



Implémentation de PostgreSQL géré pour les applications SaaS multi-locataires sur AWS

# AWS Conseils prescriptifs



# AWS Conseils prescriptifs: Implémentation de PostgreSQL géré pour les applications SaaS multi-locataires sur AWS

Copyright © 2024 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 commerciaux ciblés .....	1
Sélection d'une base de données pour une application SaaS .....	3
Choisir entre Amazon RDS et Aurora .....	6
Modèles de partitionnement SaaS multi-locataires pour PostgreSQL .....	8
Modèle de silo PostgreSQL .....	9
Modèle de groupe PostgreSQL .....	10
Modèle de pont PostgreSQL .....	12
Matrice de décision .....	15
Recommandations relatives à la sécurité au niveau des lignes .....	33
Disponibilité de PostgreSQL pour le modèle de pool .....	35
Bonnes pratiques .....	37
Comparez AWS les options pour PostgreSQL géré .....	37
Sélectionnez un modèle de partitionnement SaaS multi-locataires .....	37
Utiliser la sécurité au niveau des lignes pour les modèles de partitionnement SaaS de pool .....	37
FAQ .....	39
Quelles sont les options de gestion de PostgreSQL AWS proposées ? .....	39
Quel service est optimal pour les applications SaaS ? .....	39
Quelles exigences uniques dois-je prendre en compte si je décide d'utiliser une base de données PostgreSQL avec une application SaaS multi-tenant ? .....	39
Quels modèles puis-je utiliser pour maintenir l'isolation des données des locataires avec PostgreSQL ? .....	40
Comment maintenir l'isolation des données des locataires avec une base de données PostgreSQL unique partagée entre plusieurs locataires ? .....	40
Étapes suivantes .....	41
Ressources .....	42
Références .....	42
Partenaires .....	42
Historique du document .....	43
Glossaire .....	44
# .....	44
A .....	45
B .....	48
C .....	50

---

D .....	53
E .....	58
F .....	60
G .....	61
H .....	62
I .....	64
L .....	66
M .....	67
O .....	72
P .....	74
Q .....	77
R .....	78
S .....	81
T .....	84
U .....	86
V .....	86
W .....	87
Z .....	88
.....	lxxxix

# Implémentation de PostgreSQL géré pour les applications SaaS multi-locataires sur AWS

Tabby Ward et Thomas Davis, Amazon Web Services (AWS)

Avril 2024 ([historique du document](#))

Lorsque vous sélectionnez une base de données pour stocker des données opérationnelles, il est essentiel de prendre en compte la manière dont les données doivent être structurées, les requêtes auxquelles elles répondront, la rapidité avec laquelle elles fourniront des réponses et la résilience de la plate-forme de données elle-même. À ces considérations générales s'ajoutent les implications du logiciel en tant que service (SaaS) pour les données opérationnelles, telles que l'isolation des performances, la sécurité des locataires, ainsi que les caractéristiques uniques et les modèles de conception typiques des données pour les applications SaaS multi-locataires. Ce guide explique comment ces facteurs s'appliquent à l'utilisation d'une base de données PostgreSQL sur Amazon Web Services AWS() comme principal magasin de données opérationnelles pour une application SaaS mutualisée. Plus précisément, le guide se concentre sur deux options de AWS gestion de PostgreSQL : Amazon Aurora PostgreSQL Compatible Edition et Amazon Relational Database Service (Amazon RDS) pour PostgreSQL.

## Résultats commerciaux ciblés

Ce guide fournit une analyse détaillée des meilleures pratiques pour les applications SaaS multi-locataires utilisant Aurora PostgreSQL compatible et Amazon RDS for PostgreSQL. Nous vous recommandons d'utiliser les modèles et concepts de conception fournis dans ce guide pour informer et standardiser votre mise en œuvre d'Aurora PostgreSQL compatible ou d'Amazon RDS for PostgreSQL pour vos applications SaaS mutualisées.

Ces directives prescriptives aident à atteindre les résultats commerciaux suivants :

- Choix de l'option PostgreSQL AWS gérée la mieux adaptée à votre cas d'utilisation : ce guide compare les options relationnelles et non relationnelles pour l'utilisation des bases de données avec les applications SaaS. Il décrit également les cas d'utilisation les plus optimaux pour la compatibilité avec Aurora PostgreSQL et Amazon RDS for PostgreSQL. Ces informations vous aideront à sélectionner la meilleure option pour votre application SaaS.

