court terme et à long terme :

- Mémoire à court terme : stocke le contexte transitoire, tel que l'état du dialogue, les informations sur les tâches en cours et les observations récentes. Cela aide l'agent à maintenir la continuité des interactions et des tâches.
- Mémoire à long terme : code les connaissances persistantes issues d'expériences passées, y compris les objectifs atteints précédemment, les résultats des actions et les états environnementaux. La mémoire à long terme permet à l'agent de reconnaître des modèles, de réutiliser des stratégies et d'éviter de répéter les erreurs.

Modes d'apprentissage

Le module d'apprentissage prend en charge une gamme de paradigmes, tels que l'apprentissage supervisé, non supervisé et l'apprentissage par renforcement, qui prennent en charge différents environnements et rôles d'agent :

- Apprentissage supervisé : met à jour les modèles internes sur la base d'exemples étiquetés, souvent à partir de commentaires humains ou d'ensembles de données de formation.

Exemple : apprendre à classer les intentions des utilisateurs en fonction des conversations précédentes.

- Apprentissage non supervisé : identifie les modèles ou les structures cachés dans les données sans étiquettes explicites.

  Exemple : regroupement de signaux environnementaux pour détecter des anomalies.

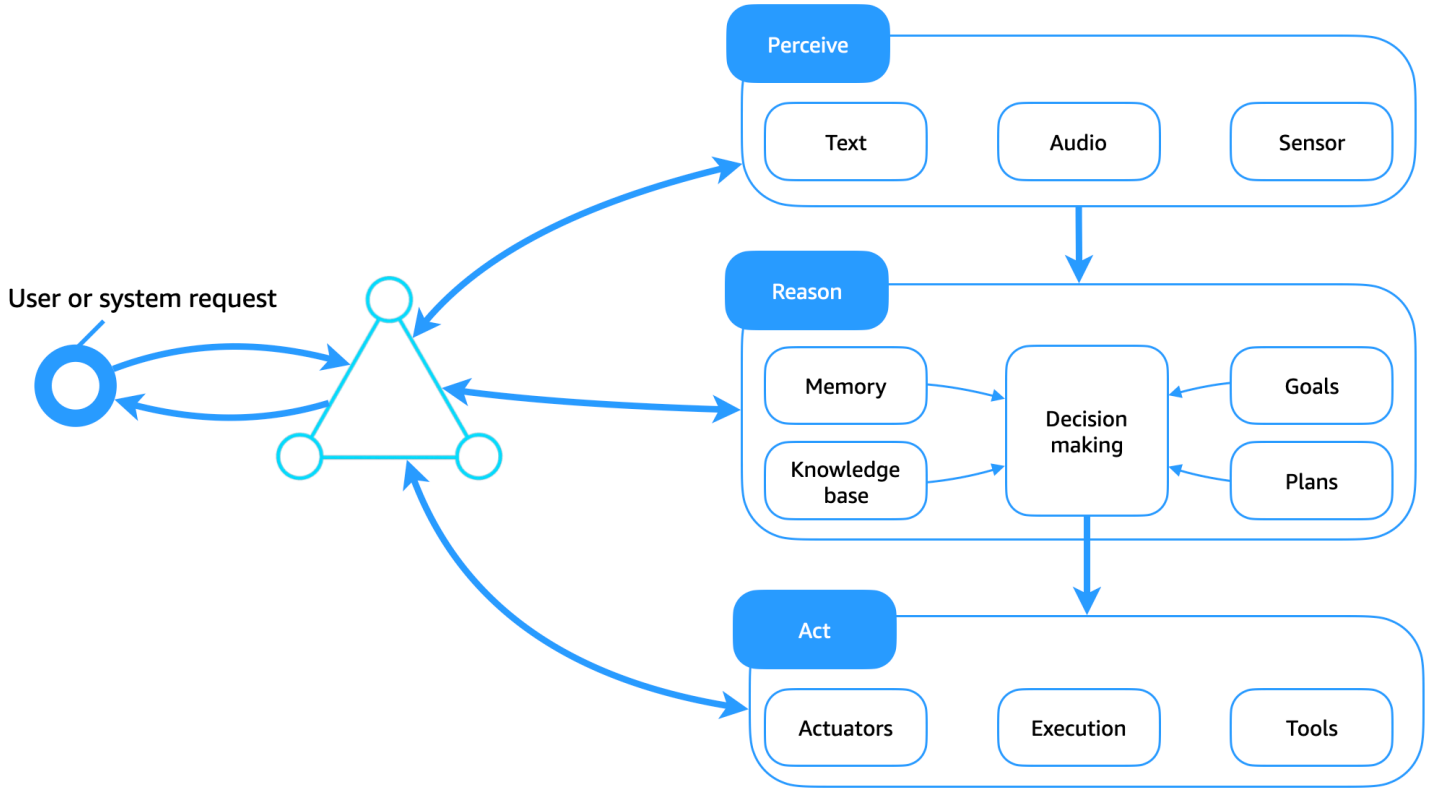
- Apprentissage par renforcement : optimise le comportement par essais et erreurs en maximisant les récompenses cumulées dans les environnements interactifs.

  Exemple : apprendre quelle stratégie permet d'accomplir les tâches le plus rapidement.

L'apprentissage s'intègre étroitement au module cognitif de l'agent. Il affine les stratégies de planification en fonction des résultats passés, améliore la prise de décision grâce à l'évaluation des succès historiques et améliore continuellement la correspondance entre la perception et l'action. Grâce à cette boucle fermée d'apprentissage et de feedback, les agents évoluent au-delà de l'exécution réactive pour devenir des systèmes d'auto-amélioration capables de s'adapter à de nouveaux objectifs, conditions et contextes au fil du temps.

Architecture d'agent traditionnelle : percevoir, raisonner, agir

Le schéma suivant illustre le fonctionnement des éléments de base abordés dans la section précédente dans le cadre du cycle percevoir, raisonner et agir.



Module Perception

Le module de perception agit comme interface sensorielle de l'agent avec le monde extérieur. Il transforme les données environnementales brutes en représentations structurées qui éclairent le raisonnement. Cela inclut la gestion de données multimodales telles que du texte, du son ou des signaux de capteurs.

- La saisie de texte peut provenir de commandes utilisateur, de documents ou de dialogues.
- L'entrée audio inclut des instructions vocales ou des sons environnementaux.
- L'entrée du capteur capture des signaux du monde réel tels que le mouvement, les flux visuels ou le GPS.

Lorsque l'entrée brute a été ingérée, le processus de perception effectue une extraction des caractéristiques, suivie de la reconnaissance d'objets ou d'événements et d'une interprétation sémantique pour créer un modèle significatif de la situation actuelle. Ces résultats fournissent un contexte structuré pour la prise de décisions en aval et ancrent le raisonnement de l'agent dans des observations du monde réel.

Module Reason

Le module de raison est le cœur cognitif de l'agent. Il évalue le contexte, formule l'intention et détermine les actions appropriées. Ce module orchestre les comportements axés sur les objectifs en utilisant à la fois les connaissances apprises et le raisonnement.

Le module Reason se compose de sous-modules étroitement intégrés :

- Mémoire : conserve l'état du dialogue, le contexte des tâches et l'historique épisodique dans des formats à court et à long terme.
- Base de connaissances : donne accès à des règles symboliques, à des ontologies ou à des modèles appris (tels que des intégrations, des faits et des politiques).
- Objectifs et plans : définit les résultats souhaités et élabore des stratégies d'action pour les atteindre. Les objectifs peuvent être mis à jour de manière dynamique et les plans peuvent être modifiés de manière adaptative en fonction des commentaires.
- Prise de décision : agit en tant que moteur d'arbitrage central en évaluant les options, en évaluant les compromis et en sélectionnant l'action suivante. Ce sous-module prend en compte les seuils de confiance, l'alignement des objectifs et les contraintes contextuelles.

Ensemble, ces composants permettent à l'agent de raisonner sur son environnement, de mettre à jour ses croyances, de sélectionner des voies et de se comporter de manière cohérente et adaptative. Le module Reason comble le fossé entre la perception et le comportement.

Module Act

Le module act exécute la décision sélectionnée par l'agent en s'interfaçant avec l'environnement numérique ou physique pour effectuer les tâches. C'est là que l'intention devient action.

Ce module comprend trois canaux fonctionnels :

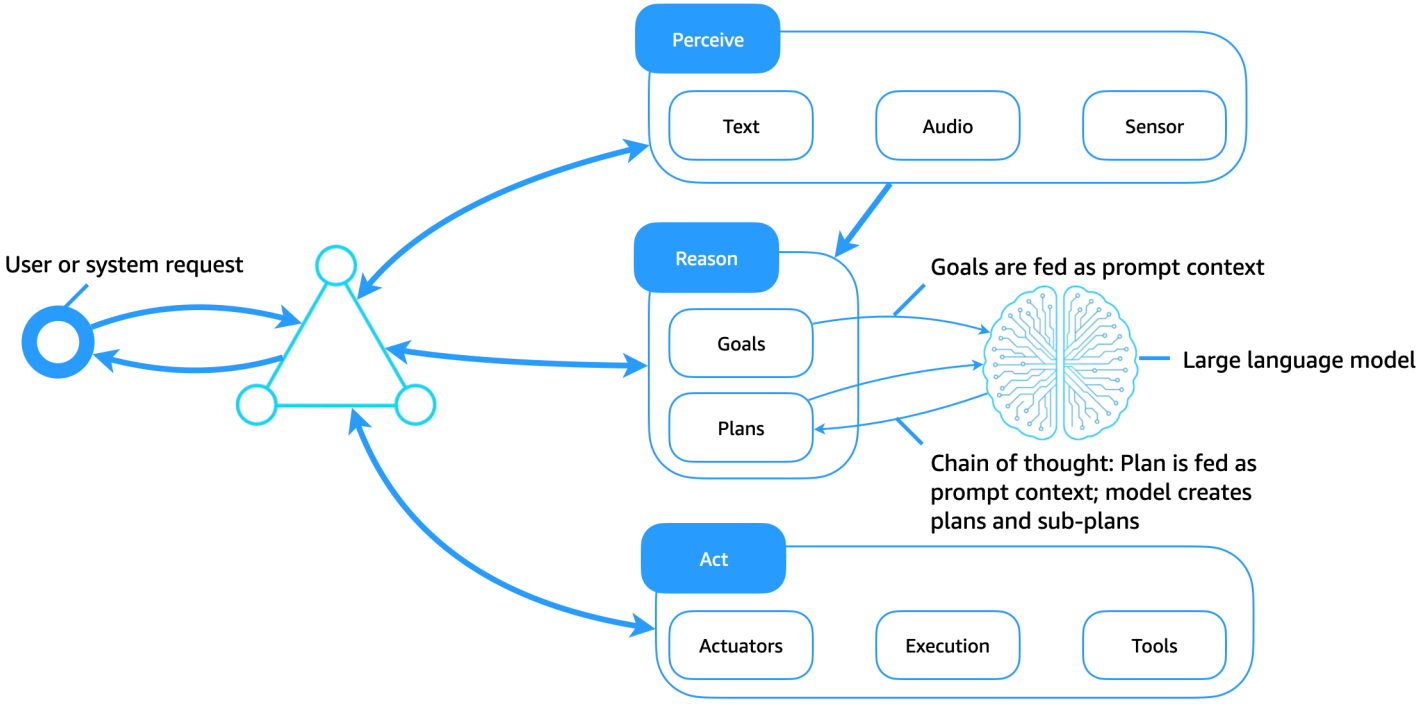
- Actionneurs : pour les agents présents physiquement (tels que les robots et les appareils IoT), ils contrôlent les interactions au niveau du matériel telles que le mouvement, la manipulation ou la signalisation.
- Exécution : gère les actions basées sur le logiciel, notamment l'invocation APIs, l'envoi de commandes et la mise à jour des systèmes.

- Outils : active des fonctionnalités telles que la recherche, la synthèse, l'exécution de code, le calcul et la gestion de documents. Ces outils sont souvent dynamiques et sensibles au contexte, ce qui accroît l'utilité de l'agent.

Les sorties du module act sont renvoyées dans l'environnement et bouclent la boucle. Ces résultats sont à nouveau perçus par l'agent. Ils mettent à jour l'état interne de l'agent et éclairent les décisions futures, complétant ainsi le cycle de perception, de raisonnement et d'action.

Agents d'IA génératifs : remplacer la logique symbolique par LLMs

Le schéma suivant montre comment les grands modèles de langage (LLMs) constituent désormais un noyau cognitif flexible et intelligent pour les agents logiciels. Contrairement aux systèmes logiques symboliques traditionnels, qui reposent sur des bibliothèques de plans statiques et des règles codées à la main, ils LLMs permettent un raisonnement adaptatif, une planification contextuelle et une utilisation dynamique des outils, qui transforment la façon dont les agents perçoivent, raisonnent et agissent.



