



Replate-forme de vos applications mainframe à l'aide d'une base de données  
IBM Db2 for z/OS partagée pour une migration progressive

## AWS Conseils prescriptifs



# AWS Conseils prescriptifs: Replate-forme de vos applications mainframe à l'aide d'une base de données IBM Db2 for z/OS partagée pour une migration progressive

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

Introduction .....	1
Résultats métier .....	2
AWS Mainframe Modernization .....	4
Replateforme .....	4
Refactorisation automatisée .....	5
Avantages du replatforming .....	5
Le processus de transformation .....	7
Planification .....	8
Découverte d'applications .....	8
Dépendances des données .....	8
Critère de capacité .....	9
Planification des vagues .....	12
Création .....	12
Cohérence des applications .....	13
Architecture .....	14
En cours d'exécution .....	16
Commission en deux phases (2PC) .....	17
Infrastructure d'exécution .....	18
Test .....	19
Environnement source .....	19
Environnement cible .....	20
Analyse .....	20
Tester votre application dans AWS Mainframe Modernization .....	21
Basculement .....	23
Disposition .....	23
Passez en direct .....	24
Restauration .....	24
Conclure .....	24
Architecture .....	24
Bonnes pratiques .....	26
Latence du réseau .....	26
Sécurité .....	27
Gouvernance des applications .....	27
Élasticité .....	28

---

Étapes suivantes .....	29
Ressources .....	31
AWS documentation .....	31
Références de Rocket Software .....	31
Références IBM .....	31
Outils .....	31
AWS Modèles et guides d'orientation prescriptifs .....	31
Historique du document .....	32
Glossaire .....	33
# .....	33
A .....	34
B .....	37
C .....	39
D .....	43
E .....	47
F .....	49
G .....	51
H .....	52
I .....	54
L .....	57
M .....	58
O .....	62
P .....	65
Q .....	68
R .....	68
S .....	72
T .....	76
U .....	77
V .....	78
W .....	78
Z .....	79
.....	lxxxi

# Replateforme de vos applications mainframe à l'aide d'une base de données IBM Db2 for partagée pour z/OS une migration progressive

Luis Gustavo Dantas et André Botura, Amazon Web Services (AWS)

Mai 2025 ([historique du document](#))

Dans le paysage technologique en constante évolution des entreprises, la modernisation des ordinateurs centraux est devenue une exigence essentielle pour les entreprises qui doivent rester compétitives et agiles. Cette transformation ne consiste pas simplement à remplacer les anciens systèmes par de nouveaux ; il s'agit d'une évolution stratégique qui comble le fossé entre les bases solides et fiables du passé et les possibilités dynamiques et innovantes du futur.

Le mainframe, qui était autrefois le leader incontesté de l'informatique d'entreprise, est aujourd'hui à un tournant décisif. Sa puissance de traitement et ses fonctionnalités de sécurité hors pair lui ont permis de rester pertinent pendant des décennies, mais les entreprises d'aujourd'hui ont besoin de systèmes capables de s'intégrer parfaitement aux services cloud, de prendre en charge les applications mobiles et de tirer parti de la puissance de l'intelligence artificielle et de l'analyse des mégadonnées.

La modernisation ne nécessite pas toujours une migration complète hors des ordinateurs centraux. Certaines entreprises optent pour des approches hybrides qui tirent parti des points forts des environnements mainframe et cloud. Cette stratégie leur permet de conserver leurs anciennes applications critiques tout en passant progressivement à des plateformes plus modernes. Cette transition technologique ne se limite pas à des mises à jour du système ; elle nécessite une transformation de la culture et des compétences organisationnelles. À mesure que les entreprises se modernisent, elles investissent à la fois dans les nouvelles technologies et dans leur main-d'œuvre en comblant les écarts entre les générations et en promouvant l'apprentissage et l'innovation continus.

Ce guide décrit une stratégie de migration progressive qui équilibre les avantages des systèmes mainframe avec ceux des technologies cloud modernes. Cette approche de replatforme progressive fait d'abord migrer la couche applicative, tout en maintenant la connectivité à votre z/OS base de données IBM Db2 for existante afin de rationaliser le processus de transition et de minimiser les

perturbations de vos opérations commerciales critiques pendant que vous adoptez de nouvelles fonctionnalités dans le cloud.

Ce guide est destiné aux décideurs techniques et aux équipes de mise en œuvre impliquées dans des initiatives de modernisation du mainframe. Les principaux publics comprennent les architectes d'entreprise et de solutions, les chefs de projets techniques et les responsables de programmes de modernisation qui doivent comprendre à la fois les aspects stratégiques et techniques du replatformage des ordinateurs centraux. Le contenu est tout aussi utile pour les équipes de mise en œuvre, notamment les développeurs d'applications mainframe AWS ou les ingénieurs cloud, les administrateurs de bases de données et les DevOps ingénieurs chargés de mettre en œuvre la modernisation.

## Résultats métier

Les entreprises ont de nombreuses raisons impérieuses de mettre à jour leurs anciennes applications. Ce processus crée un sentiment d'urgence dans tous les secteurs. Lorsque des experts plus âgés prennent leur retraite, ils laissent un important déficit de connaissances, d'où la nécessité de moderniser les systèmes avant que cette expertise ne soit perdue. En outre, les entreprises sont motivées par la nécessité de réduire les coûts, d'accroître leur agilité et de réagir rapidement à l'évolution rapide des conditions du marché.

La transformation numérique est encore intensifiée par les technologies émergentes et la demande d'expériences client améliorées. Ces facteurs, combinés aux risques associés à la maintenance de systèmes complexes, incitent les entreprises à agir rapidement pour moderniser leur infrastructure informatique.

La modernisation des ordinateurs centraux, en particulier, représente un exercice d'équilibre délicat. Les entreprises doivent préserver la stabilité et la sécurité qui font la réputation des mainframes, tout en profitant de la flexibilité et de l'évolutivité offertes par les architectures modernes. Ce processus implique des décisions complexes concernant les applications à migrer, celles à réécrire et celles à conserver sur le mainframe.

Les principaux moteurs de la modernisation incluent l'agilité et la réduction des coûts :

- Agilité et délai de mise sur le marché. Les systèmes modernes permettent d'accélérer les processus d'approvisionnement et de répondre plus rapidement aux demandes changeantes du marché. L'adoption DevOps et les SysOps pratiques peuvent améliorer considérablement la productivité et les vitesses de déploiement.

- Réduction des coûts. La modernisation entraîne souvent une réduction des coûts d'infrastructure grâce aux mesures suivantes :
  - Pay-as-you-go modèles, qui alignent les coûts sur l'utilisation réelle.
  - Réduction des frais de licence associés aux anciens systèmes.
  - Elasticité améliorée, ce qui permet une meilleure allocation des ressources.
  - Des configurations actives et à haute disponibilité, qui améliorent la résilience du système tout en optimisant l'utilisation des ressources.

Sur la base de ces facteurs commerciaux, la replateforme des applications COBOL est considérée comme une approche stratégique de la modernisation. Vous pouvez utiliser une base de données partagée pour suivre un parcours de migration progressif qui concilie le besoin de modernisation avec l'impératif de continuité des activités. Cette méthode vous permet de tirer parti des avantages des architectures modernes tout en préservant la fiabilité de vos applications COBOL. Vous pouvez ainsi gagner en agilité, en rentabilité et en innovation tout en atténuant les risques associés aux transitions abruptes et à grande échelle. L'approche de base de données DB2 partagée décrite dans ce guide fournit un pont entre les systèmes existants et les plateformes modernes, et permet un processus de modernisation plus fluide et mieux contrôlé.

Dans ce guide :

- [AWS Mainframe Modernization](#)
- [Le processus de transformation](#)
- [Bonnes pratiques](#)
- [Étapes suivantes](#)
- [Ressources](#)
- [Historique du document](#)

# AWS Mainframe Modernization

## Note

AWS Mainframe Modernization Le service (Managed Runtime Environment Experience) n'est plus ouvert aux nouveaux clients. Pour des fonctionnalités similaires à AWS Mainframe Modernization Service (expérience d'environnement d'exécution géré), explorez AWS Mainframe Modernization Service (expérience autogérée). Les clients existants peuvent continuer à utiliser le service normalement. Pour plus d'informations, consultez [AWS Mainframe Modernization la section Modification de la disponibilité](#).

Le [AWS Mainframe Modernization service](#) vous permet de migrer vos anciennes applications mainframe vers un environnement cloud natif, de préserver la logique métier et les investissements existants, d'utiliser des outils automatisés et des services d'exécution gérés, d'optimiser les performances des applications et de réduire les coûts opérationnels. Ce service rationalise le processus de modernisation afin que vous puissiez exploiter la puissance du cloud tout en préservant la valeur de vos principaux systèmes mainframe. AWS propose deux approches clés pour la modernisation des ordinateurs centraux : le replatforming et le refactoring automatisé.