- Application des meilleures pratiques SaaS par l'adoption d'un modèle de partitionnement SaaS — Ce guide décrit et compare trois grands modèles de partitionnement SaaS applicables à un système de gestion de base de données (DBMS) PostgreSQL : les modèles de pool, de pont et de silo, ainsi que leurs variantes. Ces approches reflètent les meilleures pratiques du SaaS et offrent de la flexibilité lors de la conception d'une application SaaS. L'application d'un modèle de partitionnement SaaS est essentielle à la préservation des meilleures pratiques.
- Utilisation efficace du RLS dans les modèles de partitionnement SaaS de pool : la sécurité au niveau des lignes (RLS) permet d'appliquer l'isolation des données des locataires au sein d'une seule table PostgreSQL en limitant les lignes pouvant être consultées en fonction de l'utilisateur ou d'une variable de contexte. Lorsque vous utilisez le modèle de partitionnement du pool, le protocole RLS est requis pour empêcher l'accès entre locataires.

# Sélection d'une base de données pour une application SaaS

Pour de nombreuses applications SaaS à locataires multiples, la sélection d'une base de données opérationnelle peut se résumer à un choix entre des bases de données relationnelles et non relationnelles, ou à une combinaison des deux. Pour prendre votre décision, tenez compte de ces exigences et caractéristiques de haut niveau en matière de données d'application :

- Modèle de données de l'application
- Modèles d'accès aux données
- Exigences de latence de la base
- Exigences relatives à l'intégrité des données et à l'intégrité transactionnelle (atomicité, cohérence, isolation et durabilité, ou ACID)
- Exigences relatives à la disponibilité et à la restauration entre les régions

Le tableau suivant répertorie les exigences et les caractéristiques des données des applications, et les aborde dans le contexte des offres de AWS base de données : compatible avec Aurora PostgreSQL et Amazon RDS for PostgreSQL (relationnel), et Amazon DynamoDB (non relationnel). Vous pouvez faire référence à cette matrice lorsque vous essayez de choisir entre des offres de bases de données opérationnelles relationnelles et non relationnelles.

Bases de données	Exigences et caractéristiques des données des applications SaaS				
	Modèle de données	Modèles d'accès	Exigences de latence	Intégrité des données et des transactions	Disponibilité et restauration entre les régions
Relationnel (compatible avec Aurora PostgreSQL et Amazon RDS pour PostgreSQL)	Relationnel ou hautement normalisé.	Il n'est pas nécessaire de le planifier minutieusement à l'avance.	Tolérance de latence de préférence plus élevée ; permet d'obtenir des latences plus faibles par	Intégrité élevée des données et des transactions maintenue par défaut.	Dans Amazon RDS, vous pouvez créer une réplique en lecture pour le dimension

défaut avec Aurora et en implémentant des répliques de lecture, la mise en cache et des fonctionnalités similaires.

nement et le basculement entre régions.

[Aurora automatisé principalement ce processus](#)

. Pour les configurations active-active sur plusieurs Régions AWS, vous pouvez utiliser le [transfert d'écriture conjointe](#) avec les bases de données [globales Aurora](#).

Non relationnel (Amazon DynamoDB)	Généralement dénormalisé. Ces bases de données tirent parti des modèles pour modéliser <a href="#">many-to-many relations, les éléments volumineux et les données de séries chronologiques</a> .	Tous les modèles d'accès (requêtes) aux données doivent être parfaitement compris avant qu'un modèle de données ne soit produit.	Très faible latence grâce à des options telles qu'Amazon DynamoDB Accelerator (DAX) capables d'améliorer encore les performances.	Intégrité transactionnelle optionnelle au détriment des performances. Les problèmes d'intégrité des données sont transférés à l'application.	Restauration facile entre régions et configuration active-active avec des tables globales. (La conformité ACID n'est réalisable que dans une seule AWS région.)
--------------------------------------	--	--	---	--	---

Certaines applications SaaS mutualisées peuvent avoir des modèles de données uniques ou des circonstances particulières qui sont mieux desservies par des bases de données non incluses dans le tableau précédent. Par exemple, les ensembles de données chronologiques, les ensembles de données hautement connectés ou la gestion d'un registre de transactions centralisé peuvent nécessiter l'utilisation d'un autre type de base de données. L'analyse de toutes les possibilités dépasse le cadre de ce guide. Pour obtenir une liste complète des offres de AWS base de données et savoir comment elles peuvent répondre à différents cas d'utilisation à un niveau élevé, consultez la section [Base](#) de données du livre blanc Présentation d'Amazon Web Services.