Améliorations clés

Cette architecture améliore l'architecture traditionnelle des agents comme suit :

- LLMs en tant que moteurs cognitifs : les objectifs, les plans et les requêtes sont transmis au modèle sous forme de contexte rapide. Le LLM génère des voies de raisonnement (telles que des chaînes de pensée), décompose les tâches en sous-objectifs et décide des prochaines actions.
- Utilisation de l'outil par le biais d'instructions : LLMs peut être dirigée par le biais d'agents d'utilisation d'outils ou d'une invite à appeler APIs et à rechercher, interroger, calculer et interpréter les résultats. ReAct
- Planification contextuelle : les agents génèrent ou révisent les plans de manière dynamique en fonction de l'objectif actuel de l'agent, de l'environnement de saisie et des commentaires, sans avoir besoin de bibliothèques de plans codées en dur.
- Contexte rapide sous forme de mémoire : au lieu d'utiliser des bases de connaissances symboliques, les agents encodent la mémoire, les plans et les objectifs sous forme de jetons d'invite transmis au modèle.
- Apprentissage par le biais d'un apprentissage contextuel en quelques étapes : LLMs adaptez les comportements grâce à une ingénierie rapide, ce qui réduit le besoin de reconversion explicite ou de bibliothèques de plans rigides.

Atteindre la mémoire à long terme avec les agents basés sur le LLM

Contrairement aux agents traditionnels, qui stockaient la mémoire à long terme dans des bases de connaissances structurées, les agents d'IA générative doivent fonctionner dans les limites de la fenêtre contextuelle de LLMs. Pour étendre la mémoire et favoriser la persistance de l'intelligence, les agents d'IA générative utilisent plusieurs techniques complémentaires : le stockage d'agents, la génération augmentée par extraction (RAG), l'apprentissage contextuel et le chaînage rapide, ainsi que le pré-entraînement.

Agent Store : mémoire externe à long terme

L'état de l'agent, l'historique utilisateur, les décisions et les résultats sont stockés dans une mémoire d'agent à long terme (telle qu'une base de données vectorielle, un magasin d'objets ou un magasin de documents). Les mémoires pertinentes sont récupérées à la demande et injectées dans le contexte d'invite LLM lors de l'exécution. Cela crée une boucle de mémoire persistante, dans laquelle l'agent assure la continuité entre les sessions, les tâches ou les interactions.

CHIFFON

RAG améliore les performances du LLM en combinant les connaissances récupérées avec des capacités génératives. Lorsqu'un objectif ou une requête est émis, l'agent recherche un index de

récupération (par exemple, par le biais d'une recherche sémantique de documents, de conversations antérieures ou de connaissances structurées). Les résultats récupérés sont ajoutés à l'invite LLM, qui fonde la génération sur des faits externes ou un contexte personnalisé. Cette méthode étend la mémoire effective de l'agent et améliore la fiabilité et l'exactitude des faits.

Apprentissage contextuel et chaînage rapide

Les agents conservent leur mémoire à court terme en utilisant un contexte de jeton en session et un chaînage d'invite structuré. Les éléments contextuels, tels que le plan actuel, les résultats des actions précédentes et le statut de l'agent, sont transmis entre les appels pour guider le comportement.

Préformation continue et mise au point

Pour les agents spécifiques à un domaine, il est LLMs possible de poursuivre leur formation préalable sur des collections personnalisées telles que les journaux, les données d'entreprise ou la documentation des produits. Alternativement, le réglage précis des instructions ou l'apprentissage par renforcement à partir du feedback humain (RLHF) peuvent intégrer un comportement semblable à celui d'un agent directement dans le modèle. Cela fait passer les modèles de raisonnement de la logique des délais à la représentation interne du modèle, réduit la longueur des instructions et améliore l'efficacité.

Avantages combinés de l'IA agentic

Ces techniques, lorsqu'elles sont utilisées ensemble, permettent aux agents d'IA générative de :

- Maintenez une conscience contextuelle au fil du temps.
- Adaptez le comportement en fonction de l'historique ou des préférences de l'utilisateur.
- Prenez des décisions en utilisant up-to-date des connaissances factuelles ou privées.
- Adaptez-vous aux cas d'utilisation en entreprise avec des comportements persistants, conformes et explicables.

En ajoutant de la mémoire externe, LLMs des couches de récupération et une formation continue, les agents peuvent atteindre un niveau de continuité cognitive et un objectif qui ne pouvaient pas être atteints auparavant par le biais des seuls systèmes symboliques.

Comparaison de l'IA traditionnelle aux agents logiciels et à l'IA agentique

Le tableau suivant fournit une comparaison détaillée de l'IA traditionnelle, des agents logiciels et de l'IA agentique.

Caractéristiques	IA traditionnelle	Agents logiciels	IA agentic
Exemples	Filtres anti-spam , classificateurs d'images, moteurs de recommandation	Chatbots, planifica teurs de tâches, agents de surveillance	Assistants IA, agents de développement autonomes, orchestra tions LLM multi-age nts
Modèle d'exécution	Batch ou synchrone	Programmé ou piloté par un événement	Asynchrone, piloté par les événements et axé sur les objectifs
Autonomie	Limité ; nécessite souvent une orchestra tion humaine ou externe	Moyen ; fonctionn e indépendamment dans des limites prédéfinies	Élevé ; agit de façon autonome grâce à des stratégies adaptatives
Réactivité	Réactif aux données d'entrée	Réactif à l'environ nement et aux événements	Réactif et proactif ; anticipe et initie des actions
Proactivité	Rare	Présent dans certains systèmes	Attribut de base ; entraîne un comportement orienté vers un objectif
Communication	Minimum ; généralem ent autonome ou lié à l'API	Messagerie interagent ou agent-homme	Multi-agents et human-in-the-loop interactions riches
Prise de décisions	Inférence du modèle uniquement (classifi	Raisonnement symbolique ou	Raisonnement dynamique, contextue

Caractéristiques	IA traditionnelle	Agents logiciels	IA agentic
	cation, prédiction, etc.)	décisions basées sur des règles ou scénarisées	l et basé sur les objectifs (souvent amélioré par le LLM)
Intention déléguée	Non ; exécute des tâches définies directement par l'utilisateur	Partiel ; agit pour le compte d'utilisateurs ou de systèmes dont la portée est limitée	Oui ; agit avec des objectifs délégués, souvent entre les services, les utilisate urs ou les systèmes
Apprentissage et adaptation	Souvent centré sur le modèle (par exemple, formation en machine learning)	Parfois adaptatif	Apprentissage, mémoire ou raisonnement intégrés (par exemple, feedback, autocorre ction)
Agence	Aucun ; outils pour humains	Implicite ou basique	Explicite ; fonctionn e avec un but, des objectifs et une orientation autonome
Connaissance du contexte	Faible ; apatride ou basé sur des instantanés	Modéré ; un certain suivi de l'état	Élevé ; utilise la mémoire, le contexte situationnel et les modèles environne mentaux
Rôle de l'infrast ructure	Intégré dans des applications ou des pipelines d'analyse	Middleware ou composant de couche de service	Maillage d'agents composable intégré aux systèmes cloud, sans serveur ou de périphérie

Pour résumer :

- L'IA traditionnelle est centrée sur les outils et limitée sur le plan fonctionnel. Il met l'accent sur la prédiction ou la classification.
- Les agents logiciels traditionnels introduisent l'autonomie et une communication de base, mais ils sont souvent limités par des règles ou statiques.
- L'IA agentic allie autonomie, asynchronie et capacité d'agir. Il permet à des entités intelligentes, axées sur des objectifs, de raisonner, d'agir et de s'adapter au sein de systèmes complexes. L'intelligence artificielle agentic est donc idéale pour le futur basé sur le cloud natif et piloté par l'IA.

Étapes suivantes

Ce guide décrit l'histoire et les fondements de l'IA agentique, qui représente l'évolution des agents logiciels traditionnels vers des systèmes autonomes et intelligents alimentés par l'IA générative. Il a décrit comment les premiers agents logiciels suivaient des règles et une logique prédéfinies pour automatiser les tâches dans des limites définies, et a expliqué comment l'IA agentic s'appuie sur cette base en incorporant de grands modèles linguistiques, qui permettent aux agents de raisonner, d'apprendre et de s'adapter de manière dynamique dans des environnements ouverts.

Vous pouvez explorer l'IA agentic en profondeur en consultant les publications suivantes de cette série :

- L'opérationnalisation de l'IA agentique AWS fournit une stratégie organisationnelle visant à transformer l'IA agentique issue d'expériences isolées en une infrastructure génératrice de valeur à l'échelle de l'entreprise.
- Les modèles et les flux de travail d'Agentic AI AWS abordent les plans fondamentaux et les constructions modulaires utilisés pour concevoir, composer et orchestrer des agents d'IA orientés vers des objectifs.
- Les frameworks, protocoles et outils d'IA agentic AWS couvrent les bases logicielles, les boîtes à outils et les protocoles à prendre en compte lors de la création de vos solutions d'IA agentic.
- La création d'architectures sans serveur pour l'IA agentic AWS aborde les architectures sans serveur en tant que base naturelle des charges de travail d'IA modernes et décrit comment créer des architectures sans serveur natives à l'IA dans le. AWS Cloud
- La création d'architectures multi-locataires pour l'IA agentic AWS décrit l'utilisation d'agents d'IA dans des environnements multi-locataires, y compris les considérations relatives à l'hébergement, les modèles de déploiement et les plans de contrôle.

Ressources

Pour plus d'informations sur les concepts abordés dans ce guide, consultez les guides et articles suivants.

AWS références

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

Autres références

- Hewitt, Carl, Peter Bishop et Richard Steiger. « Un formalisme ACTOR modulaire universel pour l'intelligence artificielle. » Actes de la 3e Conférence internationale conjointe sur l'intelligence artificielle (1973) : 235-245. https://www.ijcai. org/Proceedings/73/Papers/027B.pdf
- Lesser, Victor R., publications pertinentes (voir liste complète) :
  - Lesser, Victor R. et Daniel D. Corkill. « Systèmes distribués coopératifs et fonctionnellement précis. » Transactions IEEE sur les systèmes, l'homme et la cybernétique 11, n° 1 (1981) : 81-96. https://ieeexplore.ieee. org/abstract/document/4308581
  - Decker, Keith S. et Victor R. Lesser. « La communication au service de la coordination ». Atelier de l'AAAI sur la planification de la communication entre agents (1994). https://www.researchgate.net/profile/Victor-Lesser/publication/2768884_Communication_in_the_Service_of_Coordination/links/00b7d51cc2a0750cb4000000/Communication-in-the-Service-of-Coordination.pdf
  - Durfee, Edmund H., Victor R. Lesser et Daniel D. Corkill. « Tendances en matière de résolution coopérative de problèmes distribuée ». Transactions IEEE sur les connaissances et l'ingénierie des données (1989). http://mas.cs.umass. edu/Documents/ieee-tkde89.pdf
  - Durfee, Edmund H., V.R. Lesser et D.D. Corkill, « Intelligence artificielle distribuée ». Coopération par la communication dans un réseau distribué de résolution de problèmes (1987) : 29-58. https://www.academia. edu/download/79885643/durf94_1.pdf
  - Lâasri, Brigitte, Hassan Lâasri, Susan Lander et Victor Lesser. « Un modèle générique pour les agents de négociation intelligents. » Journal international des systèmes d'information coopératifs 01, n° 02 (1992) : 291-317. https://doi.org/10.1142/S0218215792000210

  - Lander, Susan E. et Victor R. Lesser. « Comprendre le rôle de la négociation dans la recherche distribuée parmi des agents hétérogènes. » IJCAI'93 : Actes de la 13e conférence internationale conjointe sur l'intelligence artificielle (1993) : 438-444. https://www.ijcai. org/Proceedings/93-1/Papers/062.pdf
  - Lander, Susan, Victor R. Lesser et Margaret E. Connell. « Stratégies de résolution des conflits pour les agents experts coopérants » CKBS'90 : Actes de la conférence de travail internationale sur les systèmes coopératifs basés sur les connaissances (octobre 1990) : 183-200. https://doi.org/10.1007/978-1-4471-1831-2_10
  - Prasad, M. V. Nagendra, Victor Lesser et Susan E. Lander. « Expériences d'apprentissage dans un système multi-agents hétérogène. » Atelier IJCAI-95 sur l'adaptation et l'apprentissage dans les systèmes multi-agents (1995) : 59-64. https://www.researchgate.net/publication/2784280_Learning_Experiments_in_a_Heterogeneous_Multi-agent_System
- Nwana, Hyacinth S. « Agents logiciels : un aperçu ». Revue d'ingénierie des connaissances 11, no. 3 (octobre/novembre 1996) : 205-244. https://teaching.shu.ac. uk/aces/rh1/elearning/multiagents/introduction/nwana.pdf
- Selfridge, Oliver G. « Pandémonium : un paradigme pour l'apprentissage ». Mécanisation des processus de pensée : actes d'un symposium tenu au National Physical Laboratory 1 (1959) : 511—529. https://aitopics. org/download/classics: 504 E1 BAC
- Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser et Illia Polosukhin. « L'attention est tout ce dont vous avez besoin. » Actes de la 31e conférence sur les systèmes de traitement de l'information neuronale (NIPS). Avancées dans les systèmes de traitement de l'information neuronale 30 (2017) : 5998-6008. https://proceedings.neurips. cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- Wooldridge, Michael et Nicholas R. Jennings. « Agents intelligents : théorie et pratique ». Revue d'ingénierie des connaissances 10, no. 2 (janvier 1995) : 115 à 152. https://www.cs.cmu.edu/ ~motionplanning/papers/sbp_papers/integrated1/woodridge _intelligent_agents.pdf

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Pour être averti des mises à jour à venir, abonnez-vous à un fil RSS.