## Replateforme

[AWS Mainframe Modernization La replatforme avec Rocket Software](#) (anciennement Micro Focus) constitue une puissante option de replatforme pour les entreprises qui souhaitent migrer leurs applications mainframe vers le cloud avec un minimum de perturbations. Cette solution vous permet de recompiler et d'exécuter votre COBOL et vos PL/I applications existants AWS sans nécessiter de modifications de code importantes.

Les principaux avantages de la solution AWS Replatform with Rocket Software sont les suivants :

- Préservation de la logique commerciale et des investissements existants
- Réduction des risques et réduction des délais de mise sur le marché
- Amélioration de l'évolutivité et des performances de l'infrastructure AWS
- Accès à des outils et pratiques de développement modernes

Vous pouvez utiliser cette solution pour conserver vos langages de programmation mainframe habituels tout en tirant parti de la flexibilité, de la rentabilité et de l'innovation du. AWS Cloud

## Refactorisation automatisée

Pour une approche plus transformatrice que le replatforming, vous pouvez utiliser [AWS Blu Age](#), qui permet de refactoriser automatiquement les applications mainframe en applications cloud natives basées sur Java. Cette solution vous aide à moderniser vos systèmes existants de manière plus complète et à les convertir en applications capables de tirer pleinement parti des technologies natives du cloud.

Les principaux avantages de AWS Blu Age sont les suivants :

- Conversion du code existant en applications Java modernes et maintenables
- Transformation automatisée qui réduit les efforts manuels et les erreurs potentielles
- Création d'applications cloud natives optimisées pour Services AWS
- Agilité améliorée et intégration facilitée avec les technologies modernes

AWS Blu Age vous aide à migrer vos applications et à les préparer pour le cloud, afin d'ouvrir de nouvelles possibilités d'innovation et de croissance. Pour plus d'informations sur cette approche, consultez la section [Refactorisation automatique des applications avec AWS Blu Age](#) dans la AWS Mainframe Modernization documentation.

## Avantages du replatforming

Ce guide décrit une approche pour la replatforme des applications COBOL du mainframe sur. AWS Cette approche vise à moderniser les systèmes existants tout en conservant temporairement IBM Db2 afin z/OS de rationaliser le processus de transition. En conservant initialement la structure de base de données existante, vous pouvez réduire la complexité et les risques lors de la migration. Cette approche progressive vous permet de bénéficier de l'évolutivité et de la rentabilité du AWS Cloud tout en préservant l'intégrité des données critiques. Les avantages de la replatforme progressive sont les suivants :

- Modernisation accélérée : le replatforming et le refactoring nécessitent généralement moins de temps et de ressources que la réinvention d'une application existante dans le cloud, car ils n'impliquent pas de réécrire l'application dans son intégralité. Cette approche favorise également

une transition plus progressive qui permet aux entreprises de se moderniser à leur propre rythme tout en bénéficiant immédiatement de l'évolutivité et de la rentabilité du. AWS Cloud

- **Atténuation des risques** : le replatforming présente plusieurs avantages par rapport au refactoring pour de nombreuses organisations. Les entreprises peuvent maintenir leur COBOL et leurs PL/I bases de code existantes, préserver des années de logique métier et minimiser les risques associés aux modifications de code à grande échelle.
- **Continuité des données et migration progressive** : l'un des principaux avantages de la replateforme est la possibilité de conserver initialement les données dans DB2 dans leur format de données d' z/OS origine. Cette stratégie évite d'avoir à recourir à des processus de migration de données immédiats, complexes et potentiellement risqués. En conservant les données dans leur environnement d'origine pendant la phase initiale, vous pouvez préserver l'intégrité des données, réduire les temps d'arrêt et minimiser le risque de perte ou de corruption des données pendant le processus de modernisation. Dans un deuxième temps, vous pouvez planifier une migration progressive et contrôlée des données vers des bases de données cloud natives qui implique des tests et une validation approfondis pendant que l'application continue de s'exécuter sur l'environnement replateforme.
- **Flexibilité et pérennité** : pour les entreprises qui investissent de manière significative dans les compétences et les applications du mainframe, le replatforming constitue une voie pragmatique de modernisation qui concilie innovation et continuité. Il offre la flexibilité nécessaire pour conserver les structures de données et les méthodes d'accès critiques dans un premier temps, tout en préparant le terrain pour les futurs efforts de modernisation, y compris la migration éventuelle des données vers des solutions entièrement natives du cloud.

Organisations peuvent suivre l'approche de la replateforme pour se moderniser à leur propre rythme et répondre à leurs besoins immédiats tout en planifiant des objectifs de transformation numérique à long terme. Cette approche donne également aux entreprises la possibilité de former leur personnel aux services cloud natifs.

# Le processus de transformation

La modernisation du mainframe est une étape essentielle pour les entreprises qui souhaitent tirer parti des avantages du cloud computing tout en préservant leurs précieuses applications existantes. Cette transformation présente des défis importants. Les applications mainframe sont généralement hautement couplées et présentent des interdépendances complexes qui ont évolué au fil des décennies d'exploitation. Cette complexité nécessite une approche prudente et méthodique de la modernisation.

Organisations doivent passer par les phases clés suivantes pour réussir leur transition :

- **Planification** : Cette phase implique une découverte complète des systèmes existants et une hiérarchisation des efforts de modernisation. Organisations évaluent leur infrastructure actuelle, identifient les applications critiques et déterminent quels systèmes doivent être modernisés en premier.
- **Construction** : Au cours de cette étape, les organisations créent des processus pour migrer les applications et développer de nouveaux systèmes et infrastructures. Cela implique la conception et la mise en œuvre de l'architecture modernisée et la compilation du code source.
- **Exécution** : cette étape consiste à créer les environnements d'exécution pour héberger les applications replatformées. Cela implique la mise en place du matériel, des logiciels et de l'infrastructure cloud nécessaires pour prendre en charge les systèmes modernisés et s'assurer qu'ils peuvent fonctionner efficacement dans le nouvel environnement.
- **Tests** : Cette phase comprend une validation rigoureuse des systèmes modernisés afin de vérifier que toutes les exigences fonctionnelles et de performance ont été satisfaites. Des tests approfondis sont effectués pour vérifier l'intégrité des données, la compatibilité du système et les performances globales du nouvel environnement.
- **Passage** : La phase finale se concentre sur la mise en œuvre de stratégies pour une transition en douceur et le contrôle du passage de l'ancien ordinateur central à l'environnement modernisé. Cela inclut une planification minutieuse du calendrier de migration et des plans d'urgence afin de minimiser les perturbations des opérations commerciales.

Les sections suivantes décrivent ces phases en détail :

- [Planification](#)
- [Bâtiment](#)

- [En cours d'exécution](#)
- [Test](#)
- [Découpe](#)

## Planification

Pour répondre efficacement aux exigences de l'ancienne application mainframe, les entreprises commencent souvent par une évaluation complète de leur environnement mainframe.

### Découverte d'applications

Dans cette phase initiale, le [Rocket Enterprise Analyzer](#) est un outil puissant qui fournit des informations approfondies sur la structure, les dépendances et la complexité des applications mainframe. Cet outil vous aide à déterminer l'ampleur de vos efforts de modernisation, les risques potentiels et les opportunités d'optimisation.

L'un des aspects cruciaux à découvrir est le réseau complexe de dépendances des données au sein des systèmes mainframe. Ces dépendances sont souvent cachées sous des couches de code existant et peuvent avoir un impact significatif sur les efforts de modernisation. En cartographiant la manière dont les différentes applications et modules interagissent avec les différentes sources de données, vous pouvez mieux comprendre les effets potentiels des modifications que vous envisagez de mettre en œuvre.

### Dépendances des données

Une évaluation approfondie des dépendances des données peut révéler des informations critiques sur le flux de données, la qualité des données et la gouvernance des données au sein de votre environnement mainframe. Ces connaissances sont précieuses pour planifier des stratégies de migration des données, garantir l'intégrité des données lors de la modernisation et identifier les opportunités d'optimisation des données. En obtenant une image claire de vos données, vous pouvez prendre des décisions plus éclairées quant aux approches de modernisation les plus efficaces et les moins perturbatrices pour vos opérations existantes.

Une analyse descendante identifiant l'utilisation des tables par les transactions ou les tâches en langage de contrôle des tâches (JCL) est essentielle pour créer une planification et une priorisation des vagues. Cette approche clarifie les relations entre les différents composants de vos systèmes

mainframe et vous aide à développer une approche stratégique et progressive de la modernisation. En identifiant les tables les plus fréquemment consultées et les processus par lesquels vous utilisez, vous pouvez hiérarchiser vos efforts de modernisation : vous pouvez d'abord vous concentrer sur les domaines à fort impact et garantir une transition plus fluide en perturbant le moins possible les opérations commerciales critiques.

En plus d'utiliser Rocket Enterprise Analyzer pour découvrir les dépendances des données, de nombreuses entreprises utilisent également leurs propres solutions personnalisées pour mieux comprendre leurs environnements mainframe. Ces outils internes exploitent souvent la richesse des informations disponibles dans le catalogue IBM Db2 et dans les enregistrements du System Management Facility (SMF).