Le reste de ce guide se concentre sur les services de base de données AWS relationnelle compatibles avec PostgreSQL : Amazon RDS et Aurora PostgreSQL compatibles. DynamoDB nécessite une approche différente pour optimiser les applications SaaS, ce qui dépasse le cadre de ce guide. Pour plus d'informations sur DynamoDB, consultez le billet de [blog Partitioning Pooled AWS Multi-Tenant SaaS](#) Data with Amazon DynamoDB.

## Choisir entre Amazon RDS et Aurora

Dans la plupart des cas, nous recommandons d'utiliser Aurora PostgreSQL compatible plutôt qu'Amazon RDS for PostgreSQL. Le tableau suivant indique les facteurs à prendre en compte au moment de choisir entre ces deux options.

composant DBMS	Amazon RDS for PostgreSQL	Compatible avec Aurora avec PostgreSQL
Evolutivité	Délai de réplication de quelques minutes, maximum de 5 répliques de lecture	Retard de réplication inférieur à une minute (généralement inférieur à 1 seconde avec les bases de données globales), maximum de 15 répliques de lecture
Reprise après un crash	Les points de contrôle espacés de 5 minutes (par défaut) peuvent ralentir les performances de la base de données	Restauration asynchrone avec threads parallèles pour une restauration rapide
Basculément	60 à 120 secondes en plus du temps de récupération en cas de panne	Généralement environ 30 secondes (y compris la récupération en cas de panne)
Stockage	Nombre maximal d'IOPS de 256 000	Les IOPS sont limitées uniquement par la taille et la capacité de l'instance Aurora
Haute disponibilité et reprise après sinistre	Deux zones de disponibilité avec une instance de secours, basculement entre régions pour lire les répliques ou les sauvegardes copiées	Trois zones de disponibilité par défaut, basculement entre régions avec les bases de données globales Aurora, <a href="#">transfert d'écriture</a> Régions AWS pour les configurations active-active

composant DBMS	Amazon RDS for PostgreSQL	Compatible avec Aurora avec PostgreSQL
Sauvegarde	Pendant la fenêtre de sauvegarde, cela peut avoir un impact sur les performances	Sauvegardes incrémentielles automatiques, sans impact sur les performances
classes d'instances de base de données	Voir <a href="#">la liste des classes d'instances Amazon RDS</a>	Voir <a href="#">la liste des classes d'instances Aurora</a>

Dans toutes les catégories décrites dans le tableau précédent, la compatibilité avec Aurora PostgreSQL est généralement la meilleure option. Cependant, Amazon RDS for PostgreSQL peut toujours être utile pour les charges de travail petites et moyennes, car il propose un plus grand choix de classes d'instances, ce qui peut constituer une option plus rentable au détriment de l'ensemble de fonctionnalités plus robustes d'Aurora.

# Modèles de partitionnement SaaS multi-locataires pour PostgreSQL

La meilleure méthode pour réaliser la mutualisation dépend des exigences de votre application SaaS. Les sections suivantes présentent des modèles de partitionnement permettant de mettre en œuvre avec succès la mutualisation dans PostgreSQL.

## Note

Les modèles présentés dans cette section s'appliquent à la fois à Amazon RDS for PostgreSQL et à Aurora PostgreSQL compatible. Les références à PostgreSQL dans cette section s'appliquent aux deux services.

Il existe trois modèles de haut niveau que vous pouvez utiliser dans PostgreSQL pour le partitionnement SaaS : silo, bridge et pool. L'image suivante résume les compromis entre les modèles de silo et de pool. Le modèle de pont est un hybride des modèles de silo et de piscine.