Modification	Description	Date
Publication initiale	—	14 juillet 2025

AWS Glossaire des directives prescriptives

Les termes suivants sont couramment utilisés dans les stratégies, les guides et les modèles fournis par les directives AWS prescriptives. Pour suggérer des entrées, veuillez utiliser le lien Faire un commentaire à la fin du glossaire.

Nombres

7 R

Sept politiques de migration courantes pour transférer des applications vers le cloud. Ces politiques s'appuient sur les 5 R identifiés par Gartner en 2011 et sont les suivantes :

- Refactorisation/réarchitecture : transférez une application et modifiez son architecture en tirant pleinement parti des fonctionnalités natives cloud pour améliorer l'agilité, les performances et la capacité de mise à l'échelle. Cela implique généralement le transfert du système d'exploitation et de la base de données. Exemple : migrez votre base de données Oracle sur site vers l'édition compatible avec Amazon Aurora PostgreSQL.
- Replateformer (déplacer et remodeler) : transférez une application vers le cloud et introduisez un certain niveau d'optimisation pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle sur site vers Amazon Relational Database Service (Amazon RDS) pour Oracle dans le. AWS Cloud
- Racheter (rachat) : optez pour un autre produit, généralement en passant d'une licence traditionnelle à un modèle SaaS. Exemple : migrez votre système de gestion de la relation client (CRM) vers Salesforce.com.
- Réhéberger (lift and shift) : transférez une application vers le cloud sans apporter de modifications pour tirer parti des fonctionnalités du cloud. Exemple : migrez votre base de données Oracle locale vers Oracle sur une EC2 instance du AWS Cloud.
- Relocaliser (lift and shift au niveau de l'hyperviseur) : transférez l'infrastructure vers le cloud sans acheter de nouveau matériel, réécrire des applications ou modifier vos opérations existantes. Vous migrez des serveurs d'une plateforme sur site vers un service cloud pour la même plateforme. Exemple : migrer une Microsoft Hyper-V application vers AWS.
- Retenir : conservez les applications dans votre environnement source. Il peut s'agir d'applications nécessitant une refactorisation majeure, que vous souhaitez retarder, et d'applications existantes que vous souhaitez retenir, car rien ne justifie leur migration sur le plan commercial.

- Retirer : mettez hors service ou supprimez les applications dont vous n'avez plus besoin dans votre environnement source.

A

ABAC

Voir contrôle d'accès basé sur les attributs.

services abstraits

Consultez la section Services gérés.

ACIDE

Voir atomicité, consistance, isolation, durabilité.

migration active-active

Méthode de migration de base de données dans laquelle la synchronisation des bases de données source et cible est maintenue (à l'aide d'un outil de réplication bidirectionnelle ou d'opérations d'écriture double), tandis que les deux bases de données gèrent les transactions provenant de la connexion d'applications pendant la migration. Cette méthode prend en charge la migration par petits lots contrôlés au lieu d'exiger un basculement ponctuel. Elle est plus flexible mais demande plus de travail qu'une migration active-passive.

migration active-passive

Méthode de migration de base de données dans laquelle les bases de données source et cible sont synchronisées, mais seule la base de données source gère les transactions liées à la connexion des applications pendant que les données sont répliquées vers la base de données cible. La base de données cible n'accepte aucune transaction pendant la migration.

fonction d'agrégation

Fonction SQL qui agit sur un groupe de lignes et calcule une valeur de retour unique pour le groupe. Des exemples de fonctions d'agrégation incluent `SUM` et`MAX`.

AI

Voir intelligence artificielle.

AIOps

Voir les opérations d'intelligence artificielle.

anonymisation

Processus de suppression définitive d'informations personnelles dans un ensemble de données. L'anonymisation peut contribuer à protéger la vie privée. Les données anonymisées ne sont plus considérées comme des données personnelles.

anti-motif

Solution fréquemment utilisée pour un problème récurrent lorsque la solution est contre-productive, inefficace ou moins efficace qu'une alternative.

contrôle des applications

Une approche de sécurité qui permet d'utiliser uniquement des applications approuvées afin de protéger un système contre les logiciels malveillants.

portefeuille d'applications

Ensemble d'informations détaillées sur chaque application utilisée par une organisation, y compris le coût de génération et de maintenance de l'application, ainsi que sa valeur métier. Ces informations sont essentielles pour le processus de découverte et d'analyse du portefeuille et permettent d'identifier et de prioriser les applications à migrer, à moderniser et à optimiser.

intelligence artificielle (IA)

Domaine de l'informatique consacré à l'utilisation des technologies de calcul pour exécuter des fonctions cognitives généralement associées aux humains, telles que l'apprentissage, la résolution de problèmes et la reconnaissance de modèles. Pour plus d'informations, veuillez consulter Qu'est-ce que l'intelligence artificielle ?

opérations d'intelligence artificielle (AIOps)

Processus consistant à utiliser des techniques de machine learning pour résoudre les problèmes opérationnels, réduire les incidents opérationnels et les interventions humaines, mais aussi améliorer la qualité du service. Pour plus d'informations sur son AIOps utilisation dans la stratégie de AWS migration, consultez le guide d'intégration des opérations.

chiffrement asymétrique

Algorithme de chiffrement qui utilise une paire de clés, une clé publique pour le chiffrement et une clé privée pour le déchiffrement. Vous pouvez partager la clé publique, car elle n'est pas utilisée pour le déchiffrement, mais l'accès à la clé privée doit être très restreint.

atomicité, cohérence, isolement, durabilité (ACID)

Ensemble de propriétés logicielles garantissant la validité des données et la fiabilité opérationnelle d'une base de données, même en cas d'erreur, de panne de courant ou d'autres problèmes.

contrôle d'accès par attributs (ABAC)

Pratique qui consiste à créer des autorisations détaillées en fonction des attributs de l'utilisateur, tels que le service, le poste et le nom de l'équipe. Pour plus d'informations, consultez ABAC pour AWS dans la documentation AWS Identity and Access Management (IAM).

source de données faisant autorité

Emplacement où vous stockez la version principale des données, considérée comme la source d'information la plus fiable. Vous pouvez copier les données de la source de données officielle vers d'autres emplacements à des fins de traitement ou de modification des données, par exemple en les anonymisant, en les expurgant ou en les pseudonymisant.

Zone de disponibilité

Un emplacement distinct au sein d'un Région AWS réseau isolé des défaillances dans d'autres zones de disponibilité et fournissant une connectivité réseau peu coûteuse et à faible latence aux autres zones de disponibilité de la même région.

AWS Cadre d'adoption du cloud (AWS CAF)

Un cadre de directives et de meilleures pratiques visant AWS à aider les entreprises à élaborer un plan efficace pour réussir leur migration vers le cloud. AWS La CAF organise ses conseils en six domaines prioritaires appelés perspectives : les affaires, les personnes, la gouvernance, les plateformes, la sécurité et les opérations. Les perspectives d'entreprise, de personnes et de gouvernance mettent l'accent sur les compétences et les processus métier, tandis que les perspectives relatives à la plateforme, à la sécurité et aux opérations se concentrent sur les compétences et les processus techniques. Par exemple, la perspective liée aux personnes cible les parties prenantes qui s'occupent des ressources humaines (RH), des fonctions de dotation en personnel et de la gestion des personnes. Dans cette perspective, la AWS CAF fournit des conseils pour le développement du personnel, la formation et les communications afin de préparer l'organisation à une adoption réussie du cloud. Pour plus d'informations, veuillez consulter le site Web AWS CAF et le livre blanc AWS CAF.

AWS Cadre de qualification de la charge de travail (AWS WQF)

Outil qui évalue les charges de travail liées à la migration des bases de données, recommande des stratégies de migration et fournit des estimations de travail. AWS Le WQF est inclus avec

AWS Schema Conversion Tool (AWS SCT). Il analyse les schémas de base de données et les objets de code, le code d'application, les dépendances et les caractéristiques de performance, et fournit des rapports d'évaluation.

B

mauvais bot

Un bot destiné à perturber ou à nuire à des individus ou à des organisations.

BCP

Consultez la section Planification de la continuité des activités.

graphique de comportement

Vue unifiée et interactive des comportements des ressources et des interactions au fil du temps. Vous pouvez utiliser un graphique de comportement avec Amazon Detective pour examiner les tentatives de connexion infructueuses, les appels d'API suspects et les actions similaires. Pour plus d'informations, veuillez consulter Data in a behavior graph dans la documentation Detective.

système de poids fort

Système qui stocke d'abord l'octet le plus significatif. Voir aussi endianité.

classification binaire

Processus qui prédit un résultat binaire (l'une des deux classes possibles). Par exemple, votre modèle de machine learning peut avoir besoin de prévoir des problèmes tels que « Cet e-mail est-il du spam ou non ? » ou « Ce produit est-il un livre ou une voiture ? ».

filtre de Bloom

Structure de données probabiliste et efficace en termes de mémoire qui est utilisée pour tester si un élément fait partie d'un ensemble.

déploiement bleu/vert

Stratégie de déploiement dans laquelle vous créez deux environnements distincts mais identiques. Vous exécutez la version actuelle de l'application dans un environnement (bleu) et la nouvelle version de l'application dans l'autre environnement (vert). Cette stratégie vous permet de revenir rapidement en arrière avec un impact minimal.

bot

Application logicielle qui exécute des tâches automatisées sur Internet et simule l'activité ou l'interaction humaine. Certains robots sont utiles ou bénéfiques, comme les robots d'exploration Web qui indexent des informations sur Internet. D'autres robots, appelés « bots malveillants », sont destinés à perturber ou à nuire à des individus ou à des organisations.

botnet

Réseaux de robots infectés par des logiciels malveillants et contrôlés par une seule entité, connue sous le nom d'herder ou d'opérateur de bots. Les botnets sont le mécanisme le plus connu pour faire évoluer les bots et leur impact.

branche

Zone contenue d'un référentiel de code. La première branche créée dans un référentiel est la branche principale. Vous pouvez créer une branche à partir d'une branche existante, puis développer des fonctionnalités ou corriger des bogues dans la nouvelle branche. Une branche que vous créez pour générer une fonctionnalité est communément appelée branche de fonctionnalités. Lorsque la fonctionnalité est prête à être publiée, vous fusionnez à nouveau la branche de fonctionnalités dans la branche principale. Pour plus d'informations, consultez À propos des branches (GitHub documentation).

accès par brise-vitre

Dans des circonstances exceptionnelles et par le biais d'un processus approuvé, il s'agit d'un moyen rapide pour un utilisateur d'accéder à un accès auquel Compte AWS il n'est généralement pas autorisé. Pour plus d'informations, consultez l'indicateur Implementation break-glass procedures dans le guide Well-Architected AWS .

stratégie existante (brownfield)

L'infrastructure existante de votre environnement. Lorsque vous adoptez une stratégie existante pour une architecture système, vous concevez l'architecture en fonction des contraintes des systèmes et de l'infrastructure actuels. Si vous étendez l'infrastructure existante, vous pouvez combiner des politiques brownfield (existantes) et greenfield (inédites).

cache de tampon

Zone de mémoire dans laquelle sont stockées les données les plus fréquemment consultées.

capacité métier

Ce que fait une entreprise pour générer de la valeur (par exemple, les ventes, le service client ou le marketing). Les architectures de microservices et les décisions de développement

peuvent être dictées par les capacités métier. Pour plus d'informations, veuillez consulter la section Organisation en fonction des capacités métier du livre blanc Exécution de microservices conteneurisés sur AWS.

planification de la continuité des activités (BCP)

Plan qui tient compte de l'impact potentiel d'un événement perturbateur, tel qu'une migration à grande échelle, sur les opérations, et qui permet à une entreprise de reprendre ses activités rapidement.

C

CAF

Voir le cadre d'adoption du AWS cloud.

déploiement de Canary

Diffusion lente et progressive d'une version pour les utilisateurs finaux. Lorsque vous êtes sûr, vous déployez la nouvelle version et remplacez la version actuelle dans son intégralité.

CCo E

Voir le Centre d'excellence du cloud.

CDC

Consultez la section Capture des données de modification.

capture des données de modification (CDC)

Processus de suivi des modifications apportées à une source de données, telle qu'une table de base de données, et d'enregistrement des métadonnées relatives à ces modifications. Vous pouvez utiliser la CDC à diverses fins, telles que l'audit ou la réplication des modifications dans un système cible afin de maintenir la synchronisation.

ingénierie du chaos

Introduire intentionnellement des défaillances ou des événements perturbateurs pour tester la résilience d'un système. Vous pouvez utiliser AWS Fault Injection Service (AWS FIS) pour effectuer des expériences qui stressent vos AWS charges de travail et évaluer leur réponse.

CI/CD

Découvrez l'intégration continue et la livraison continue.

classification

Processus de catégorisation qui permet de générer des prédictions. Les modèles de ML pour les problèmes de classification prédisent une valeur discrète. Les valeurs discrètes se distinguent toujours les unes des autres. Par exemple, un modèle peut avoir besoin d'évaluer la présence ou non d'une voiture sur une image.

chiffrement côté client

Chiffrement des données localement, avant que la cible ne les Service AWS reçoive.