## Critère de capacité

L'une des étapes de la planification de votre projet de replateforme de mainframe consiste à recueillir des informations détaillées sur votre consommation de charge de travail actuelle. Ces données vous aideront à prévoir et à fournir avec précision la capacité initiale requise dans votre environnement cloud cible. Par exemple, nous vous recommandons de collecter des données de consommation d'un million d'instructions par seconde (MIPS) par heure pour les transactions en ligne et les transactions par lots à partir des tâches IBM Customer Information Control System (CICS) ou du système de gestion de l'information (IMS) et du langage de contrôle des tâches (JCL) d'IBM.

IBM propose une gamme variée de [modèles de tarification](#) pour le MIPS dans l'informatique centrale, et bon nombre de ces modèles sont centrés sur les pics d'utilisation. Parmi ces modèles basés sur les heures de pointe, le plus courant est le pic continu de quatre heures.

Les coûts des ordinateurs centraux incluent cinq domaines clés qui ont un impact significatif sur les dépenses globales :

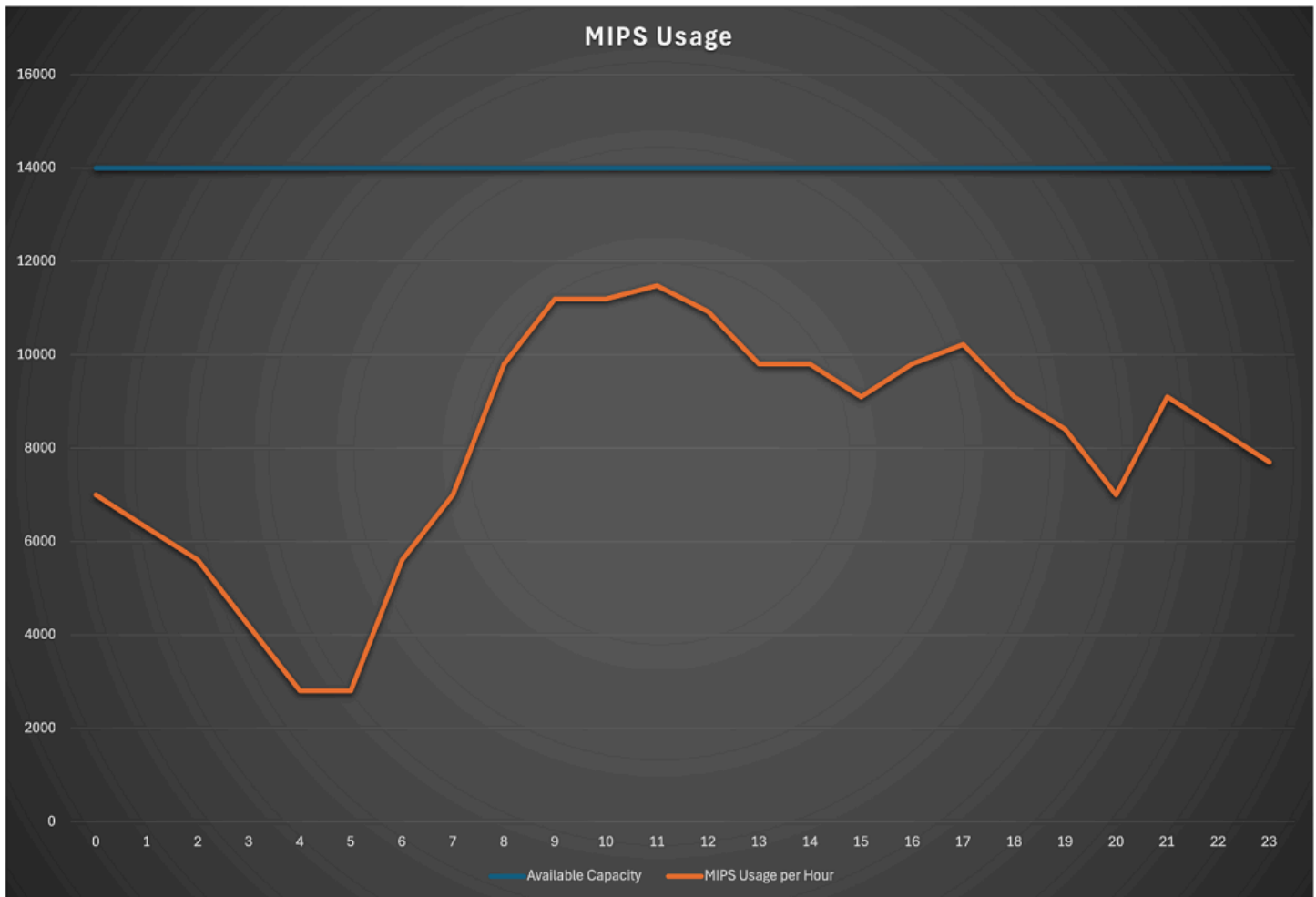
- Les licences logicielles constituent souvent un élément majeur. Il couvre les systèmes d'exploitation, les intergiciels, les bases de données et diverses applications, et les coûts sont parfois liés à la capacité ou à l'utilisation des machines.
- Les dépenses matérielles incluent l'achat initial ou la location d'équipement central, la maintenance continue et les mises à niveau.
- Les coûts de stockage peuvent être importants en raison des grandes quantités de données gérées et impliquent des systèmes de disques, des bibliothèques de bandes et des logiciels de gestion associés.

- Les dépenses de personnel couvrent les salaires des professionnels spécialisés dans les ordinateurs centraux tels que les programmeurs système et les administrateurs de bases de données.
- Les mesures de reprise après sinistre et de continuité des activités, notamment les systèmes de sauvegarde, le matériel redondant et les installations de reprise hors site, représentent un investissement important pour garantir une haute disponibilité et une reprise rapide.

Ces cinq catégories de coûts, associées aux frais basés sur le MIPS, constituent le cœur de la plupart des budgets des ordinateurs centraux. Toutefois, leurs proportions relatives peuvent varier considérablement en fonction de la taille de votre entreprise, de son secteur d'activité et de ses modèles d'utilisation spécifiques du mainframe.

Les données MIPS horaires sont cruciales pour obtenir une compréhension complète des modèles de charge de travail et des performances de votre mainframe. Contrairement aux moyennes quotidiennes ou mensuelles, les données horaires fournissent des informations détaillées qui révèlent les fluctuations nuancées de l'utilisation des ressources de votre système au cours de la journée. Ce niveau de détail est inestimable pour évaluer avec précision les besoins de performance et de capacité de votre application dans le cloud.

En analysant les données MIPS horaires, vous pouvez identifier les périodes de pointe d'utilisation, identifier les tendances et identifier les goulets d'étranglement potentiels susceptibles d'être masqués dans les données agrégées, comme le montre le schéma suivant. Cette granularité permet une planification plus précise des capacités, aide à optimiser l'allocation des ressources et peut potentiellement entraîner des économies de coûts et une amélioration de l'efficacité du système.



Les données MIPS horaires constituent également un outil de référence de performance essentiel. Il établit une base de référence détaillée des performances de votre système, ce qui est particulièrement utile lorsque vous planifiez ou évaluez des modifications du système, telles que des migrations ou des mises à niveau. En comparant les données MIPS horaires avant et après les modifications, vous pouvez mesurer avec précision l'impact de ces modifications sur les performances de votre système et vous assurer que votre mainframe continue de répondre aux besoins de votre entreprise.

Pour collecter des données MIPS horaires, plusieurs options s'offrent à vous. L'une des approches consiste à utiliser directement les enregistrements SMF. Ces enregistrements fournissent une mine d'informations sur l'activité du système et l'utilisation des ressources. Vous pouvez également utiliser des outils spécialisés tels que l'outil IBM Sub-Capacity Reporting Tool (SCRT), qui peut simplifier le processus de collecte et d'analyse des données MIPS.

Quelle que soit la méthode choisie, il est important de collecter les données sur une longue période, idéalement plusieurs mois. Cette période de collecte prolongée vous permet de tenir compte des variations cycliques de votre charge de travail, telles que les pics de end-of-month traitement ou les fluctuations saisonnières. En capturant ces modèles à long terme, vous pouvez obtenir une image plus précise et plus complète des caractéristiques de performance de votre mainframe, ce qui permet une prise de décision plus éclairée et une gestion des capacités plus efficace.

## Planification des vagues

Vous pouvez utiliser les informations que vous collectez pour hiérarchiser de manière stratégique vos initiatives de replatforme du mainframe. Une approche prudente consiste à commencer par des charges de travail moins critiques, telles que les transactions commerciales non essentielles ou les tâches par lots, afin de permettre aux équipes d'acquérir de l'expérience et d'affiner les processus avec un minimum de risques pour les opérations essentielles. En outre, il peut être avantageux de considérer les charges de travail en lecture seule comme les premières candidates à la migration, car ces charges de travail sont généralement moins complexes et présentent moins de risques d'incohérences dans les données. Cette approche vous permet de renforcer la confiance et l'élan dans vos efforts de replatforme.