Modèle de partitionnement	Avantages	Inconvénients
Silo	<ul style="list-style-type: none"> <li>Alignement</li> <li>Aucun impact entre locataires</li> <li>Réglage au niveau du locataire</li> <li>Disponibilité au niveau du locataire</li> </ul>	<ul style="list-style-type: none"> <li>Agilité</li> <li>Pas de gestion centralisée</li> <li>Complexité du déploiement</li> <li>Coût</li> </ul>
Piscine	<ul style="list-style-type: none"> <li>Agilité</li> <li>Optimisation des coûts</li> <li>Gestion centralisée</li> <li>Déploiement simplifié</li> </ul>	<ul style="list-style-type: none"> <li>Impact sur les différents locataires</li> <li>Alertes de mise</li> <li>Disponibilité du tout ou rien</li> </ul>
Pont	<ul style="list-style-type: none"> <li>Un certain alignement en matière de conformité</li> </ul>	<ul style="list-style-type: none"> <li>Quelques défis en matière de conformité</li> </ul>

Modèle de partitionnement	Avantages	Inconvénients
	<ul style="list-style-type: none"><li>• Agilité</li><li>• Optimisation des coûts</li><li>• Gestion centralisée</li></ul>	<ul style="list-style-type: none"><li>• Disponibilité « tout ou rien » (la plupart du temps)</li><li>• Impact sur les différents locataires</li><li>• Complexité du déploiement</li></ul>

Les sections suivantes décrivent chaque modèle plus en détail.

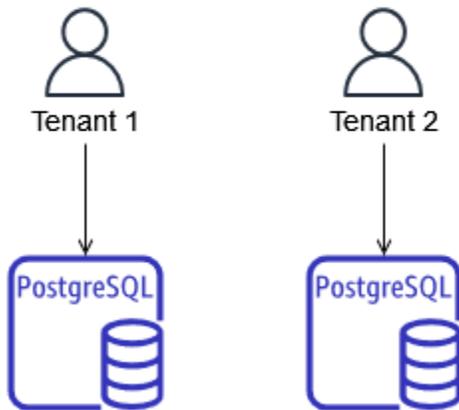
Modèles de partitionnement :

- [Modèle de silo PostgreSQL](#)
- [Modèle de groupe PostgreSQL](#)
- [Modèle de pont PostgreSQL](#)
- [Matrice de décision](#)

## Modèle de silo PostgreSQL

Le modèle de silo est mis en œuvre en fournissant une instance PostgreSQL pour chaque locataire d'une application. Le modèle de silo excelle en termes de performance des locataires et d'isolation en matière de sécurité, et élimine complètement le phénomène des voisins bruyants. Le phénomène du voisinage bruyant se produit lorsque l'utilisation d'un système par un locataire affecte les performances d'un autre locataire. Le modèle de silo vous permet d'adapter les performances spécifiquement à chaque locataire et de limiter éventuellement les pannes au silo d'un locataire spécifique. Cependant, ce qui motive généralement l'adoption d'un modèle de silo, ce sont les contraintes réglementaires et de sécurité strictes. Ces contraintes peuvent être motivées par les clients SaaS. Par exemple, les clients SaaS peuvent exiger que leurs données soient isolées en raison de contraintes internes, et les fournisseurs SaaS peuvent proposer un tel service moyennant des frais supplémentaires.