Centre d'excellence du cloud (CCoE)

Une équipe multidisciplinaire qui dirige les efforts d'adoption du cloud au sein d'une organisation, notamment en développant les bonnes pratiques en matière de cloud, en mobilisant des ressources, en établissant des délais de migration et en guidant l'organisation dans le cadre de transformations à grande échelle. Pour plus d'informations, consultez les CCoarticles électroniques du blog sur la stratégie AWS Cloud d'entreprise.

cloud computing

Technologie cloud généralement utilisée pour le stockage de données à distance et la gestion des appareils IoT. Le cloud computing est généralement associé à la technologie informatique de pointe.

modèle d'exploitation du cloud

Dans une organisation informatique, modèle d'exploitation utilisé pour créer, faire évoluer et optimiser un ou plusieurs environnements cloud. Pour plus d'informations, consultez la section Création de votre modèle d'exploitation cloud.

étapes d'adoption du cloud

Les quatre phases que les entreprises traversent généralement lorsqu'elles migrent vers AWS Cloud :

- Projet : exécution de quelques projets liés au cloud à des fins de preuve de concept et d'apprentissage
- Base : réaliser des investissements fondamentaux pour accélérer votre adoption du cloud (par exemple, créer une zone de landing zone, définir un CCo E, établir un modèle opérationnel)
- Migration : migration d'applications individuelles
- Réinvention : optimisation des produits et services et innovation dans le cloud

Ces étapes ont été définies par Stephen Orban dans le billet de blog The Journey Toward Cloud-First & the Stages of Adoption publié sur le blog AWS Cloud Enterprise Strategy. Pour plus d'informations sur leur lien avec la stratégie de AWS migration, consultez le guide de préparation à la migration.

CMDB

Consultez la base de données de gestion des configurations.

référentiel de code

Emplacement où le code source et d'autres ressources, comme la documentation, les exemples et les scripts, sont stockés et mis à jour par le biais de processus de contrôle de version. Les référentiels cloud courants incluent GitHub ouBitbucket Cloud. Chaque version du code est appelée branche. Dans une structure de microservice, chaque référentiel est consacré à une seule fonctionnalité. Un seul pipeline CI/CD peut utiliser plusieurs référentiels.

cache passif

Cache tampon vide, mal rempli ou contenant des données obsolètes ou non pertinentes. Cela affecte les performances, car l'instance de base de données doit lire à partir de la mémoire principale ou du disque, ce qui est plus lent que la lecture à partir du cache tampon.

données gelées

Données rarement consultées et généralement historiques. Lorsque vous interrogez ce type de données, les requêtes lentes sont généralement acceptables. Le transfert de ces données vers des niveaux ou classes de stockage moins performants et moins coûteux peut réduire les coûts.

vision par ordinateur (CV)

Domaine de l'IA qui utilise l'apprentissage automatique pour analyser et extraire des informations à partir de formats visuels tels que des images numériques et des vidéos. Par exemple, Amazon SageMaker AI fournit des algorithmes de traitement d'image pour les CV.

dérive de configuration

Pour une charge de travail, une modification de configuration par rapport à l'état attendu. Cela peut entraîner une non-conformité de la charge de travail, et cela est généralement progressif et involontaire.

base de données de gestion des configurations (CMDB)

Référentiel qui stocke et gère les informations relatives à une base de données et à son environnement informatique, y compris les composants matériels et logiciels ainsi que leurs

configurations. Vous utilisez généralement les données d'une CMDB lors de la phase de découverte et d'analyse du portefeuille de la migration.

pack de conformité

Ensemble de AWS Config règles et d'actions correctives que vous pouvez assembler pour personnaliser vos contrôles de conformité et de sécurité. Vous pouvez déployer un pack de conformité en tant qu'entité unique dans une région Compte AWS et, ou au sein d'une organisation, à l'aide d'un modèle YAML. Pour plus d'informations, consultez la section Packs de conformité dans la AWS Config documentation.

intégration continue et livraison continue (CI/CD)

Processus d'automatisation des étapes de source, de construction, de test, de préparation et de production du processus de publication du logiciel. CI/CD est communément décrit comme un pipeline. CI/CD peut vous aider à automatiser les processus, à améliorer la productivité, à améliorer la qualité du code et à accélérer les livraisons. Pour plus d'informations, veuillez consulter Avantages de la livraison continue. CD peut également signifier déploiement continu. Pour plus d'informations, veuillez consulter Livraison continue et déploiement continu.

CV

Voir vision par ordinateur.

D

données au repos

Données stationnaires dans votre réseau, telles que les données stockées.

classification des données

Processus permettant d'identifier et de catégoriser les données de votre réseau en fonction de leur sévérité et de leur sensibilité. Il s'agit d'un élément essentiel de toute stratégie de gestion des risques de cybersécurité, car il vous aide à déterminer les contrôles de protection et de conservation appropriés pour les données. La classification des données est une composante du pilier de sécurité du AWS Well-Architected Framework. Pour plus d'informations, veuillez consulter Classification des données.

dérive des données

Une variation significative entre les données de production et les données utilisées pour entraîner un modèle ML, ou une modification significative des données d'entrée au fil du temps. La dérive

des données peut réduire la qualité, la précision et l'équité globales des prédictions des modèles ML.

données en transit

Données qui circulent activement sur votre réseau, par exemple entre les ressources du réseau.

maillage de données

Un cadre architectural qui fournit une propriété des données distribuée et décentralisée avec une gestion et une gouvernance centralisées.

minimisation des données

Le principe de collecte et de traitement des seules données strictement nécessaires. La pratique de la minimisation des données AWS Cloud peut réduire les risques liés à la confidentialité, les coûts et l'empreinte carbone de vos analyses.

périmètre de données

Ensemble de garde-fous préventifs dans votre AWS environnement qui permettent de garantir que seules les identités fiables accèdent aux ressources fiables des réseaux attendus. Pour plus d'informations, voir Création d'un périmètre de données sur AWS.

prétraitement des données

Pour transformer les données brutes en un format facile à analyser par votre modèle de ML. Le prétraitement des données peut impliquer la suppression de certaines colonnes ou lignes et le traitement des valeurs manquantes, incohérentes ou en double.

provenance des données

Le processus de suivi de l'origine et de l'historique des données tout au long de leur cycle de vie, par exemple la manière dont les données ont été générées, transmises et stockées.

sujet des données

Personne dont les données sont collectées et traitées.

entrepôt des données

Un système de gestion des données qui prend en charge les informations commerciales, telles que les analyses. Les entrepôts de données contiennent généralement de grandes quantités de données historiques et sont généralement utilisés pour les requêtes et les analyses.

langage de définition de base de données (DDL)

Instructions ou commandes permettant de créer ou de modifier la structure des tables et des objets dans une base de données.

langage de manipulation de base de données (DML)

Instructions ou commandes permettant de modifier (insérer, mettre à jour et supprimer) des informations dans une base de données.

DDL

Voir langage de définition de base de données.

ensemble profond

Sert à combiner plusieurs modèles de deep learning à des fins de prédiction. Vous pouvez utiliser des ensembles profonds pour obtenir une prévision plus précise ou pour estimer l'incertitude des prédictions.

deep learning

Un sous-champ de ML qui utilise plusieurs couches de réseaux neuronaux artificiels pour identifier le mappage entre les données d'entrée et les variables cibles d'intérêt.

defense-in-depth

Approche de la sécurité de l'information dans laquelle une série de mécanismes et de contrôles de sécurité sont judicieusement répartis sur l'ensemble d'un réseau informatique afin de protéger la confidentialité, l'intégrité et la disponibilité du réseau et des données qu'il contient. Lorsque vous adoptez cette stratégie AWS, vous ajoutez plusieurs contrôles à différentes couches de la AWS Organizations structure afin de sécuriser les ressources. Par exemple, une defense-in-depth approche peut combiner l'authentification multifactorielle, la segmentation du réseau et le chiffrement.

administrateur délégué

Dans AWS Organizations, un service compatible peut enregistrer un compte AWS membre pour administrer les comptes de l'organisation et gérer les autorisations pour ce service. Ce compte est appelé administrateur délégué pour ce service. Pour plus d'informations et une liste des services compatibles, veuillez consulter la rubrique Services qui fonctionnent avec AWS Organizations dans la documentation AWS Organizations .

déploiement

Processus de mise à disposition d'une application, de nouvelles fonctionnalités ou de corrections de code dans l'environnement cible. Le déploiement implique la mise en œuvre de modifications dans une base de code, puis la génération et l'exécution de cette base de code dans les environnements de l'application.

environnement de développement

Voir environnement.

contrôle de détection

Contrôle de sécurité conçu pour détecter, journaliser et alerter après la survenue d'un événement. Ces contrôles constituent une deuxième ligne de défense et vous alertent en cas d'événements de sécurité qui ont contourné les contrôles préventifs en place. Pour plus d'informations, veuillez consulter la rubrique Contrôles de détection dans Implementing security controls on AWS.

cartographie de la chaîne de valeur du développement (DVSM)

Processus utilisé pour identifier et hiérarchiser les contraintes qui nuisent à la rapidité et à la qualité du cycle de vie du développement logiciel. DVSM étend le processus de cartographie de la chaîne de valeur initialement conçu pour les pratiques de production allégée. Il met l'accent sur les étapes et les équipes nécessaires pour créer et transférer de la valeur tout au long du processus de développement logiciel.

jumeau numérique

Représentation virtuelle d'un système réel, tel qu'un bâtiment, une usine, un équipement industriel ou une ligne de production. Les jumeaux numériques prennent en charge la maintenance prédictive, la surveillance à distance et l'optimisation de la production.

tableau des dimensions

Dans un schéma en étoile, table plus petite contenant les attributs de données relatifs aux données quantitatives d'une table de faits. Les attributs des tables de dimensions sont généralement des champs de texte ou des nombres discrets qui se comportent comme du texte. Ces attributs sont couramment utilisés pour la contrainte des requêtes, le filtrage et l'étiquetage des ensembles de résultats.

catastrophe

Un événement qui empêche une charge de travail ou un système d'atteindre ses objectifs commerciaux sur son site de déploiement principal. Ces événements peuvent être des

catastrophes naturelles, des défaillances techniques ou le résultat d'actions humaines, telles qu'une mauvaise configuration involontaire ou une attaque de logiciel malveillant.

reprise après sinistre (DR)

La stratégie et le processus que vous utilisez pour minimiser les temps d'arrêt et les pertes de données causés par un sinistre. Pour plus d'informations, consultez Disaster Recovery of Workloads on AWS : Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Voir langage de manipulation de base de données.

conception axée sur le domaine

Approche visant à développer un système logiciel complexe en connectant ses composants à des domaines évolutifs, ou objectifs métier essentiels, que sert chaque composant. Ce concept a été introduit par Eric Evans dans son ouvrage Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston : Addison-Wesley Professional, 2003). Pour plus d'informations sur l'utilisation du design piloté par domaine avec le modèle de figuier étrangleur, veuillez consulter Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

Voir reprise après sinistre.

détection de dérive

Suivi des écarts par rapport à une configuration de référence. Par exemple, vous pouvez l'utiliser AWS CloudFormation pour détecter la dérive des ressources du système ou AWS Control Tower pour détecter les modifications de votre zone d'atterrissage susceptibles d'affecter le respect des exigences de gouvernance.

DVSM

Voir la cartographie de la chaîne de valeur du développement.

E

EDA

Voir analyse exploratoire des données.

EDI

Voir échange de données informatisé.

informatique de périphérie

Technologie qui augmente la puissance de calcul des appareils intelligents en périphérie d'un réseau IoT. Comparé au cloud computing, l'informatique de pointe peut réduire la latence des communications et améliorer le temps de réponse.

échange de données informatisé (EDI)

L'échange automatique de documents commerciaux entre les organisations. Pour plus d'informations, voir Qu'est-ce que l'échange de données informatisé ?

chiffrement

Processus informatique qui transforme des données en texte clair, lisibles par l'homme, en texte chiffré.

clé de chiffrement

Chaîne cryptographique de bits aléatoires générée par un algorithme cryptographique. La longueur des clés peut varier, et chaque clé est conçue pour être imprévisible et unique.

endianisme

Ordre selon lequel les octets sont stockés dans la mémoire de l'ordinateur. Les systèmes de poids fort stockent d'abord l'octet le plus significatif. Les systèmes de poids faible stockent d'abord l'octet le moins significatif.

point de terminaison

Voir point de terminaison de service.

service de point de terminaison

Service que vous pouvez héberger sur un cloud privé virtuel (VPC) pour le partager avec d'autres utilisateurs. Vous pouvez créer un service de point de terminaison avec AWS PrivateLink et accorder des autorisations à d'autres Comptes AWS ou à AWS Identity and Access Management (IAM) principaux. Ces comptes ou principaux peuvent se connecter à votre service de point de terminaison de manière privée en créant des points de terminaison d'un VPC d'interface. Pour plus d'informations, veuillez consulter Création d'un service de point de terminaison dans la documentation Amazon Virtual Private Cloud (Amazon VPC).

planification des ressources d'entreprise (ERP)

Système qui automatise et gère les principaux processus métier (tels que la comptabilité, le MES et la gestion de projet) pour une entreprise.

chiffrement d'enveloppe

Processus de chiffrement d'une clé de chiffrement à l'aide d'une autre clé de chiffrement. Pour plus d'informations, consultez la section Chiffrement des enveloppes dans la documentation AWS Key Management Service (AWS KMS).

environment

Instance d'une application en cours d'exécution. Les types d'environnement les plus courants dans le cloud computing sont les suivants :

- Environnement de développement : instance d'une application en cours d'exécution à laquelle seule l'équipe principale chargée de la maintenance de l'application peut accéder. Les environnements de développement sont utilisés pour tester les modifications avant de les promouvoir dans les environnements supérieurs. Ce type d'environnement est parfois appelé environnement de test.
- Environnements inférieurs : tous les environnements de développement d'une application, tels que ceux utilisés pour les générations et les tests initiaux.
- Environnement de production : instance d'une application en cours d'exécution à laquelle les utilisateurs finaux peuvent accéder. Dans un CI/CD pipeline, l'environnement de production est le dernier environnement de déploiement.
- Environnements supérieurs : tous les environnements accessibles aux utilisateurs autres que l'équipe de développement principale. Ils peuvent inclure un environnement de production, des environnements de préproduction et des environnements pour les tests d'acceptation par les utilisateurs.