En outre, le regroupement des charges de travail qui partagent des tables DB2 pour les opérations d'écriture ou de mise à jour peut rationaliser le processus de migration. En identifiant ces charges de travail interconnectées, vous pouvez planifier des vagues de migration cohérentes qui préservent l'intégrité des données et minimisent le besoin de solutions intermédiaires complexes. Cette stratégie réduit non seulement le risque de conflits de données, mais optimise également le calendrier global de replatforme en traitant simultanément les composants connexes. En fin de compte, cette approche de priorisation basée sur les données garantit une prise en compte équilibrée de la criticité, de la complexité et de l'interdépendance, et conduit à un processus de modernisation du mainframe plus efficace et plus réussi.

## Création

L'utilisation d'une base de données DB2 partagée permet l'exécution simultanée d'applications identiques ou cohérentes dans les environnements mainframe et cloud. Cette approche présente plusieurs avantages lorsque vous maintenez la même version d'application sur les deux plateformes, et améliore la flexibilité et la fiabilité de vos opérations.

L'un des principaux avantages de cette stratégie est la capacité à mettre en œuvre un plan de réduction efficace. Si des problèmes surviennent pendant la migration ou le déploiement, le fait de disposer de la même version d'application permet de revenir facilement à l'état précédent et de minimiser les temps d'arrêt et les incohérences potentielles entre les données.

## Cohérence des applications

La mise en miroir des composants de l'application depuis un gestionnaire de contrôle de source distribué vers le mainframe constitue une approche stratégique lors du processus de replateforme. Cette méthode prend en charge l'utilisation d'outils modernes de gestion du code source tout en maintenant la synchronisation avec l'environnement du mainframe. Ce processus de mise en miroir est temporaire et ne dure que jusqu'à ce que la charge de travail soit entièrement fonctionnelle en production sur la plate-forme distribuée.

En migrant le code source de votre application replateforme vers un outil de gestion des modifications distribué, vous pouvez tirer parti de plusieurs avantages offerts par les gestionnaires de code source modernes. Il s'agit des licences suivantes :

- **Collaboration améliorée** : les outils distribués fournissent souvent un meilleur support à la collaboration en équipe en incluant des fonctionnalités telles que les pull requests, les révisions de code et les stratégies de branchement.
- **Contrôle de version amélioré** : les systèmes modernes offrent un contrôle de version plus précis et facilitent le suivi des modifications et la gestion des différentes versions du code.
- **Intégration aux CI/CD pipelines** : de nombreux outils distribués s'intègrent parfaitement aux pipelines d'intégration continue et de déploiement continu (CI/CD), ce qui rationalise le processus de développement.
- **Visibilité et traçabilité améliorées** : ces outils fournissent souvent des tableaux de bord et des fonctionnalités de reporting de qualité supérieure, et offrent un meilleur aperçu du processus de développement.
- **Support aux pratiques de développement modernes** : les systèmes distribués sont généralement mieux adaptés aux méthodologies et DevOps pratiques agiles.

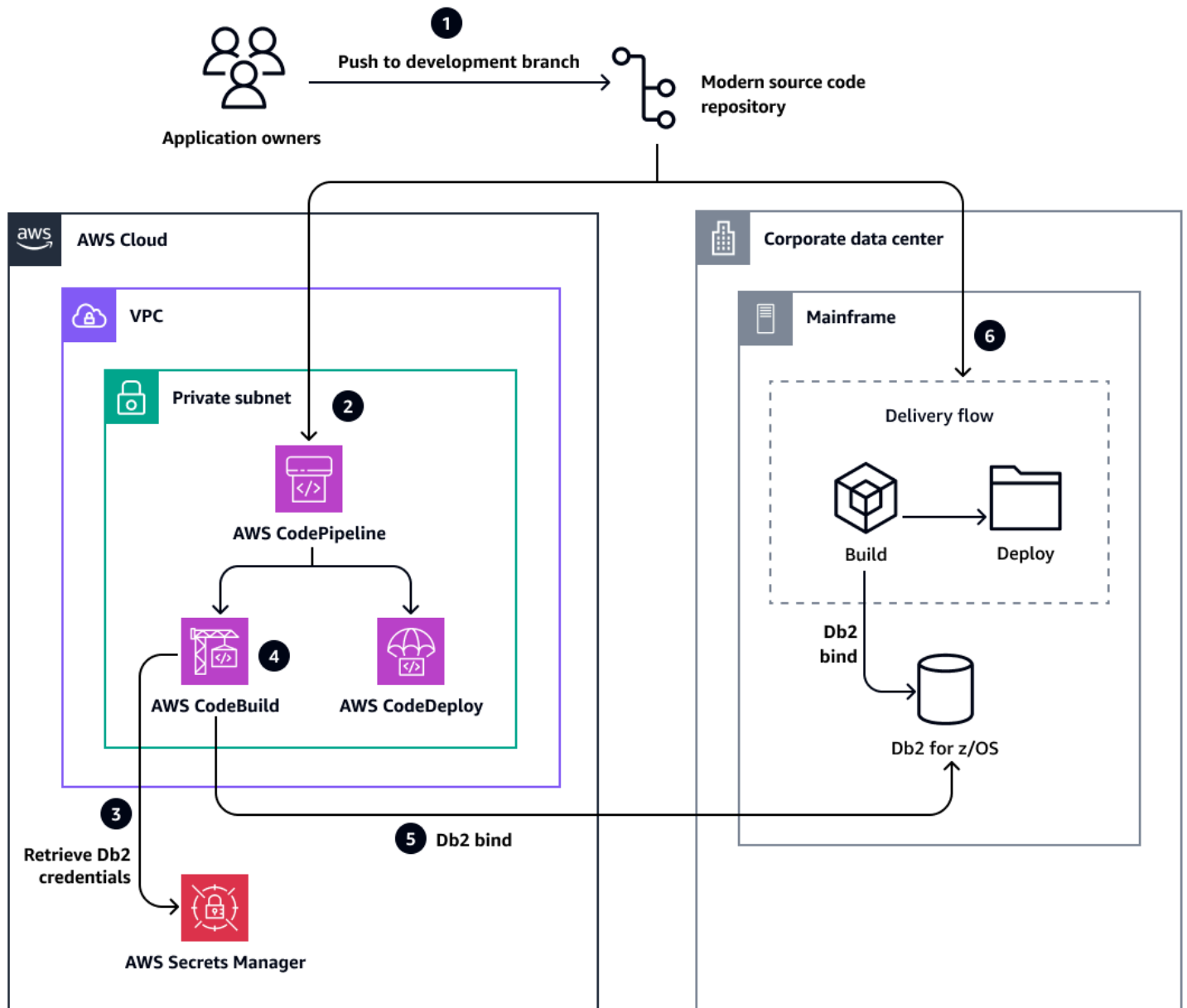
Le processus de mise en miroir implique la synchronisation du code entre le gestionnaire de contrôle de source distribué et le mainframe. Cela garantit que les deux environnements restent cohérents pendant la période de transition. Cependant, vous devez implémenter la mise en miroir sous forme de synchronisation unidirectionnelle, dans le cadre de laquelle les mises à jour circulent du système

distribué vers le mainframe, plutôt que de manière bidirectionnelle. Cette approche assure la cohérence et prévient les conflits potentiels qui pourraient découler de mises à jour simultanées dans les deux environnements.

En adoptant cette stratégie de mise en miroir, vous pouvez progressivement transférer vos efforts de développement vers la plate-forme distribuée tout en garantissant le maintien de l'environnement mainframe. up-to-date Cela permet une transition plus fluide et un filet de sécurité pendant le processus de replateforme. Lorsque la charge de travail est entièrement fonctionnelle et stable dans l'environnement de production distribué, vous pouvez supprimer progressivement le processus de mise en miroir et terminer la migration vers le système de gestion de code source moderne.

## Architecture

Le schéma suivant montre comment un système de gestion de code source distribué peut refléter les composants de l'application et maintenir la synchronisation entre les environnements mainframe AWS Cloud et mainframe. L' AWS Cloud environnement utilise CI/CD des services tels que [AWS CodeBuild](#)[AWS CodePipeline](#), et [AWS CodeDeploy](#) pour créer et déployer l'application.



Dans ce flux de travail :

1. Les propriétaires de l'application fournissent une nouvelle version de l'application dans la branche de développement du référentiel de code source.
2. La nouvelle version se déclenche AWS CodePipeline.
3. AWS CodeBuild récupère les informations d'identification DB2 à partir de [AWS Secrets Manager](#)
4. CodeBuild compile l'application.
5. CodeBuild utilise Db2 pour z/OS lier l'application.