### Silo model (separate PostgreSQL instances or clusters for each tenant)

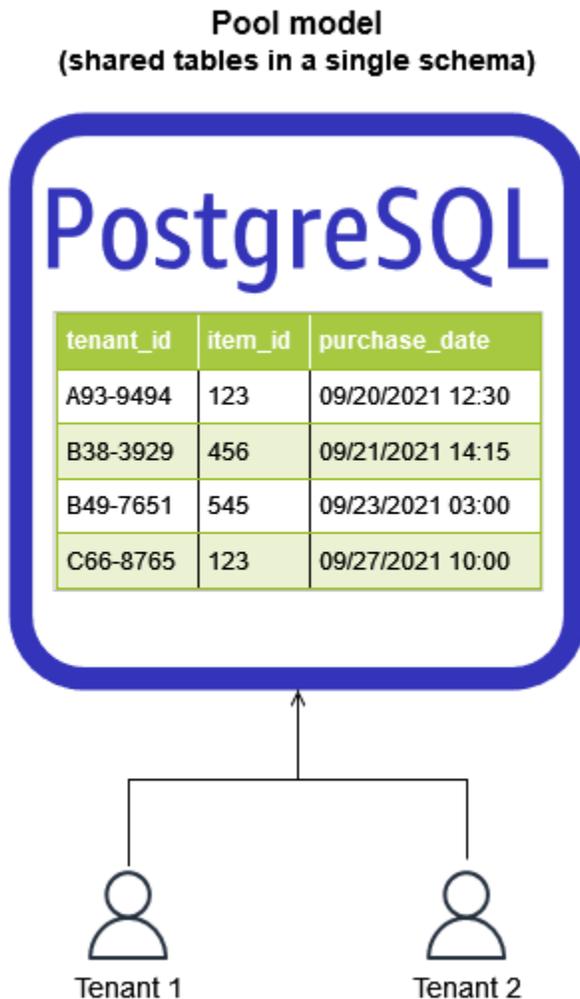


Bien que le modèle de silo puisse être nécessaire dans certains cas, il présente de nombreux inconvénients. Il est souvent difficile d'utiliser le modèle de silo de manière rentable, car la gestion de la consommation de ressources sur plusieurs instances de PostgreSQL peut s'avérer complexe. En outre, la nature distribuée des charges de travail des bases de données dans ce modèle rend plus difficile le maintien d'une vue centralisée de l'activité des locataires. La gestion d'un si grand nombre de charges de travail indépendantes augmente les frais opérationnels et administratifs. Le modèle en silo complique également l'intégration des locataires et leur fait perdre du temps, car vous devez fournir des ressources spécifiques aux locataires. En outre, l'ensemble du système SaaS peut être plus difficile à faire évoluer, car le nombre toujours croissant d'instances PostgreSQL spécifiques aux locataires demandera plus de temps opérationnel à administrer. Une dernière considération est qu'une application ou une couche d'accès aux données devra maintenir un mappage des locataires avec leurs instances PostgreSQL associées, ce qui complique encore la mise en œuvre de ce modèle.

## Modèle de groupe PostgreSQL

Le modèle de pool est mis en œuvre en provisionnant une instance PostgreSQL unique (Amazon RDS ou Aurora) et en utilisant la [sécurité au niveau des lignes \(RLS\)](#) pour maintenir l'isolation des données des locataires. Les politiques RLS limitent les lignes d'une table qui sont SELECT renvoyées par les requêtes ou les lignes affectées par DELETE les commandes INSERT UPDATE, et. Le modèle de pool centralise toutes les données des locataires dans un seul schéma PostgreSQL, ce qui le rend nettement plus rentable et nécessite moins de frais opérationnels pour sa maintenance. La surveillance de cette solution est également nettement plus simple grâce à sa centralisation.

Cependant, la surveillance des impacts spécifiques aux locataires dans le modèle de piscine nécessite généralement des instruments supplémentaires dans l'application. Cela est dû au fait que PostgreSQL ne sait pas par défaut quel locataire consomme des ressources. L'intégration des locataires est simplifiée car aucune nouvelle infrastructure n'est requise. Cette agilité facilite la mise en œuvre de flux de travail rapides et automatisés pour l'intégration des locataires.



Bien que le modèle de pool soit généralement plus rentable et plus simple à administrer, il présente certains inconvénients. Le phénomène de voisinage bruyant ne peut pas être complètement éliminé dans un modèle de piscine. Il est toutefois possible d'y remédier en veillant à ce que les ressources appropriées soient disponibles sur l'instance PostgreSQL et en utilisant des stratégies visant à réduire la charge dans PostgreSQL, telles que le déchargement des requêtes pour lire des répliques ou vers Amazon ElastiCache. Une surveillance efficace joue également un rôle dans la réponse aux préoccupations relatives à l'isolation des performances des locataires, car l'instrumentation

des applications peut enregistrer et surveiller les activités spécifiques des locataires. Enfin, certains clients SaaS peuvent ne pas trouver la séparation logique fournie par RLS suffisante et demander des mesures d'isolation supplémentaires.