épopée

Dans les méthodologies agiles, catégories fonctionnelles qui aident à organiser et à prioriser votre travail. Les épopées fournissent une description détaillée des exigences et des tâches d'implémentation. Par exemple, les points forts de la AWS CAF en matière de sécurité incluent la gestion des identités et des accès, les contrôles de détection, la sécurité des infrastructures, la protection des données et la réponse aux incidents. Pour plus d'informations sur les épopées dans la stratégie de migration AWS , veuillez consulter le guide d'implémentation du programme.

ERP

Voir Planification des ressources d'entreprise.

analyse exploratoire des données (EDA)

Processus d'analyse d'un jeu de données pour comprendre ses principales caractéristiques. Vous collectez ou agrégez des données, puis vous effectuez des enquêtes initiales pour trouver des modèles, détecter des anomalies et vérifier les hypothèses. L'EDA est réalisée en calculant des statistiques récapitulatives et en créant des visualisations de données.

F

tableau des faits

La table centrale dans un schéma en étoile. Il stocke des données quantitatives sur les opérations commerciales. Généralement, une table de faits contient deux types de colonnes : celles qui contiennent des mesures et celles qui contiennent une clé étrangère pour une table de dimensions.

échouer rapidement

Une philosophie qui utilise des tests fréquents et progressifs pour réduire le cycle de vie du développement. C'est un élément essentiel d'une approche agile.

limite d'isolation des défauts

Dans le AWS Cloud, une limite telle qu'une zone de disponibilité Région AWS, un plan de contrôle ou un plan de données qui limite l'effet d'une panne et contribue à améliorer la résilience des charges de travail. Pour plus d'informations, consultez la section Limites d'isolation des AWS pannes.

branche de fonctionnalités

Voir succursale.

fonctionnalités

Les données d'entrée que vous utilisez pour faire une prédiction. Par exemple, dans un contexte de fabrication, les fonctionnalités peuvent être des images capturées périodiquement à partir de la ligne de fabrication.

importance des fonctionnalités

Le niveau d'importance d'une fonctionnalité pour les prédictions d'un modèle. Il s'exprime généralement sous la forme d'un score numérique qui peut être calculé à l'aide de différentes techniques, telles que la méthode Shapley Additive Explanations (SHAP) et les gradients intégrés. Pour plus d'informations, voir Interprétabilité du modèle d'apprentissage automatique avec AWS.

transformation de fonctionnalité

Optimiser les données pour le processus de ML, notamment en enrichissant les données avec des sources supplémentaires, en mettant à l'échelle les valeurs ou en extrayant plusieurs ensembles d'informations à partir d'un seul champ de données. Cela permet au modèle de ML de tirer parti des données. Par exemple, si vous décomposez la date « 2021-05-27 00:15:37 » en « 2021 », « mai », « jeudi » et « 15 », vous pouvez aider l'algorithme d'apprentissage à apprendre des modèles nuancés associés à différents composants de données.

invitation en quelques coups

Fournir à un LLM un petit nombre d'exemples illustrant la tâche et le résultat souhaité avant de lui demander d'effectuer une tâche similaire. Cette technique est une application de l'apprentissage contextuel, dans le cadre de laquelle les modèles apprennent à partir d'exemples (prises de vue) intégrés dans des instructions. Les instructions en quelques clics peuvent être efficaces pour les tâches qui nécessitent un formatage, un raisonnement ou des connaissances de domaine spécifiques. Voir également l'invite Zero-Shot.

FGAC

Découvrez le contrôle d'accès détaillé.

contrôle d'accès détaillé (FGAC)

Utilisation de plusieurs conditions pour autoriser ou refuser une demande d'accès.

migration instantanée (flash-cut)

Méthode de migration de base de données qui utilise la réplication continue des données par le biais de la capture des données de modification afin de migrer les données dans les plus brefs délais, au lieu d'utiliser une approche progressive. L'objectif est de réduire au maximum les temps d'arrêt.

FM

Voir le modèle de fondation.

modèle de fondation (FM)

Un vaste réseau neuronal d'apprentissage profond qui s'est entraîné sur d'énormes ensembles de données généralisées et non étiquetées. FMs sont capables d'effectuer une grande variété de tâches générales, telles que comprendre le langage, générer du texte et des images et converser en langage naturel. Pour plus d'informations, voir Que sont les modèles de base ?

G

IA générative

Sous-ensemble de modèles d'IA qui ont été entraînés sur de grandes quantités de données et qui peuvent utiliser une simple invite textuelle pour créer de nouveaux contenus et artefacts, tels que des images, des vidéos, du texte et du son. Pour plus d'informations, consultez Qu'est-ce que l'IA générative.

blocage géographique

Voir les restrictions géographiques.

restrictions géographiques (blocage géographique)

Sur Amazon CloudFront, option permettant d'empêcher les utilisateurs de certains pays d'accéder aux distributions de contenu. Vous pouvez utiliser une liste d'autorisation ou une liste de blocage pour spécifier les pays approuvés et interdits. Pour plus d'informations, consultez la section Restreindre la distribution géographique de votre contenu dans la CloudFront documentation.

Flux de travail Gitflow

Approche dans laquelle les environnements inférieurs et supérieurs utilisent différentes branches dans un référentiel de code source. Le flux de travail Gitflow est considéré comme existant, et le flux de travail basé sur les troncs est l'approche moderne préférée.

image dorée

Un instantané d'un système ou d'un logiciel utilisé comme modèle pour déployer de nouvelles instances de ce système ou logiciel. Par exemple, dans le secteur de la fabrication, une image dorée peut être utilisée pour fournir des logiciels sur plusieurs appareils et contribue à améliorer la vitesse, l'évolutivité et la productivité des opérations de fabrication des appareils.

stratégie inédite

L'absence d'infrastructures existantes dans un nouvel environnement. Lorsque vous adoptez une stratégie inédite pour une architecture système, vous pouvez sélectionner toutes les nouvelles technologies sans restriction de compatibilité avec l'infrastructure existante, également appelée brownfield. Si vous étendez l'infrastructure existante, vous pouvez combiner des politiques brownfield (existantes) et greenfield (inédites).

barrière de protection

Règle de haut niveau qui permet de régir les ressources, les politiques et la conformité au sein des unités organisationnelles (OUs). Les barrières de protection préventives appliquent des politiques pour garantir l'alignement sur les normes de conformité. Elles sont mises en œuvre à l'aide de politiques de contrôle des services et de limites des autorisations IAM. Les barrières de protection de détection détectent les violations des politiques et les problèmes de conformité, et génèrent des alertes pour y remédier. Ils sont implémentés à l'aide d'Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, d'Amazon Inspector et de AWS Lambda contrôles personnalisés.

H

HA

Découvrez la haute disponibilité.

migration de base de données hétérogène

Migration de votre base de données source vers une base de données cible qui utilise un moteur de base de données différent (par exemple, Oracle vers Amazon Aurora). La migration hétérogène fait généralement partie d'un effort de réarchitecture, et la conversion du schéma peut s'avérer une tâche complexe. AWS propose AWS SCT qui facilite les conversions de schémas.

haute disponibilité (HA)

Capacité d'une charge de travail à fonctionner en continu, sans intervention, en cas de difficultés ou de catastrophes. Les systèmes HA sont conçus pour basculer automatiquement, fournir constamment des performances de haute qualité et gérer différentes charges et défaillances avec un impact minimal sur les performances.

modernisation des historiens

Approche utilisée pour moderniser et mettre à niveau les systèmes de technologie opérationnelle (OT) afin de mieux répondre aux besoins de l'industrie manufacturière. Un historien est un type de base de données utilisé pour collecter et stocker des données provenant de diverses sources dans une usine.

données de rétention

Partie de données historiques étiquetées qui n'est pas divulguée dans un ensemble de données utilisé pour entraîner un modèle d'apprentissage automatique. Vous pouvez utiliser les données de blocage pour évaluer les performances du modèle en comparant les prévisions du modèle aux données de blocage.

migration de base de données homogène

Migration de votre base de données source vers une base de données cible qui partage le même moteur de base de données (par exemple, Microsoft SQL Server vers Amazon RDS for SQL Server). La migration homogène s'inscrit généralement dans le cadre d'un effort de réhébergement ou de replateforme. Vous pouvez utiliser les utilitaires de base de données natifs pour migrer le schéma.

données chaudes

Données fréquemment consultées, telles que les données en temps réel ou les données translationnelles récentes. Ces données nécessitent généralement un niveau ou une classe de stockage à hautes performances pour fournir des réponses rapides aux requêtes.

correctif

Solution d'urgence à un problème critique dans un environnement de production. En raison de son urgence, un correctif est généralement créé en dehors du flux de travail de DevOps publication habituel.

période de soins intensifs

Immédiatement après le basculement, période pendant laquelle une équipe de migration gère et surveille les applications migrées dans le cloud afin de résoudre les problèmes éventuels. En règle générale, cette période dure de 1 à 4 jours. À la fin de la période de soins intensifs, l'équipe de migration transfère généralement la responsabilité des applications à l'équipe des opérations cloud.

I

IaC

Considérez l'infrastructure comme un code.

politique basée sur l’identité

Politique attachée à un ou plusieurs principaux IAM qui définit leurs autorisations au sein de l' AWS Cloud environnement.

application inactive

Application dont l'utilisation moyenne du processeur et de la mémoire se situe entre 5 et 20 % sur une période de 90 jours. Dans un projet de migration, il est courant de retirer ces applications ou de les retenir sur site.

IIo T

Voir Internet industriel des objets.

infrastructure immuable

Modèle qui déploie une nouvelle infrastructure pour les charges de travail de production au lieu de mettre à jour, d'appliquer des correctifs ou de modifier l'infrastructure existante. Les infrastructures immuables sont intrinsèquement plus cohérentes, fiables et prévisibles que les infrastructures mutables. Pour plus d'informations, consultez les meilleures pratiques de déploiement à l'aide d'une infrastructure immuable dans le AWS Well-Architected Framework.

VPC entrant (d'entrée)

Dans une architecture AWS multi-comptes, un VPC qui accepte, inspecte et achemine les connexions réseau depuis l'extérieur d'une application. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau avec les fonctions entrantes, sortantes et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

migration incrémentielle

Stratégie de basculement dans le cadre de laquelle vous migrez votre application par petites parties au lieu d'effectuer un basculement complet unique. Par exemple, il se peut que vous ne transfériez que quelques microservices ou utilisateurs vers le nouveau système dans un premier temps. Après avoir vérifié que tout fonctionne correctement, vous pouvez transférer

progressivement des microservices ou des utilisateurs supplémentaires jusqu'à ce que vous puissiez mettre hors service votre système hérité. Cette stratégie réduit les risques associés aux migrations de grande ampleur.

Industry 4.0

Un terme introduit par Klaus Schwab en 2016 pour désigner la modernisation des processus de fabrication grâce aux avancées en matière de connectivité, de données en temps réel, d'automatisation, d'analyse et d'IA/ML.

infrastructure

Ensemble des ressources et des actifs contenus dans l'environnement d'une application.

infrastructure en tant que code (IaC)

Processus de mise en service et de gestion de l'infrastructure d'une application via un ensemble de fichiers de configuration. IaC est conçue pour vous aider à centraliser la gestion de l'infrastructure, à normaliser les ressources et à mettre à l'échelle rapidement afin que les nouveaux environnements soient reproductibles, fiables et cohérents.

Internet industriel des objets (IIoT)

L'utilisation de capteurs et d'appareils connectés à Internet dans les secteurs industriels tels que la fabrication, l'énergie, l'automobile, les soins de santé, les sciences de la vie et l'agriculture. Pour plus d'informations, voir Élaboration d'une stratégie de transformation numérique de l'Internet des objets (IIoT) industriel.

VPC d'inspection

Dans une architecture AWS multi-comptes, un VPC centralisé qui gère les inspections du trafic réseau VPCs entre (identique ou Régions AWS différent), Internet et les réseaux locaux. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau avec les fonctions entrantes, sortantes et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

Internet des objets (IoT)

Réseau d'objets physiques connectés dotés de capteurs ou de processeurs intégrés qui communiquent avec d'autres appareils et systèmes via Internet ou via un réseau de communication local. Pour plus d'informations, veuillez consulter la section Qu'est-ce que l'IoT ?.

interprétabilité

Caractéristique d'un modèle de machine learning qui décrit dans quelle mesure un être humain peut comprendre comment les prédictions du modèle dépendent de ses entrées. Pour plus d'informations, voir Interprétabilité du modèle d'apprentissage automatique avec AWS.