6. Le flux de livraison du mainframe crée et déploie également l'application.

## En cours d'exécution

Pour garantir des performances optimales et une faible latence entre votre application basée sur le cloud et votre base de données sur site, nous vous recommandons d'implémenter [AWS Direct Connect](#). Ce service fournit une connexion réseau dédiée entre AWS le centre de données de votre entreprise et offre des performances réseau plus cohérentes et une latence réduite par rapport aux connexions Internet. Cela est particulièrement crucial pour les opérations de base de données qui nécessitent des temps de réponse rapides.

Pour garantir la haute disponibilité (HA) et l'élasticité de l'application qui s'exécute AWS, vous pouvez implémenter une architecture robuste à l'aide des composants suivants :

- Elastic Load Balancing (ELB) : vous pouvez déployer un équilibreur de charge pour répartir le trafic entrant sur plusieurs instances Amazon Elastic Compute Cloud (Amazon EC2) sur lesquelles votre application s'exécute. Cela garantit une répartition uniforme de la charge de travail et fournit un point d'entrée unique pour les demandes des clients.
- Groupe Auto Scaling : les instances EC2 qui hébergent l'application peuvent être organisées dans un groupe Auto Scaling. Cela permet à l'infrastructure d'ajuster automatiquement le nombre d'instances en fonction de mesures prédéfinies telles que l'utilisation du processeur ou le trafic réseau. Pendant les périodes de pointe, des instances supplémentaires peuvent être lancées pour gérer une charge accrue, tandis que pendant les périodes plus calmes, les instances inutiles peuvent être résiliées afin d'optimiser les coûts.
- Instances EC2 : l'application peut être déployée sur des instances EC2 au sein du groupe Auto Scaling. Ces instances doivent être réparties sur plusieurs zones de disponibilité afin d'améliorer la tolérance aux pannes et de garantir une haute disponibilité.
- Déploiement multi-AZ : en répartissant les instances d'application sur plusieurs zones de disponibilité, le système peut résister à la défaillance d'une seule zone de disponibilité sans impact significatif sur la disponibilité globale.

Cette architecture permet à l'application d'évoluer de manière fluide en fonction de la demande tout en maintenant une haute disponibilité. L'équilibreur de charge garantit que le trafic est réparti uniformément entre les instances saines, et le groupe Auto Scaling gère le nombre d'instances en fonction de la charge de travail réelle.

Pour améliorer encore la fiabilité, vous pouvez mettre en œuvre un système de surveillance et d'alerte robuste en utilisant [Amazon CloudWatch](#) pour détecter les problèmes de performance ou les défaillances et y répondre rapidement. En outre, des tests réguliers des capacités de mise à l'échelle automatique et des scénarios de basculement garantiront que le système se comporte comme prévu dans différentes conditions de charge et lors de défaillances potentielles.

En adoptant cette approche, vous pouvez bénéficier de l'évolutivité et de la flexibilité du AWS Cloud tout en maintenant une connexion sécurisée à votre base de données Db2 locale. Cette configuration hybride constitue une excellente voie vers une migration complète vers le cloud et permet une transition progressive et une atténuation des risques tout au long du processus.

## Commission en deux phases (2PC)

[AWS Mainframe Modernization Replatform with Rocket Software](#) permet de prendre en charge les transactions de validation en deux phases (2PC) grâce à la mise en œuvre d'une architecture étendue (XA). Cette fonctionnalité est essentielle au maintien de l'intégrité des données dans les systèmes distribués, en particulier dans les environnements mainframe où les transactions complexes concernent souvent plusieurs ressources.

L'architecture XA, intégrée à AWS Replatform with Rocket Software, permet de coordonner les transactions entre diverses ressources telles que les bases de données et les files d'attente de messages. Cette intégration garantit que toutes les parties d'une transaction distribuée sont validées ou annulées à l'unisson, afin de maintenir la cohérence dans l'ensemble du système.

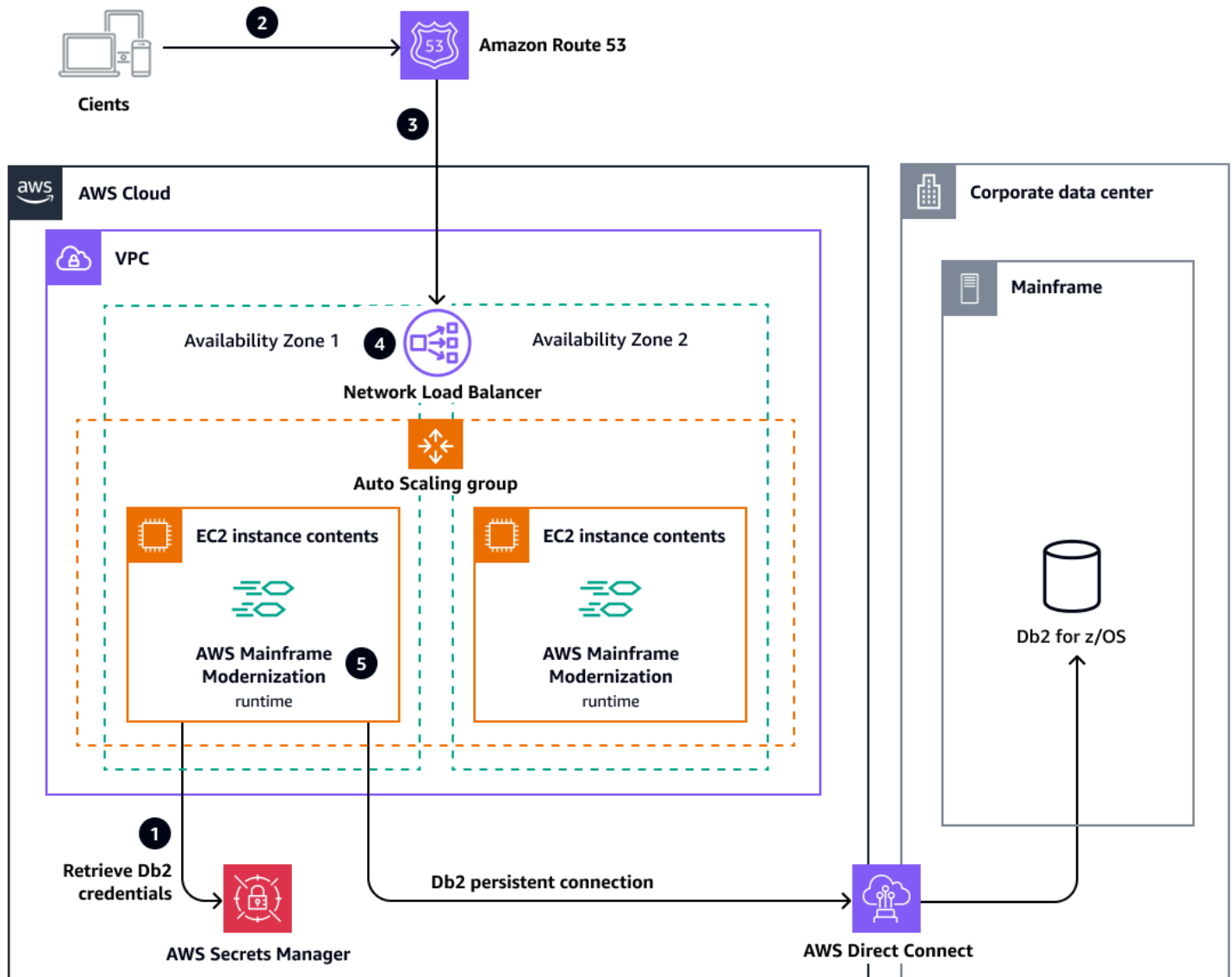
Le processus 2PC comprend deux phases :

- Phase de préparation : le responsable des transactions interroge tous les responsables de ressources impliqués dans la transaction pour s'assurer qu'ils sont prêts à s'engager.
- Phase de validation : si tous les responsables de ressources répondent positivement, le responsable des transactions leur demande de valider les modifications. Si l'un des responsables des ressources ne peut pas s'engager, tous les responsables sont invités à annuler les modifications.

En utilisant XA, AWS Replatform with Rocket Software fournit une solution fiable et évolutive pour gérer des transactions complexes et distribuées dans des environnements mainframe modernisés. Cette fonctionnalité est essentielle pour les entreprises qui souhaitent migrer leurs applications mainframe vers le cloud sans compromettre l'intégrité ou les performances transactionnelles.

## Infrastructure d'exécution

Le schéma suivant montre un environnement hautement disponible et élastique comprenant deux zones de AWS Cloud disponibilité, des instances EC2 dans un groupe Auto Scaling, un Network Load Balancer et une connexion dédiée entre AWS l'environnement et l'environnement mainframe via. AWS Direct Connect



Dans cette architecture :

1. Lorsque le AWS Mainframe Modernization moteur d'exécution démarre, il récupère les informations d'identification Db2 [AWS Secrets Manager](#) et ouvre une connexion permanente avec Db2 for z/OS.