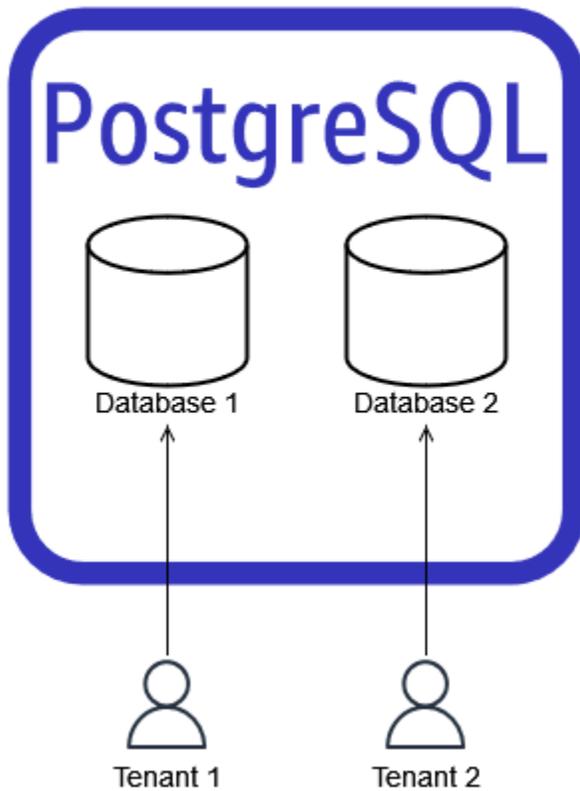
## Modèle de pont PostgreSQL

Le modèle de pont PostgreSQL est une combinaison d'approches groupées et cloisonnées. Comme dans le modèle groupé, vous fournissez une instance PostgreSQL unique pour chaque locataire. Pour maintenir l'isolation des données des locataires, vous utilisez des constructions logiques PostgreSQL. Dans le schéma suivant, les bases de données PostgreSQL sont utilisées pour séparer logiquement les données.

### Note

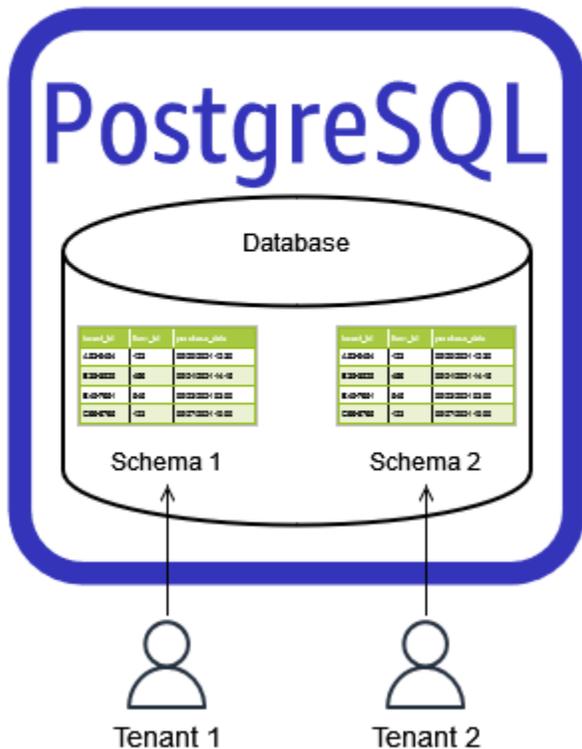
Une base de données PostgreSQL ne fait pas référence à une instance de base de données compatible Amazon RDS for PostgreSQL ou Aurora PostgreSQL distincte. Il fait plutôt référence à une construction logique du système de gestion de base de données PostgreSQL pour séparer les données.

### Bridge model with separate databases (separate databases in a single instance)



Vous pouvez également implémenter le modèle de pont en utilisant une seule base de données PostgreSQL, avec des schémas spécifiques au locataire dans chaque base de données, comme illustré dans le schéma suivant.

## Bridge model with separate schemas (separate schemas in a single database)



Le modèle de pont présente les mêmes problèmes d'isolation des performances des voisins et des locataires bruyants que le modèle de piscine. Cela entraîne également des frais d'exploitation et de provisionnement supplémentaires en exigeant la mise en service de bases de données ou de schémas distincts par locataire. Cela nécessite une surveillance efficace pour répondre rapidement aux préoccupations des locataires en matière de performance. Cela nécessite également une instrumentation d'application pour surveiller l'utilisation spécifique au locataire. Dans l'ensemble, le modèle bridge peut être considéré comme une alternative au RLS qui augmente légèrement l'effort d'intégration des locataires en nécessitant de nouvelles bases de données ou de nouveaux schémas PostgreSQL. Comme pour le modèle de silo, une application ou une couche d'accès aux données devra gérer un mappage des locataires avec leurs bases de données ou schémas PostgreSQL associés.

## Matrice de décision

Pour décider quel modèle de partitionnement SaaS multi-locataires vous devez utiliser avec PostgreSQL, consultez la matrice de décision suivante. La matrice analyse ces quatre options de partitionnement :

- Silo — instance ou cluster PostgreSQL distinct pour chaque locataire.
- Pont avec bases de données distinctes : une base de données distincte pour chaque tenant dans une instance ou un cluster PostgreSQL unique.
- Pont avec des schémas distincts : schéma distinct pour chaque tenant dans une base de données PostgreSQL unique, dans une instance ou un cluster PostgreSQL unique.
- Pool : tables partagées pour les locataires dans une instance et un schéma uniques.