IoT

Voir Internet des objets.

Bibliothèque d'informations informatiques (ITIL)

Ensemble de bonnes pratiques pour proposer des services informatiques et les aligner sur les exigences métier. L'ITIL constitue la base de l'ITSM.

gestion des services informatiques (ITSM)

Activités associées à la conception, à la mise en œuvre, à la gestion et à la prise en charge de services informatiques d'une organisation. Pour plus d'informations sur l'intégration des opérations cloud aux outils ITSM, veuillez consulter le guide d'intégration des opérations.

ITIL

Consultez la bibliothèque d'informations informatiques.

ITSM

Voir Gestion des services informatiques.

L

contrôle d'accès basé sur des étiquettes (LBAC)

Une implémentation du contrôle d'accès obligatoire (MAC) dans laquelle une valeur d'étiquette de sécurité est explicitement attribuée aux utilisateurs et aux données elles-mêmes. L'intersection entre l'étiquette de sécurité utilisateur et l'étiquette de sécurité des données détermine les lignes et les colonnes visibles par l'utilisateur.

zone de destination

Une zone d'atterrissage est un AWS environnement multi-comptes bien conçu, évolutif et sécurisé. Il s'agit d'un point de départ à partir duquel vos entreprises peuvent rapidement lancer et déployer des charges de travail et des applications en toute confiance dans leur environnement de sécurité et d'infrastructure. Pour plus d'informations sur les zones de destination, veuillez consulter Setting up a secure and scalable multi-account AWS environment.

grand modèle de langage (LLM)

Un modèle d'intelligence artificielle basé sur le deep learning qui est préentraîné sur une grande quantité de données. Un LLM peut effectuer plusieurs tâches, telles que répondre à des questions, résumer des documents, traduire du texte dans d'autres langues et compléter des phrases. Pour plus d'informations, voir Que sont LLMs.

migration de grande envergure

Migration de 300 serveurs ou plus.

LBAC

Voir contrôle d'accès basé sur des étiquettes.

principe de moindre privilège

Bonne pratique de sécurité qui consiste à accorder les autorisations minimales nécessaires à l'exécution d'une tâche. Pour plus d'informations, veuillez consulter la rubrique Accorder les autorisations de moindre privilège dans la documentation IAM.

lift and shift

Voir 7 Rs.

système de poids faible

Système qui stocke d'abord l'octet le moins significatif. Voir aussi endianité.

LLM

Voir le grand modèle de langage.

environnements inférieurs

Voir environnement.

M

machine learning (ML)

Type d'intelligence artificielle qui utilise des algorithmes et des techniques pour la reconnaissance et l'apprentissage de modèles. Le ML analyse et apprend à partir de données enregistrées, telles que les données de l'Internet des objets (IoT), pour générer un modèle statistique basé sur des modèles. Pour plus d'informations, veuillez consulter Machine Learning.

branche principale

Voir succursale.

malware

Logiciel conçu pour compromettre la sécurité ou la confidentialité de l'ordinateur. Les logiciels malveillants peuvent perturber les systèmes informatiques, divulguer des informations sensibles ou obtenir un accès non autorisé. Parmi les malwares, on peut citer les virus, les vers, les rançongiciels, les chevaux de Troie, les logiciels espions et les enregistreurs de frappe.

services gérés

Services AWS pour lequel AWS fonctionnent la couche d'infrastructure, le système d'exploitation et les plateformes, et vous accédez aux points de terminaison pour stocker et récupérer des données. Amazon Simple Storage Service (Amazon S3) et Amazon DynamoDB sont des exemples de services gérés. Ils sont également appelés services abstraits.

système d'exécution de la fabrication (MES)

Un système logiciel pour le suivi, la surveillance, la documentation et le contrôle des processus de production qui convertissent les matières premières en produits finis dans l'atelier.

MAP

Voir Migration Acceleration Program.

mécanisme

Processus complet au cours duquel vous créez un outil, favorisez son adoption, puis inspectez les résultats afin de procéder aux ajustements nécessaires. Un mécanisme est un cycle qui se renforce et s'améliore lorsqu'il fonctionne. Pour plus d'informations, voir Création de mécanismes dans le cadre AWS Well-Architected.

compte membre

Tous, à l' Comptes AWS exception du compte de gestion, qui font partie d'une organisation dans AWS Organizations. Un compte ne peut être membre que d'une seule organisation à la fois.

MAILLES

Voir le système d'exécution de la fabrication.

Transport télémétrique en file d'attente de messages (MQTT)

Protocole de communication léger machine-to-machine (M2M), basé sur le modèle de publication/d'abonnement, pour les appareils IoT aux ressources limitées.

microservice

Un petit service indépendant qui communique via un réseau bien défini APIs et qui est généralement détenu par de petites équipes autonomes. Par exemple, un système d'assurance peut inclure des microservices qui mappent à des capacités métier, telles que les ventes ou le marketing, ou à des sous-domaines, tels que les achats, les réclamations ou l'analytique. Les avantages des microservices incluent l'agilité, la flexibilité de la mise à l'échelle, la facilité de déploiement, la réutilisation du code et la résilience. Pour plus d'informations, consultez la section Intégration de microservices à l'aide de services AWS sans serveur.

architecture de microservices

Approche de création d'une application avec des composants indépendants qui exécutent chaque processus d'application en tant que microservice. Ces microservices communiquent via une interface bien définie en utilisant Lightweight. APIs Chaque microservice de cette architecture peut être mis à jour, déployé et mis à l'échelle pour répondre à la demande de fonctions spécifiques d'une application. Pour plus d'informations, consultez la section Implémentation de microservices sur AWS.

Programme d'accélération des migrations (MAP)

Un AWS programme qui fournit un support de conseil, des formations et des services pour aider les entreprises à établir une base opérationnelle solide pour passer au cloud, et pour aider à compenser le coût initial des migrations. MAP inclut une méthodologie de migration pour exécuter les migrations héritées de manière méthodique, ainsi qu'un ensemble d'outils pour automatiser et accélérer les scénarios de migration courants.

migration à grande échelle

Processus consistant à transférer la majeure partie du portefeuille d'applications vers le cloud par vagues, un plus grand nombre d'applications étant déplacées plus rapidement à chaque vague. Cette phase utilise les bonnes pratiques et les enseignements tirés des phases précédentes pour implémenter une usine de migration d'équipes, d'outils et de processus en vue de rationaliser la migration des charges de travail grâce à l'automatisation et à la livraison agile. Il s'agit de la troisième phase de la stratégie de migration AWS.

usine de migration

Équipes interfonctionnelles qui rationalisent la migration des charges de travail grâce à des approches automatisées et agiles. Les équipes de Migration Factory comprennent généralement des responsables des opérations, des analystes commerciaux et des propriétaires, des ingénieurs de migration, des développeurs et DevOps des professionnels travaillant dans le cadre de sprints.

Entre 20 et 50 % du portefeuille d'applications d'entreprise est constitué de modèles répétés qui peuvent être optimisés par une approche d'usine. Pour plus d'informations, veuillez consulter la rubrique discussion of migration factories et le guide Cloud Migration Factory dans cet ensemble de contenus.

métadonnées de migration

Informations relatives à l'application et au serveur nécessaires pour finaliser la migration. Chaque modèle de migration nécessite un ensemble de métadonnées de migration différent. Les exemples de métadonnées de migration incluent le sous-réseau cible, le groupe de sécurité et le AWS compte.

modèle de migration

Tâche de migration reproductible qui détaille la stratégie de migration, la destination de la migration et l'application ou le service de migration utilisé. Exemple : réorganisez la migration vers Amazon EC2 avec le service de migration AWS d'applications.

Évaluation du portefeuille de migration (MPA)

Outil en ligne qui fournit des informations pour valider l'analyse de rentabilisation en faveur de la migration vers le. AWS Cloud La MPA propose une évaluation détaillée du portefeuille (dimensionnement approprié des serveurs, tarification, comparaison du coût total de possession, analyse des coûts de migration), ainsi que la planification de la migration (analyse et collecte des données d'applications, regroupement des applications, priorisation des migrations et planification des vagues). L'outil MPA (connexion requise) est disponible gratuitement pour tous les AWS consultants et consultants APN Partner.

Évaluation de la préparation à la migration (MRA)

Processus qui consiste à obtenir des informations sur l'état de préparation d'une organisation au cloud, à identifier les forces et les faiblesses et à élaborer un plan d'action pour combler les lacunes identifiées, à l'aide du AWS CAF. Pour plus d'informations, veuillez consulter le guide de préparation à la migration. La MRA est la première phase de la stratégie de migration AWS.

stratégie de migration

L'approche utilisée pour migrer une charge de travail vers le AWS Cloud. Pour plus d'informations, reportez-vous aux 7 R de ce glossaire et à Mobiliser votre organisation pour accélérer les migrations à grande échelle.

ML

Voir apprentissage automatique.

modernisation

Transformation d'une application obsolète (héritée ou monolithique) et de son infrastructure en un système agile, élastique et hautement disponible dans le cloud afin de réduire les coûts, de gagner en efficacité et de tirer parti des innovations. Pour plus d'informations, consultez la section Stratégie de modernisation des applications dans le AWS Cloud.

évaluation de la préparation à la modernisation

Évaluation qui permet de déterminer si les applications d'une organisation sont prêtes à être modernisées, d'identifier les avantages, les risques et les dépendances, et qui détermine dans quelle mesure l'organisation peut prendre en charge l'état futur de ces applications. Le résultat de l'évaluation est un plan de l'architecture cible, une feuille de route détaillant les phases de développement et les étapes du processus de modernisation, ainsi qu'un plan d'action pour combler les lacunes identifiées. Pour plus d'informations, consultez la section Évaluation de l'état de préparation à la modernisation des applications dans le AWS Cloud.

applications monolithiques (monolithes)

Applications qui s'exécutent en tant que service unique avec des processus étroitement couplés. Les applications monolithiques ont plusieurs inconvénients. Si une fonctionnalité de l'application connaît un pic de demande, l'architecture entière doit être mise à l'échelle. L'ajout ou l'amélioration des fonctionnalités d'une application monolithique devient également plus complexe lorsque la base de code s'élargit. Pour résoudre ces problèmes, vous pouvez utiliser une architecture de microservices. Pour plus d'informations, veuillez consulter Decomposing monoliths into microservices.

MPA

Voir Évaluation du portefeuille de migration.

MQTT

Voir Message Queuing Telemetry Transport.

classification multi-classes

Processus qui permet de générer des prédictions pour plusieurs classes (prédiction d'un résultat parmi plus de deux). Par exemple, un modèle de ML peut demander « Ce produit est-il un livre, une voiture ou un téléphone ? » ou « Quelle catégorie de produits intéresse le plus ce client ? ».

infrastructure mutable

Modèle qui met à jour et modifie l'infrastructure existante pour les charges de travail de production. Pour améliorer la cohérence, la fiabilité et la prévisibilité, le AWS Well-Architected Framework recommande l'utilisation d'une infrastructure immuable comme meilleure pratique.

O

OAC

Voir Contrôle d'accès à l'origine.

OAI

Voir l'identité d'accès à l'origine.

OCM

Voir gestion du changement organisationnel.

migration hors ligne

Méthode de migration dans laquelle la charge de travail source est supprimée au cours du processus de migration. Cette méthode implique un temps d'arrêt prolongé et est généralement utilisée pour de petites charges de travail non critiques.

OI

Consultez la section Intégration des opérations.

OLA

Voir l'accord au niveau opérationnel.

migration en ligne

Méthode de migration dans laquelle la charge de travail source est copiée sur le système cible sans être mise hors ligne. Les applications connectées à la charge de travail peuvent continuer à fonctionner pendant la migration. Cette méthode implique un temps d'arrêt nul ou minimal et est généralement utilisée pour les charges de travail de production critiques.

OPC-UA

Voir Open Process Communications - Architecture unifiée.