**Note**

AWS Mainframe Modernization Le service (Managed Runtime Environment Experience) n'est plus ouvert aux nouveaux clients. Pour des fonctionnalités similaires à AWS Mainframe Modernization Service (expérience d'environnement d'exécution géré), explorez AWS Mainframe Modernization Service (expérience autogérée). Les clients existants peuvent continuer à utiliser le service normalement. Pour plus d'informations, consultez [AWS Mainframe Modernization la section Modification de la disponibilité](#).

2. Les clients lient l'adresse Network Load Balancer dans [Amazon Route 53](#).
3. Route 53 redirige les transactions vers le Network Load Balancer.
4. Le Network Load Balancer répartit les transactions entre plusieurs instances EC2.
5. La charge de travail exécutée AWS Mainframe Modernization interagit avec Db2 car z/OS en utilisant une connexion persistante via. AWS Direct Connect

## Test

Lorsque vous reconfigurez une application COBOL tout en conservant DB2 en z/OS tant que base de données partagée, il est essentiel de s'assurer que le nouveau système fonctionne de la même manière que le système d'origine. Cet environnement hybride présente des défis et des opportunités de test uniques. La stratégie suivante décrit une approche complète des tests d'équivalence fonctionnelle et est conçue pour valider les performances, l'intégrité des données et l'intégration transparente de l'application replatforme avec la base de données Db2 for z/OS existante.

Commencez par identifier les processus métier et les transactions critiques qui doivent être comparés entre les systèmes. Créez ensuite un plan de test détaillé avec des scénarios spécifiques qui évalueront efficacement l'équivalence fonctionnelle de ces transactions. Enfin, développez des ensembles de données de test complets qui couvrent tous les scénarios identifiés et assurez-vous qu'ils sont identiques pour les deux systèmes afin de permettre une comparaison précise.

## Environnement source

- Instantané initial (premier instantané) :

- Assurez-vous que la table de données n'est pas utilisée par d'autres applications pendant le test, car cela peut affecter le test d'équivalence.
- Prenez un instantané du Db2 pour les z/OS tables utilisées par la transaction avant d'exécuter des tests.
- Test du système source :
  - Exécutez la suite complète de tests sur l'application COBOL d'origine.
  - Enregistrez toutes les transactions, entrées et sorties.
  - Surveillez les performances du système et l'utilisation des ressources.
- Instantané des tests effectués après la source (deuxième instantané) :
  - Prenez un autre instantané de la z/OS base de données DB2 for après avoir terminé les tests du système source.

## Environnement cible

- Réinitialiser la base de données :
  - Restaurez l'état initial de la base de données à l'aide du premier instantané.
- Test du système cible (environnement replatformé) :
  - Exécutez la même série de tests sur l'application replatformée.
  - Assurez-vous que tous les tests du système cible utilisent les mêmes entrées que les tests du système source.
  - Surveillez les performances du système et l'utilisation des ressources.
- Instantané du test post-cible (troisième instantané) :
  - Prenez un dernier instantané de la z/OS base de données DB2 for après avoir terminé les tests du système cible.

## Analyse

- Comparaison et analyse :
  - Comparez les deuxième et troisième instantanés pour identifier les éventuelles incohérences dans les données.
  - Analysez les résultats des tests et comparez les résultats des systèmes source et cible.

- Évaluez les indicateurs de performance entre les deux environnements.
- Tests d'intégration :
  - Effectuez des tests impliquant à la fois l'application replatforme et les composants COBOL restants.
  - Vérifiez une interaction fluide entre les deux environnements.
- Tests de reprise et de restauration :
  - Testez des scénarios dans lesquels un environnement tombe en panne et l'autre prend le relais.
  - Garantissez la cohérence et l'intégrité des données lors des situations de basculement.
- Tests de charge et de stress :
  - Effectuez des tests avec différentes charges pour évaluer le comportement du système hybride en cas de stress.
  - Identifiez les goulets d'étranglement ou les problèmes de performance dans l'un ou l'autre environnement.
- Documentation et rapports :
  - Documentez tous les résultats des tests, les écarts et les indicateurs de performance.
  - Préparez un rapport complet qui compare les systèmes source et cible.

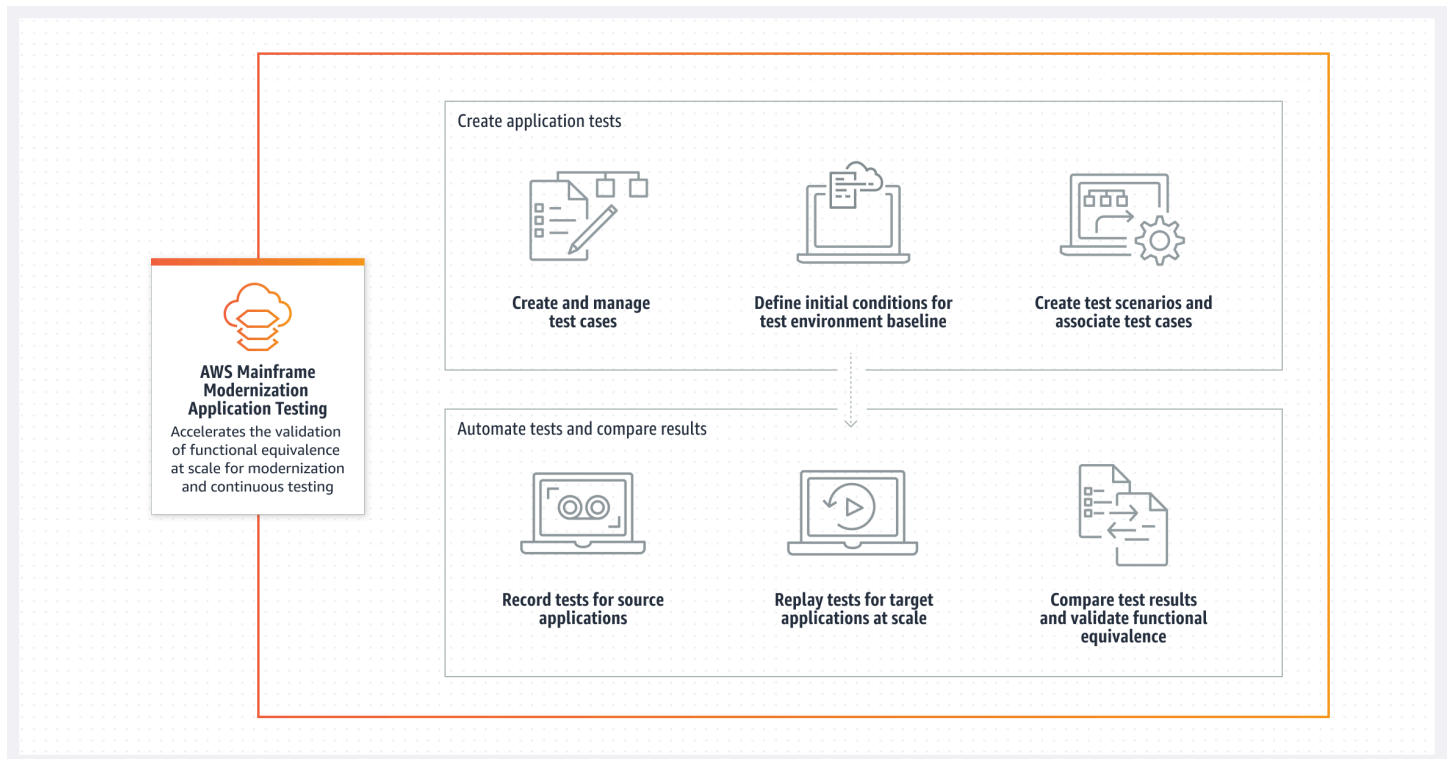
## Tester votre application dans AWS Mainframe Modernization

Le [AWS Mainframe Modernization Application Testings](#) service automatise l'exécution des tests d'applications à grande échelle. AWS Tests d'applications permet d'optimiser et de réduire les coûts des projets de modernisation des applications mainframe et de test.

### Note

AWS Mainframe Modernization Le service (Managed Runtime Environment Experience) n'est plus ouvert aux nouveaux clients. Pour des fonctionnalités similaires à AWS Mainframe Modernization Service (expérience d'environnement d'exécution géré), explorez AWS Mainframe Modernization Service (expérience autogérée). Les clients existants peuvent continuer à utiliser le service normalement. Pour plus d'informations, consultez [AWS Mainframe Modernization la section Modification de la disponibilité](#).

Le schéma suivant montre comment AWS Tests d'applications fonctionne à un niveau élevé.



Le processus comprend les étapes suivantes :

1. Créez et gérez les cas de test, qui constituent la plus petite unité d'actions de test. Identifiez les types de données qui représentent le mieux l'équivalence fonctionnelle entre les systèmes source et cible.
2. Définissez la configuration de l'environnement de test en spécifiant CloudFormation des modèles et des attributs supplémentaires.
3. Créez des suites de tests, qui sont des collections de cas de test.
4. Chargement et réexécution des ensembles de données : capturez les ensembles de données d'entrée et de sortie sur le mainframe, téléchargez-les AWS, puis rejouez le scénario de test sur le système cible.
5. Comparez les ensembles de données source et cible. AWS Tests d'applications compare automatiquement les ensembles de données de sortie des systèmes source et cible. Passez en revue et évaluez-les pour identifier les incohérences.