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Cas d'utilisation	L'isolation des données avec un contrôle total de l'utilisation des ressources est une exigence essentielle, sinon vous avez des locataires très importants et très sensibles aux performances.	L'isolation des données est une exigence essentielle, et des références croisées limitées ou inexistantes aux données des locataires sont requises.	Nombre modéré de locataires avec une quantité modérée de données. Il s'agit du modèle préféré si vous devez croiser les données des locataires.	Grand nombre de locataires avec moins de données par locataire.
Agilité d'intégration des	Très (Une nouvelle instance ou un	Moyennement (Nécessite la création d'une	Moyennement (Nécessite de créer un	Option la plus rapide. (Une configuration

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
nouveaux locataires	nouveau cluster est requis pour chaque locataire .)	nouvelle base de données pour que chaque locataire puisse stocker les objets du schéma.)	nouveau schéma pour que chaque locataire puisse stocker des objets.)	minimale est requise.)

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Effort et efficacité de configuration du pool de connexions de base de données	<p>Des efforts importants sont nécessaires. (Un pool de connexions par locataire.)</p> <p>Moins efficace. (Aucun partage de connexion à la base de données entre les locataires.)</p>	<p>Des efforts importants sont nécessaires. (Une configuration de pool de connexions par locataire, sauf si vous utilisez le <a href="#">proxy Amazon RDS</a>.)</p> <p>Moins efficace. (Aucun partage de connexion à la base de données entre les locataires et le nombre total de connexions. L'utilisation par tous les locataires est fonction de la classe d'instance de base de données.)</p>	<p>Moins d'efforts requis. (Configuration d'un pool de connexions unique pour tous les locataires.)</p> <p>Modérément efficace. (Réutilisation de la connexion via la SET SCHEMA commande SET ROLE ou en mode pool de sessions uniquement. Les commandes provoquent également l'épinglage de session lors de l'utilisation d'Amazon RDS Proxy, mais les pools de connexions clients peuvent être éliminés et des connexions</p>	<p>Le moins d'effort requis.</p> <p>Très efficace (Un pool de connexions pour tous les locataires et une réutilisation efficace des connexions entre tous les locataires. Les limites de connexion à la base de données sont fonction de la classe d'instance de base de données.)</p>

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
			directes peuvent être établies pour chaque demande pour des raisons d'efficacité.)	

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Maintenance de la base de <a href="#">données (gestion des vides)</a> et utilisation des ressources	Gestion simplifiée.	Complexité (Cela peut entraîner une forte consommation de ressources, car il faut ensuite démarrer un aspirateur pour chaque base de données <code>vacuum_timeout</code> , ce qui entraîne une utilisation importante du processeur du lanceur automatique. Le nettoyage des tables du catalogue du système PostgreSQL pour chaque base de données peut également entraîner une surcharge supplémentaire.)	Tableaux de catalogue de grande taille pour le système PostgreSQL. ( <code>pg_catalog</code> Taille totale en dizaines de Go, en fonction du nombre de locataires et de relations). Il faudra probablement modifier les paramètres liés à l'aspiration pour contrôler le gonflement de la table.)	Les tables peuvent être volumineuses, en fonction du nombre de locataires et des données par locataire. (Il faudra probablement modifier les paramètres liés à l'aspiration pour contrôler le gonflement de la table.)

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Effort de gestion des extensions	Effort important (pour chaque base de données dans des instances distinctes).	Effort important (au niveau de chaque base de données).	Effort minimal (une seule fois dans la base de données commune).	Effort minimal (une seule fois dans la base de données commune).
Effort de déploiement des modifications	Alertes d'effort (Connect à chaque instance distincte et déployez les modifications.)	Alertes d'effort (Connect à chaque base de données et à chaque schéma, puis déployez les modifications.)	Effort (Connect à une base de données commune et déployez les modifications pour chaque schéma.)	Niveau d'effort (Connect à une base de données commune et déployez les modifications.)
Déploiement du changement — étendue de l'impact	Minimale. (Locataire célibataire concerné.)	Minimale. (Locataire célibataire concerné.)	Minimale. (Locataire célibataire concerné.)	Très grand. (Tous les locataires concernés.)