Communications par processus ouvert - Architecture unifiée (OPC-UA)

Un protocole de communication machine-to-machine (M2M) pour l'automatisation industrielle. L'OPC-UA fournit une norme d'interopérabilité avec des schémas de cryptage, d'authentification et d'autorisation des données.

accord au niveau opérationnel (OLA)

Accord qui précise ce que les groupes informatiques fonctionnels s'engagent à fournir les uns aux autres, afin de prendre en charge un contrat de niveau de service (SLA).

examen de l'état de préparation opérationnelle (ORR)

Une liste de questions et de bonnes pratiques associées qui vous aident à comprendre, à évaluer, à prévenir ou à réduire l'ampleur des incidents et des défaillances possibles. Pour plus d'informations, voir Operational Readiness Reviews (ORR) dans le AWS Well-Architected Framework.

technologie opérationnelle (OT)

Systèmes matériels et logiciels qui fonctionnent avec l'environnement physique pour contrôler les opérations, les équipements et les infrastructures industriels. Dans le secteur manufacturier, l'intégration des systèmes OT et des technologies de l'information (IT) est au cœur des transformations de l'industrie 4.0.

intégration des opérations (OI)

Processus de modernisation des opérations dans le cloud, qui implique la planification de la préparation, l'automatisation et l'intégration. Pour en savoir plus, veuillez consulter le guide d'intégration des opérations.

journal de suivi d'organisation

Un parcours créé par AWS CloudTrail qui enregistre tous les événements pour tous les membres Comptes AWS d'une organisation dans AWS Organizations. Ce journal de suivi est créé dans chaque Compte AWS qui fait partie de l'organisation et suit l'activité de chaque compte. Pour plus d'informations, consultez la section Création d'un suivi pour une organisation dans la CloudTrail documentation.

gestion du changement organisationnel (OCM)

Cadre pour gérer les transformations métier majeures et perturbatrices du point de vue des personnes, de la culture et du leadership. L'OCM aide les organisations à se préparer et à effectuer la transition vers de nouveaux systèmes et de nouvelles politiques en accélérant

l'adoption des changements, en abordant les problèmes de transition et en favorisant des changements culturels et organisationnels. Dans la stratégie de AWS migration, ce cadre est appelé accélération du personnel, en raison de la rapidité du changement requise dans les projets d'adoption du cloud. Pour plus d'informations, veuillez consulter le guide OCM.

contrôle d'accès d'origine (OAC)

Dans CloudFront, une option améliorée pour restreindre l'accès afin de sécuriser votre contenu Amazon Simple Storage Service (Amazon S3). L'OAC prend en charge tous les compartiments S3 dans leur ensemble Régions AWS, le chiffrement côté serveur avec AWS KMS (SSE-KMS) et les requêtes dynamiques PUT adressées au compartiment S3. DELETE

identité d'accès d'origine (OAI)

Dans CloudFront, une option permettant de restreindre l'accès afin de sécuriser votre contenu Amazon S3. Lorsque vous utilisez OAI, il CloudFront crée un principal auprès duquel Amazon S3 peut s'authentifier. Les principaux authentifiés ne peuvent accéder au contenu d'un compartiment S3 que par le biais d'une distribution spécifique CloudFront . Voir également OAC, qui fournit un contrôle d'accès plus précis et amélioré.

ORR

Voir l'examen de l'état de préparation opérationnelle.

DE

Voir technologie opérationnelle.

VPC sortant (de sortie)

Dans une architecture AWS multi-comptes, un VPC qui gère les connexions réseau initiées depuis une application. L'architecture AWS de référence de sécurité recommande de configurer votre compte réseau avec les fonctions entrantes, sortantes et d'inspection VPCs afin de protéger l'interface bidirectionnelle entre votre application et l'Internet en général.

P

limite des autorisations

Politique de gestion IAM attachée aux principaux IAM pour définir les autorisations maximales que peut avoir l'utilisateur ou le rôle. Pour plus d'informations, veuillez consulter la rubrique Limites des autorisations dans la documentation IAM.

informations personnelles identifiables (PII)

Informations qui, lorsqu'elles sont consultées directement ou associées à d'autres données connexes, peuvent être utilisées pour déduire raisonnablement l'identité d'une personne. Les exemples d'informations personnelles incluent les noms, les adresses et les coordonnées.

PII

Voir les informations personnelles identifiables.

manuel stratégique

Ensemble d'étapes prédéfinies qui capturent le travail associé aux migrations, comme la fourniture de fonctions d'opérations de base dans le cloud. Un manuel stratégique peut revêtir la forme de scripts, de runbooks automatisés ou d'un résumé des processus ou des étapes nécessaires au fonctionnement de votre environnement modernisé.

PLC

Voir contrôleur logique programmable.

PLM

Consultez la section Gestion du cycle de vie des produits.

politique

Objet capable de définir les autorisations (voir la politique basée sur l'identité), de spécifier les conditions d'accès (voir la politique basée sur les ressources) ou de définir les autorisations maximales pour tous les comptes d'une organisation dans AWS Organizations (voir la politique de contrôle des services).

persistance polyglotte

Choix indépendant de la technologie de stockage de données d'un microservice en fonction des modèles d'accès aux données et d'autres exigences. Si vos microservices utilisent la même technologie de stockage de données, ils peuvent rencontrer des difficultés d'implémentation ou présenter des performances médiocres. Les microservices sont plus faciles à mettre en œuvre, atteignent de meilleures performances, ainsi qu'une meilleure capacité de mise à l'échelle s'ils utilisent l'entrepôt de données le mieux adapté à leurs besoins. Pour plus d'informations, veuillez consulter Enabling data persistence in microservices.

évaluation du portefeuille

Processus de découverte, d'analyse et de priorisation du portefeuille d'applications afin de planifier la migration. Pour plus d'informations, veuillez consulter Evaluating migration readiness.

predicate

Une condition de requête qui renvoie `true` ou`false`, généralement située dans une `WHERE` clause.

prédicat pushdown

Technique d'optimisation des requêtes de base de données qui filtre les données de la requête avant le transfert. Cela réduit la quantité de données qui doivent être extraites et traitées à partir de la base de données relationnelle et améliore les performances des requêtes.

contrôle préventif

Contrôle de sécurité conçu pour empêcher qu'un événement ne se produise. Ces contrôles constituent une première ligne de défense pour empêcher tout accès non autorisé ou toute modification indésirable de votre réseau. Pour plus d'informations, veuillez consulter Preventative controls dans Implementing security controls on AWS.

principal

Entité AWS capable d'effectuer des actions et d'accéder aux ressources. Cette entité est généralement un utilisateur root pour un Compte AWS rôle IAM ou un utilisateur. Pour plus d'informations, veuillez consulter la rubrique Principal dans Termes et concepts relatifs aux rôles, dans la documentation IAM.

confidentialité dès la conception

Une approche d'ingénierie système qui prend en compte la confidentialité tout au long du processus de développement.

zones hébergées privées

Conteneur contenant des informations sur la manière dont vous souhaitez qu'Amazon Route 53 réponde aux requêtes DNS pour un domaine et ses sous-domaines au sein d'un ou de plusieurs VPCs domaines. Pour plus d'informations, veuillez consulter Working with private hosted zones dans la documentation Route 53.

contrôle proactif

Contrôle de sécurité conçu pour empêcher le déploiement de ressources non conformes. Ces contrôles analysent les ressources avant qu'elles ne soient provisionnées. Si la ressource n'est pas conforme au contrôle, elle n'est pas provisionnée. Pour plus d'informations, consultez le guide de référence sur les contrôles dans la AWS Control Tower documentation et consultez la section Contrôles proactifs dans Implémentation des contrôles de sécurité sur AWS.

gestion du cycle de vie des produits (PLM)

Gestion des données et des processus d'un produit tout au long de son cycle de vie, depuis la conception, le développement et le lancement, en passant par la croissance et la maturité, jusqu'au déclin et au retrait.

environnement de production

Voir environnement.

contrôleur logique programmable (PLC)

Dans le secteur manufacturier, un ordinateur hautement fiable et adaptable qui surveille les machines et automatise les processus de fabrication.

chaînage rapide

Utiliser le résultat d'une invite LLM comme entrée pour l'invite suivante afin de générer de meilleures réponses. Cette technique est utilisée pour décomposer une tâche complexe en sous-tâches ou pour affiner ou développer de manière itérative une réponse préliminaire. Cela permet d'améliorer la précision et la pertinence des réponses d'un modèle et permet d'obtenir des résultats plus précis et personnalisés.

pseudonymisation

Processus de remplacement des identifiants personnels dans un ensemble de données par des valeurs fictives. La pseudonymisation peut contribuer à protéger la vie privée. Les données pseudonymisées sont toujours considérées comme des données personnelles.

publish/subscribe (pub/sub)

Modèle qui permet des communications asynchrones entre les microservices afin d'améliorer l'évolutivité et la réactivité. Par exemple, dans un MES basé sur des microservices, un microservice peut publier des messages d'événements sur un canal auquel d'autres microservices peuvent s'abonner. Le système peut ajouter de nouveaux microservices sans modifier le service de publication.

Q

plan de requête

Série d'étapes, telles que des instructions, utilisées pour accéder aux données d'un système de base de données relationnelle SQL.

régression du plan de requêtes

Le cas où un optimiseur de service de base de données choisit un plan moins optimal qu'avant une modification donnée de l'environnement de base de données. Cela peut être dû à des changements en termes de statistiques, de contraintes, de paramètres d'environnement, de liaisons de paramètres de requêtes et de mises à jour du moteur de base de données.

R

Matrice RACI

Voir responsable, responsable, consulté, informé (RACI).

CHIFFON

Voir Retrieval Augmented Generation.

rançongiciel

Logiciel malveillant conçu pour bloquer l'accès à un système informatique ou à des données jusqu'à ce qu'un paiement soit effectué.

Matrice RASCI

Voir responsable, responsable, consulté, informé (RACI).

RCAC

Voir contrôle d'accès aux lignes et aux colonnes.

réplica en lecture

Copie d'une base de données utilisée en lecture seule. Vous pouvez acheminer les requêtes vers le réplica de lecture pour réduire la charge sur votre base de données principale.

réarchitecte

Voir 7 Rs.

objectif de point de récupération (RPO)

Durée maximale acceptable depuis le dernier point de récupération des données. Il détermine ce qui est considéré comme étant une perte de données acceptable entre le dernier point de reprise et l’interruption du service.

objectif de temps de récupération (RTO)

Le délai maximum acceptable entre l'interruption du service et le rétablissement du service.

refactoriser

Voir 7 Rs.

Région

Ensemble de AWS ressources dans une zone géographique. Chacun Région AWS est isolé et indépendant des autres pour garantir tolérance aux pannes, stabilité et résilience. Pour plus d'informations, voir Spécifier ce que Régions AWS votre compte peut utiliser.

régression

Technique de ML qui prédit une valeur numérique. Par exemple, pour résoudre le problème « Quel sera le prix de vente de cette maison ? », un modèle de ML pourrait utiliser un modèle de régression linéaire pour prédire le prix de vente d'une maison sur la base de faits connus à son sujet (par exemple, la superficie en mètres carrés).

réhéberger

Voir 7 Rs.

version

Dans un processus de déploiement, action visant à promouvoir les modifications apportées à un environnement de production.

déplacer

Voir 7 Rs.

replateforme

Voir 7 Rs.

rachat

Voir 7 Rs.

résilience

La capacité d'une application à résister aux perturbations ou à s'en remettre. La haute disponibilité et la reprise après sinistre sont des considérations courantes lors de la planification de la résilience dans le AWS Cloud. Pour plus d'informations, consultez AWS Cloud Résilience.

politique basée sur les ressources

Politique attachée à une ressource, comme un compartiment Amazon S3, un point de terminaison ou une clé de chiffrement. Ce type de politique précise les principaux auxquels l'accès est autorisé, les actions prises en charge et toutes les autres conditions qui doivent être remplies.

matrice responsable, redevable, consulté et informé (RACI)

Une matrice qui définit les rôles et les responsabilités de toutes les parties impliquées dans les activités de migration et les opérations cloud. Le nom de la matrice est dérivé des types de responsabilité définis dans la matrice : responsable (R), responsable (A), consulté (C) et informé (I). Le type de support (S) est facultatif. Si vous incluez le support, la matrice est appelée matrice RASCI, et si vous l'excluez, elle est appelée matrice RACI.

contrôle réactif

Contrôle de sécurité conçu pour permettre de remédier aux événements indésirables ou aux écarts par rapport à votre référence de sécurité. Pour plus d'informations, veuillez consulter la rubrique Responsive controls dans Implementing security controls on AWS.

retain

Voir 7 Rs.

se retirer

Voir 7 Rs.

Génération augmentée de récupération (RAG)

Technologie d'IA générative dans laquelle un LLM fait référence à une source de données faisant autorité qui se trouve en dehors de ses sources de données de formation avant de générer une réponse. Par exemple, un modèle RAG peut effectuer une recherche sémantique dans la base de connaissances ou dans les données personnalisées d'une organisation. Pour plus d'informations, voir Qu'est-ce que RAG ?

rotation

Processus de mise à jour périodique d'un secret pour empêcher un attaquant d'accéder aux informations d'identification.

contrôle d'accès aux lignes et aux colonnes (RCAC)

Utilisation d'expressions SQL simples et flexibles dotées de règles d'accès définies. Le RCAC comprend des autorisations de ligne et des masques de colonnes.

RPO

Voir l'objectif du point de récupération.

RTO

Voir l'objectif en matière de temps de rétablissement.

runbook

Ensemble de procédures manuelles ou automatisées nécessaires à l'exécution d'une tâche spécifique. Elles visent généralement à rationaliser les opérations ou les procédures répétitives présentant des taux d'erreur élevés.

S

SAML 2.0

Un standard ouvert utilisé par de nombreux fournisseurs d'identité (IdPs). Cette fonctionnalité permet l'authentification unique fédérée (SSO), afin que les utilisateurs puissent se connecter AWS Management Console ou appeler les opérations de l' AWS API sans que vous ayez à créer un utilisateur dans IAM pour tous les membres de votre organisation. Pour plus d'informations sur la fédération SAML 2.0, veuillez consulter À propos de la fédération SAML 2.0 dans la documentation IAM.

SCADA

Voir Contrôle de supervision et acquisition de données.