Pour plus d'informations, consultez la documentation [AWS Mainframe Modernization](#).

# Basculement

Dans le cadre de la modernisation des mainframes, l'un des défis les plus critiques consiste à minimiser les temps d'arrêt et les risques lors de la transition vers une nouvelle plateforme. La stratégie blue/green de déploiement propose une approche puissante et flexible de la migration du système.

Le déploiement bleu/vert est une technique qui réduit les temps d'arrêt et les risques en exécutant deux environnements de production identiques appelés bleu et vert. Voici comment cela fonctionne dans le contexte de la modernisation des ordinateurs centraux :

- Environnement bleu : il s'agit de votre système mainframe actuel qui gère l'ensemble du trafic de production.
- Environnement vert : il s'agit de votre nouvelle plateforme modernisée prête à prendre le relais.  
AWS

La blue/green stratégie de transition comprend les étapes suivantes : approvisionnement, mise en ligne, annulation en cas de problème et conclusion.

## Disposition

Au cours de cette étape, vous devez configurer le nouvel environnement (vert) AWS en suivant les étapes suivantes :

1. Replate-forme de l'environnement : la zone hébergée [Route 53](#) doit contenir un [enregistrement DNS](#) pointant vers l'environnement du mainframe (bleu).
2. Vérifiez la connectivité : assurez-vous que la connexion est correcte entre votre gestionnaire de transactions Compte AWS et celui des gestionnaires de transactions locaux et DB2 pour z/OS la base de données.
3. Exécutez des tests de fumée : utilisez l'adresse de l'équilibreur de AWS charge pour accéder à l'environnement replatforme et effectuez des tests de fumée complets pour vérifier les points suivants :
  - Toutes les charges de travail attendues sont disponibles.
  - 3270 transactions sont traitées correctement.
  - Les interactions de données avec Db2 for z/OS fonctionnent comme prévu.

## Passez en direct

Au cours de cette étape, vous déplacez le trafic vers l'environnement vert et vous surveillez les changements.

1. Utilisez les politiques de routage du trafic de Route 53 pour déplacer le trafic :
  - Option A : Vous pouvez déplacer le trafic en une seule fois.
  - Option B : Vous pouvez également utiliser une distribution pondérée progressive.
2. Surveillez et validez :
  - Surveillez attentivement l' AWS environnement lorsque le trafic change.
  - Vérifiez le traitement des transactions 3270.
  - Vérifiez Db2 pour la z/OS communication.
  - Surveillez les problèmes de performance.
  - Demandez aux utilisateurs de valider les résultats des transactions.

## Restauration

En cas de problème, vous pouvez rapidement mettre à jour Route 53 pour rediriger le trafic vers l'environnement mainframe local (bleu).

Vous devez étudier et résoudre les problèmes avant de tenter un nouveau transfert.

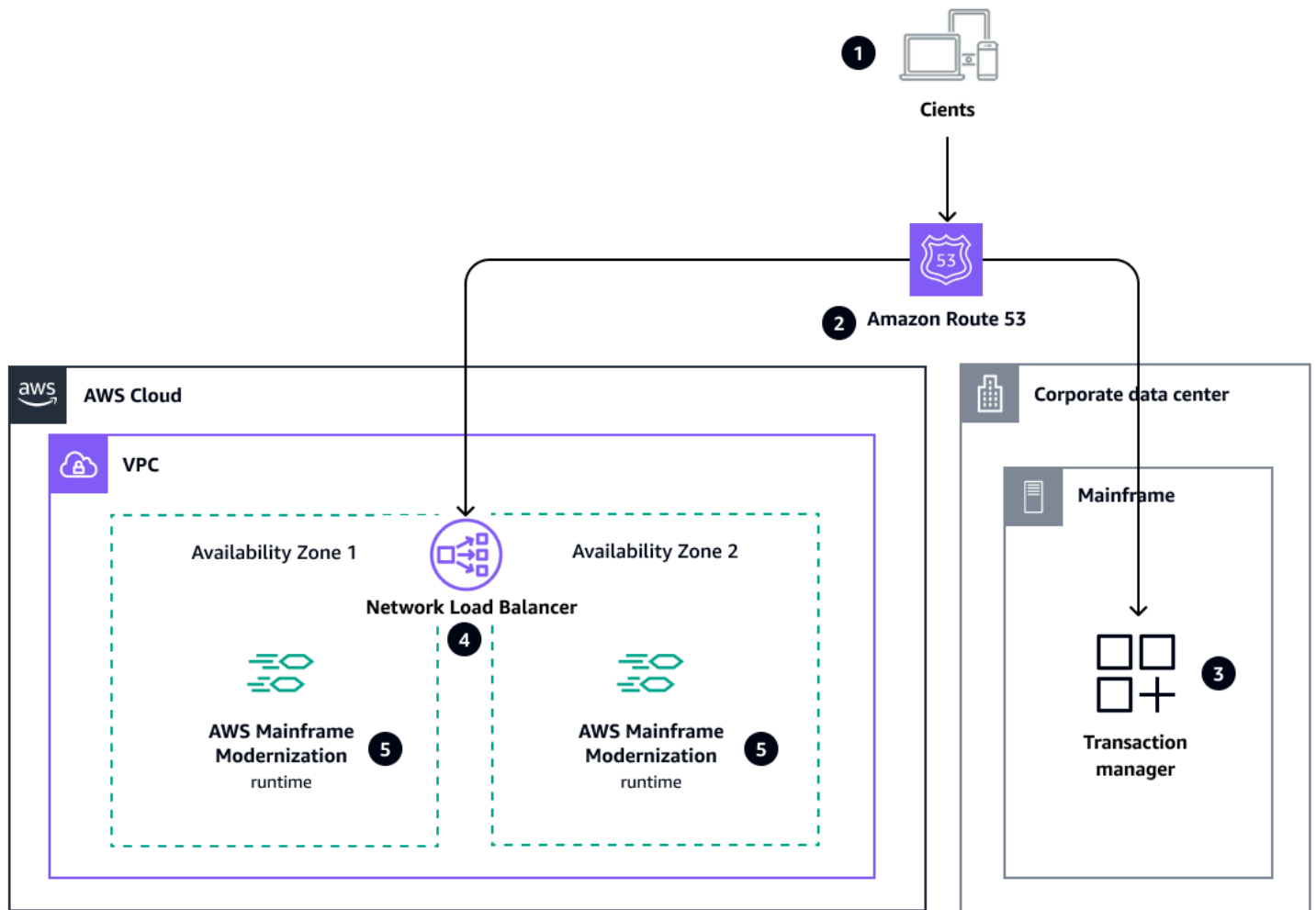
## Conclure

Après avoir surveillé le trafic et vérifié que votre environnement écologique fonctionne correctement, vous pouvez augmenter progressivement le trafic des applications à AWS.

Après une période stable, vous pouvez mettre hors service l'environnement de transactions du mainframe (bleu) et conserver la z/OS base de données DB2 for sur site.

## Architecture

Le schéma suivant illustre le flux de transition.



Le processus de transition comprend les étapes suivantes :

1. Les applications clientes, les frontends et les backends des frontends (BFFs) envoient des transactions au nom de domaine Route 53.
2. Route 53 achemine la connexion vers le gestionnaire de transactions du mainframe ou vers le Network Load Balancer, selon la politique de routage définie.
3. Le gestionnaire de transactions traite les transactions envoyées au mainframe.
4. Le Network Load Balancer distribue les transactions aux environnements de replatforme disponibles pour traitement.
5. Les environnements AWS Mainframe Modernization de replatforme traitent les demandes.

# Bonnes pratiques

Cette section décrit un ensemble de bonnes pratiques permettant de relever les principaux défis liés à la replateforme des charges de travail du mainframe vers des environnements cloud tout en conservant la base de données sur DB2 pour z/OS.