SCP

Voir la politique de contrôle des services.

secret

Dans AWS Secrets Manager des informations confidentielles ou restreintes, telles qu'un mot de passe ou des informations d'identification utilisateur, que vous stockez sous forme cryptée. Il comprend la valeur secrète et ses métadonnées. La valeur secrète peut être binaire, une chaîne unique ou plusieurs chaînes. Pour plus d'informations, voir Que contient le secret d'un Secrets Manager ? dans la documentation de Secrets Manager.

sécurité dès la conception

Une approche d'ingénierie système qui prend en compte la sécurité tout au long du processus de développement.

contrôle de sécurité

Barrière de protection technique ou administrative qui empêche, détecte ou réduit la capacité d'un assaillant d'exploiter une vulnérabilité de sécurité. Il existe quatre principaux types de contrôles de sécurité : préventifs, détectifs, réactifs et proactifs.

renforcement de la sécurité

Processus qui consiste à réduire la surface d'attaque pour la rendre plus résistante aux attaques. Cela peut inclure des actions telles que la suppression de ressources qui ne sont plus requises, la mise en œuvre des bonnes pratiques de sécurité consistant à accorder le moindre privilège ou la désactivation de fonctionnalités inutiles dans les fichiers de configuration.

système de gestion des informations et des événements de sécurité (SIEM)

Outils et services qui associent les systèmes de gestion des informations de sécurité (SIM) et de gestion des événements de sécurité (SEM). Un système SIEM collecte, surveille et analyse les données provenant de serveurs, de réseaux, d'appareils et d'autres sources afin de détecter les menaces et les failles de sécurité, mais aussi de générer des alertes.

automatisation des réponses de sécurité

Action prédéfinie et programmée conçue pour répondre automatiquement à un événement de sécurité ou y remédier. Ces automatisations servent de contrôles de sécurité détectifs ou réactifs qui vous aident à mettre en œuvre les meilleures pratiques AWS de sécurité. Parmi les actions de réponse automatique, citons la modification d'un groupe de sécurité VPC, l'application de correctifs à une EC2 instance Amazon ou la rotation des informations d'identification.

chiffrement côté serveur

Chiffrement des données à destination, par celui Service AWS qui les reçoit.

Politique de contrôle des services (SCP)

Politique qui fournit un contrôle centralisé des autorisations pour tous les comptes d'une organisation dans AWS Organizations. SCPs définissez des garde-fous ou des limites aux actions qu'un administrateur peut déléguer à des utilisateurs ou à des rôles. Vous pouvez les utiliser SCPs comme listes d'autorisation ou de refus pour spécifier les services ou les actions autorisés ou interdits. Pour plus d'informations, consultez la section Politiques de contrôle des services dans la AWS Organizations documentation.

point de terminaison du service

URL du point d'entrée pour un Service AWS. Pour vous connecter par programmation au service cible, vous pouvez utiliser un point de terminaison. Pour plus d'informations, veuillez consulter la rubrique Service AWS endpoints dans Références générales AWS.

contrat de niveau de service (SLA)

Accord qui précise ce qu'une équipe informatique promet de fournir à ses clients, comme le temps de disponibilité et les performances des services.

indicateur de niveau de service (SLI)

Mesure d'un aspect des performances d'un service, tel que son taux d'erreur, sa disponibilité ou son débit.

objectif de niveau de service (SLO)

Mesure cible qui représente l'état d'un service, tel que mesuré par un indicateur de niveau de service.

modèle de responsabilité partagée

Un modèle décrivant la responsabilité que vous partagez en matière AWS de sécurité et de conformité dans le cloud. AWS est responsable de la sécurité du cloud, alors que vous êtes responsable de la sécurité dans le cloud. Pour de plus amples informations, veuillez consulter Modèle de responsabilité partagée.

SIEM

Consultez les informations de sécurité et le système de gestion des événements.

point de défaillance unique (SPOF)

Défaillance d'un seul composant critique d'une application susceptible de perturber le système.

SLA

Voir le contrat de niveau de service.

SLI

Voir l'indicateur de niveau de service.

SLO

Voir l'objectif de niveau de service.

split-and-seed modèle

Modèle permettant de mettre à l'échelle et d'accélérer les projets de modernisation. Au fur et à mesure que les nouvelles fonctionnalités et les nouvelles versions de produits sont définies, l'équipe principale se divise pour créer des équipes de produit. Cela permet de mettre à l'échelle les capacités et les services de votre organisation, d'améliorer la productivité des développeurs et de favoriser une innovation rapide. Pour plus d'informations, consultez la section Approche progressive de la modernisation des applications dans le. AWS Cloud

SPOF

Voir point de défaillance unique.

schéma en étoile

Structure organisationnelle de base de données qui utilise une grande table de faits pour stocker les données transactionnelles ou mesurées et utilise une ou plusieurs tables dimensionnelles plus petites pour stocker les attributs des données. Cette structure est conçue pour être utilisée dans un entrepôt de données ou à des fins de business intelligence.

modèle de figuier étrangleur

Approche de modernisation des systèmes monolithiques en réécrivant et en remplaçant progressivement les fonctionnalités du système jusqu'à ce que le système hérité puisse être mis hors service. Ce modèle utilise l'analogie d'un figuier de vigne qui se développe dans un arbre existant et qui finit par supplanter son hôte. Le schéma a été présenté par Martin Fowler comme un moyen de gérer les risques lors de la réécriture de systèmes monolithiques. Pour obtenir un exemple d'application de ce modèle, veuillez consulter Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

sous-réseau

Plage d'adresses IP dans votre VPC. Un sous-réseau doit se trouver dans une seule zone de disponibilité.

contrôle de supervision et acquisition de données (SCADA)

Dans le secteur manufacturier, un système qui utilise du matériel et des logiciels pour surveiller les actifs physiques et les opérations de production.

chiffrement symétrique

Algorithme de chiffrement qui utilise la même clé pour chiffrer et déchiffrer les données.

tests synthétiques

Tester un système de manière à simuler les interactions des utilisateurs afin de détecter les problèmes potentiels ou de surveiller les performances. Vous pouvez utiliser Amazon CloudWatch Synthetics pour créer ces tests.

invite du système

Technique permettant de fournir un contexte, des instructions ou des directives à un LLM afin d'orienter son comportement. Les instructions du système aident à définir le contexte et à établir des règles pour les interactions avec les utilisateurs.

T

balises

Des paires clé-valeur qui agissent comme des métadonnées pour organiser vos AWS ressources. Les balises peuvent vous aider à gérer, identifier, organiser, rechercher et filtrer des ressources. Pour plus d'informations, veuillez consulter la rubrique Balisage de vos AWS ressources.

variable cible

La valeur que vous essayez de prédire dans le cadre du ML supervisé. Elle est également qualifiée de variable de résultat. Par exemple, dans un environnement de fabrication, la variable cible peut être un défaut du produit.

liste de tâches

Outil utilisé pour suivre les progrès dans un runbook. Liste de tâches qui contient une vue d'ensemble du runbook et une liste des tâches générales à effectuer. Pour chaque tâche générale, elle inclut le temps estimé nécessaire, le propriétaire et l'avancement.

environnement de test

Voir environnement.

entraînement

Pour fournir des données à partir desquelles votre modèle de ML peut apprendre. Les données d'entraînement doivent contenir la bonne réponse. L'algorithme d'apprentissage identifie des modèles dans les données d'entraînement, qui mettent en correspondance les attributs des données d'entrée avec la cible (la réponse que vous souhaitez prédire). Il fournit un modèle de ML

qui capture ces modèles. Vous pouvez alors utiliser le modèle de ML pour obtenir des prédictions sur de nouvelles données pour lesquelles vous ne connaissez pas la cible.

passerelle de transit

Un hub de transit réseau que vous pouvez utiliser pour interconnecter vos réseaux VPCs et ceux sur site. Pour plus d'informations, voir Qu'est-ce qu'une passerelle de transit dans la AWS Transit Gateway documentation.

flux de travail basé sur jonction

Approche selon laquelle les développeurs génèrent et testent des fonctionnalités localement dans une branche de fonctionnalités, puis fusionnent ces modifications dans la branche principale. La branche principale est ensuite intégrée aux environnements de développement, de préproduction et de production, de manière séquentielle.

accès sécurisé

Accorder des autorisations à un service que vous spécifiez pour effectuer des tâches au sein de votre organisation AWS Organizations et dans ses comptes en votre nom. Le service de confiance crée un rôle lié au service dans chaque compte, lorsque ce rôle est nécessaire, pour effectuer des tâches de gestion à votre place. Pour plus d'informations, consultez la section Utilisation AWS Organizations avec d'autres AWS services dans la AWS Organizations documentation.

réglage

Pour modifier certains aspects de votre processus d'entraînement afin d'améliorer la précision du modèle de ML. Par exemple, vous pouvez entraîner le modèle de ML en générant un ensemble d'étiquetage, en ajoutant des étiquettes, puis en répétant ces étapes plusieurs fois avec différents paramètres pour optimiser le modèle.

équipe de deux pizzas

Une petite DevOps équipe que vous pouvez nourrir avec deux pizzas. Une équipe de deux pizzas garantit les meilleures opportunités de collaboration possible dans le développement de logiciels.

U

incertitude

Un concept qui fait référence à des informations imprécises, incomplètes ou inconnues susceptibles de compromettre la fiabilité des modèles de ML prédictifs. Il existe deux types

d'incertitude : l'incertitude épistémique est causée par des données limitées et incomplètes, alors que l'incertitude aléatoire est causée par le bruit et le caractère aléatoire inhérents aux données. Pour plus d'informations, veuillez consulter le guide Quantifying uncertainty in deep learning systems.

tâches indifférenciées

Également connu sous le nom de « levage de charges lourdes », ce travail est nécessaire pour créer et exploiter une application, mais qui n'apporte pas de valeur directe à l'utilisateur final ni d'avantage concurrentiel. Les exemples de tâches indifférenciées incluent l'approvisionnement, la maintenance et la planification des capacités.

environnements supérieurs

Voir environnement.

V

mise à vide

Opération de maintenance de base de données qui implique un nettoyage après des mises à jour incrémentielles afin de récupérer de l'espace de stockage et d'améliorer les performances.

contrôle de version

Processus et outils permettant de suivre les modifications, telles que les modifications apportées au code source dans un référentiel.

Appairage de VPC

Une connexion entre deux VPCs qui vous permet d'acheminer le trafic en utilisant des adresses IP privées. Pour plus d'informations, veuillez consulter la rubrique Qu'est-ce que l'appairage de VPC ? dans la documentation Amazon VPC.

vulnérabilités

Défaut logiciel ou matériel qui compromet la sécurité du système.

W

cache actif

Cache tampon qui contient les données actuelles et pertinentes fréquemment consultées. L'instance de base de données peut lire à partir du cache tampon, ce qui est plus rapide que la lecture à partir de la mémoire principale ou du disque.

données chaudes

Données rarement consultées. Lorsque vous interrogez ce type de données, des requêtes modérément lentes sont généralement acceptables.

fonction de fenêtre

Fonction SQL qui effectue un calcul sur un groupe de lignes liées d'une manière ou d'une autre à l'enregistrement en cours. Les fonctions de fenêtre sont utiles pour traiter des tâches, telles que le calcul d'une moyenne mobile ou l'accès à la valeur des lignes en fonction de la position relative de la ligne en cours.

charge de travail

Ensemble de ressources et de code qui fournit une valeur métier, par exemple une application destinée au client ou un processus de backend.

flux de travail

Groupes fonctionnels d'un projet de migration chargés d'un ensemble de tâches spécifique. Chaque flux de travail est indépendant, mais prend en charge les autres flux de travail du projet. Par exemple, le flux de travail du portefeuille est chargé de prioriser les applications, de planifier les vagues et de collecter les métadonnées de migration. Le flux de travail du portefeuille fournit ces actifs au flux de travail de migration, qui migre ensuite les serveurs et les applications.

VER

Voir écrire une fois, lire plusieurs.

WQF

Voir le cadre AWS de qualification de la charge de travail.

écrire une fois, lire plusieurs (WORM)

Modèle de stockage qui écrit les données une seule fois et empêche leur suppression ou leur modification. Les utilisateurs autorisés peuvent lire les données autant de fois que nécessaire,

mais ils ne peuvent pas les modifier. Cette infrastructure de stockage de données est considérée comme immuable.

Z

exploit Zero-Day

Une attaque, généralement un logiciel malveillant, qui tire parti d'une vulnérabilité de type « jour zéro ».

vulnérabilité « jour zéro »

Une faille ou une vulnérabilité non atténuée dans un système de production. Les acteurs malveillants peuvent utiliser ce type de vulnérabilité pour attaquer le système. Les développeurs prennent souvent conscience de la vulnérabilité à la suite de l'attaque.

invite Zero-Shot

Fournir à un LLM des instructions pour effectuer une tâche, mais aucun exemple (plans) pouvant aider à la guider. Le LLM doit utiliser ses connaissances pré-entraînées pour gérer la tâche. L'efficacité de l'invite zéro dépend de la complexité de la tâche et de la qualité de l'invite. Voir également les instructions en quelques clics.

application zombie

Application dont l'utilisation moyenne du processeur et de la mémoire est inférieure à 5 %. Dans un projet de migration, il est courant de retirer ces applications.

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.