## Latence du réseau

Pour prévoir avec précision l'impact de latence lié à la séparation de l'application de la base de données DB2 lors d'un effort de replateforme, nous vous recommandons d'effectuer une évaluation approfondie du nombre d'appels Db2 pour les transactions et les processus par lots. Cette évaluation doit être réalisée à l'aide de données de trace et doit inclure les étapes suivantes :

- Collectez des données de suivi : collectez des traces détaillées de transactions représentatives et de tâches par lots, et assurez-vous que les traces capturent toutes les interactions Db2, y compris les entrées et les sorties.
- Analysez les données de suivi : comptez le nombre d'entrées et de sorties Db2 pour chaque transaction et chaque traitement par lots, et calculez le nombre moyen d'interactions Db2 par transaction et par traitement par lots.
- Mesurez les temps de réponse actuels : vérifiez si l'accès à DB2 est conforme au contrat de niveau de service (SLA) de votre application.
- Estimez la latence du réseau : déterminez la latence réseau attendue entre l'application replateforme et la base de données DB2. Tenez compte de facteurs tels que la distance physique, l'infrastructure réseau et les goulets d'étranglement potentiels.
- Calculez l'impact potentiel : pour chaque transaction et chaque traitement par lots, multipliez le nombre d'entrées et de sorties de Db2 par la latence estimée du réseau. Ajoutez ce temps calculé aux temps de réponse actuels pour prévoir le nouveau temps de traitement total.
- Évaluez les résultats : déterminez si l'augmentation de latence prévue est acceptable pour les besoins de votre entreprise et identifiez les transactions ou les processus susceptibles de nécessiter une optimisation ou une refonte pour atténuer les problèmes de latence.
- Envisagez des stratégies d'atténuation : explorez des options telles que le regroupement de connexions, la mise en cache ou la récupération de données par lots afin de réduire le nombre d'interactions Db2 individuelles. Évaluez la possibilité de rapprocher les données fréquemment consultées du niveau de l'application.

En suivant ces étapes, vous serez en mesure de prendre des décisions basées sur les données concernant la faisabilité de votre stratégie de replatforme et d'identifier les éventuels problèmes de performance avant qu'ils n'aient un impact sur votre environnement de production. Cette approche permettra de garantir une transition fluide tout en maintenant des niveaux de performance acceptables pour vos applications dépendantes de la base de données.

## Sécurité

- Sécurisez le développement de votre application : utilisez un sous-réseau privé dans le cloud privé virtuel (VPC) pour l' AWS CodeBuild exécuter afin de garantir l'isolation et une sécurité renforcée. Implémentez un contexte sécurisé Db2 à partir du CodeBuild sous-réseau CIDR pour un accès sécurisé à la base de données pendant le processus de création.
- Sécurisez votre environnement d'exécution : utilisez un contexte sécurisé Db2 issu du sous-réseau d'exécution CIDR pour des connexions de base de données sécurisées.
- Gérez les informations d'identification de la base de données en toute sécurité : mettez en œuvre un calendrier régulier de rotation des informations d'identification afin de minimiser le risque d'accès non autorisé. Stockez les informations d'identification DB2 en toute sécurité dans AWS Secrets Manager
- Établissez la sécurité du réseau : mettez en œuvre des règles strictes de segmentation du réseau et de pare-feu pour protéger à la fois les environnements de construction et d'exécution. Utilisez la bonne combinaison de AWS Direct Connect et AWS Site-to-Site VPN pour atteindre le niveau de sécurité nécessaire.
- Appliquer le chiffrement : appliquez le chiffrement des données en transit entre votre application et Db2 for z/OS.

## Gouvernance des applications

- Établissez une source fiable : définissez le nouveau système de gestion de la configuration logicielle (SCM), par exemple, GitHub comme source fiable unique pour le code d'application migré. Cela garantit la cohérence et élimine les différences de version entre les environnements cloud et mainframe pendant la période de transition.
- Mettre à jour le processus de gestion des modifications : Mettez à jour le processus de gestion des modifications afin de prendre en compte les modifications du code dans ce nouveau paradigme à double environnement. Ce processus devrait inclure :

- Des flux de travail d'approbation clairs pour les modifications de code.
- Révisions de code obligatoires avant de fusionner le code dans la branche principale.
- Procédures de déploiement synchronisées pour garantir que les deux environnements reçoivent des mises à jour simultanément.
- Mécanismes de restauration en cas de problème dans l'un ou l'autre environnement.

## Élasticité

L'élasticité du cloud computing introduit un changement de paradigme qui modifie de manière significative la structure de coûts et la gestion des ressources du mainframe. Contrairement à l'environnement mainframe traditionnel, qui repose sur des modèles de tarification à capacité fixe et basés sur les pics, les plateformes cloud offrent une évolutivité dynamique et une pay-as-you-go approche susceptible de permettre de réaliser des économies substantielles et d'améliorer l'efficacité opérationnelle.

Dans un environnement cloud, les entreprises peuvent augmenter ou diminuer leurs ressources informatiques en temps réel en fonction de la demande réelle, ce qui élimine le besoin de surprovisionnement pour faire face aux pics de charge. Cette élasticité permet aux entreprises de ne payer que pour les ressources qu'elles consomment au lieu d'investir dans des licences matérielles et logicielles coûteuses pour faire face à des pics d'utilisation occasionnels.

Pour plus de détails sur le fonctionnement de la tarification AWS, consultez la section [AWS Tarification](#).

## Étapes suivantes

La modernisation du mainframe est une initiative complexe et critique qui exige des connaissances spécialisées et des solutions avancées. Vous pouvez accélérer votre processus de modernisation et obtenir des résultats commerciaux plus rapides grâce à [des partenariats stratégiques](#) qui vous aideront à accomplir les tâches suivantes :

- **Évaluez et hiérarchisez** : passez en revue vos applications mainframe et identifiez celles qui conviennent à la replatforme tout en conservant la base de données sur DB2 pour z/OS. Tenez compte de facteurs tels que la complexité, la criticité commerciale et le retour sur investissement (ROI) potentiel.
- **Élaborez une stratégie de migration** : créez un plan détaillé pour la replatforme des applications que vous avez sélectionnées, y compris les délais, l'allocation des ressources et les stratégies d'atténuation des risques.
- **Évaluer les outils et les technologies** : recherchez et sélectionnez les outils et technologies appropriés pour faciliter le processus de replatforme, tels que les plateformes de modernisation des applications ou les outils de conversion de code.
- **Communiquez avec des experts** : envisagez de vous associer à des spécialistes de la modernisation des ordinateurs centraux ou à des sociétés de conseil ayant de l'expérience dans le domaine des projets de replatforme.
- **Preuve de concept** : Commencez par une validation de concept à petite échelle pour valider votre approche et identifier les défis potentiels avant de passer à des applications plus importantes.
- **Tests et validation** : développez une stratégie de test complète pour garantir que vos applications replatformes fonctionnent correctement et préservent l'intégrité des données avec votre base de données DB2 for existante. z/OS
- **Formation et transfert de connaissances** : préparez votre équipe au nouvel environnement en dispensant une formation sur les applications replatformes et sur les nouveaux outils ou technologies introduits.
- **Implémentation progressive** : envisagez une approche progressive de la replatforme, dans le cadre de laquelle vous migrez progressivement les applications tout en surveillant les performances et en résolvant les problèmes éventuels.
- **Optimisation continue** : après la replatforme, surveillez et optimisez en permanence les performances de vos applications et leurs interactions avec la z/OS base de données DB2 pour garantir un succès à long terme.

- Modernisez à votre rythme : Maintenant que la charge de travail est épuisée AWS et que vous tirez déjà parti du cloud, commencez à planifier la phase de refonte de votre modernisation.

## Ressources

Pour plus d'informations sur la migration et la modernisation des mainframes, consultez les ressources suivantes.

### AWS documentation

- [Configuration d'Amazon Route 53 en tant que service DNS](#)
- [Acheminement du trafic vers un équilibreur de charge ELB](#)
- [Routage pondéré](#)
- [Replateforme d'applications avec Rocket Software](#)

### Références de Rocket Software

- [Interface d'appel externe \(ECI\) Micro Focus](#)
- [Services Web du CICS](#)

### Références IBM

- [Contextes fiables](#) (IBM Db2 pour la z/OS documentation)

### Outils

- [Serveur Rocket Enterprise](#)

### AWS Modèles et guides d'orientation prescriptifs

- [Créez des programmes COBOL DB2 en utilisant et AWS Mainframe ModernizationAWS CodeBuild](#)
- [DevOps pour AWS Mainframe Modernization](#)
- [Modernisation du mainframe : modèles de découplage pour la migration du code d'application](#)
- [Sécurisez et rationalisez l'accès des utilisateurs dans une base de données de fédération DB2 AWS en utilisant des contextes fiables](#)

## 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
<a href="#">Publication initiale</a>	—	7 mai 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 sur site vers Oracle sur une instance EC2 dans le 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 réplique 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 etMAX.

## 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 Gestion des identités et des accès AWS (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'une 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, c'est 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

Voir [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 ou Bitbucket 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 Gestion des identités et des accès AWS (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).

## environnement

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 étapes 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 replatforme. 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

### laC

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.

### Ilo 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

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 connus sous le nom de 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'exception des 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 AWS avec le service de migration 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 peuvent accéder au contenu d'un compartiment S3 uniquement via 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 informations de contact.

### 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](#).

## policy

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.

## 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 les 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 \(RACI\)](#).

### RAG

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

Un 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 en matière AWS de sécurité. Parmi les actions de réponse automatique, citons la modification d'un groupe de sécurité VPC, l'application de correctifs à une instance Amazon EC2 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 définissent 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, voir [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

## tags

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.