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Gestion et effort des performances des requêtes	Performances de requêtes gérables.	Performances de requêtes gérables.	Performances de requêtes gérables.	Des efforts importants sont probablement nécessaires pour maintenir les performances des requêtes. (Au fil du temps, les requêtes peuvent s'exécuter plus lentement en raison de l'augmentation de la taille des tables. Vous pouvez utiliser le partitionnement des tables et le partitionnement des bases de données pour maintenir les performances.)

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Impact sur les ressources entre locataires	Pas d'impact (Aucun partage de ressources entre les locataires.)	Impact modéré (Les locataires partagent des ressources communes telles que le processeur et la mémoire de l'instance.)	Impact modéré (Les locataires partagent des ressources communes telles que le processeur et la mémoire de l'instance.)	Impact (Les locataires s'influencent mutuellement en termes de ressources, de conflits de verrouillage, etc.)
Réglage au niveau du locataire (par exemple, création d'index supplémentaires par locataire ou modification des paramètres de base de données pour un locataire particulier)	Possible	Plutôt possible (Des modifications au niveau du schéma peuvent être apportées pour chaque locataire, mais les paramètres de base de données sont globaux pour tous les locataires.)	Plutôt possible (Des modifications au niveau du schéma peuvent être apportées pour chaque locataire, mais les paramètres de base de données sont globaux pour tous les locataires.)	Pas possible (Les tables sont partagées par tous les locataires.)

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Rééquilibrer les efforts en faveur des locataires soucieux des performances	Minimale. (Pas besoin de rééquilibrer. Adaptez les ressources du serveur et des E/S pour gérer ce scénario.)	Modéré. (Utilisez la réplication logique oupg_dump pour exporter la base de données, mais les interruptions peuvent être longues en fonction de la taille des données. Vous pouvez utiliser la fonctionnalité de base de données transportable d'Amazon RDS for PostgreSQL afin de copier les bases de données entre les instances plus rapidement.)	Modéré mais impliquant probablement de longs temps d'arrêt. (Utilisez la réplication logique oupg_dump exportez le schéma, mais les interruptions peuvent être longues en fonction de la taille des données.)	Important, car tous les locataires partagent les mêmes tables. (Le partitionnement de la base de données nécessite de tout copier vers une autre instance et d'effectuer une étape supplémentaire pour nettoyer les données des locataires.)  Nécessite très probablement une modification de la logique de l'application.

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Indisponibilité de la base de données des mises à niveau de version	Temps d'arrêt standard. (Cela dépend de la taille du catalogue du système PostgreSQL.)	Des temps d'arrêt plus longs sont probables . (En fonction de la taille du catalogue du système, le délai peut varier. (Les tables du catalogue du système PostgreSQL sont également dupliquées dans les bases de données)	Des temps d'arrêt plus longs sont probables . (En fonction de la taille du catalogue du système PostgreSQL, le délai peut varier.)	Temps d'arrêt standard. (Cela dépend de la taille du catalogue du système PostgreSQL.)
Frais d'administration (par exemple, pour l'analyse des journaux de base de données ou la surveillance des tâches de sauvegarde)	Effort	Niveau d'effort	Niveau d'effort	Niveau d'effort

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Disponibilité au niveau du locataire	Au plus haut (Chaque locataire échoue et se rétablit indépendamment.)	Étendue d'impact plus étendue. (Tous les locataires échouent et se rétablissent ensemble en cas de problèmes matériels ou de ressources.)	Étendue d'impact plus étendue. (Tous les locataires échouent et se rétablissent ensemble en cas de problèmes matériels ou de ressources.)	Étendue d'impact plus étendue. (Tous les locataires échouent et se rétablissent ensemble en cas de problèmes matériels ou de ressources.)
Effort de sauvegarde et de restauration au niveau du locataire	Alertes d'effort (Chaque locataire peut être sauvegardé et restauré indépendamment.)	Effort (Utilisez l'exportation et l'importation logiques pour chaque locataire. Un peu de codage et d'automatisation sont nécessaires.)	Effort (Utilisez l'exportation et l'importation logiques pour chaque locataire. Un peu de codage et d'automatisation sont nécessaires.)	Alertes d'effort (Tous les locataires partagent les mêmes tables.)
Effort de point-in-time rétablissement au niveau des locataires	Niveau d'effort (Utilisez la restauration instantanée en utilisant des instantanés, ou utilisez le retour en arrière dans Amazon Aurora.)	Effort (Utilisez la restauration instantanée, suivie de l'exportation/importation. Cependant, cette opération sera lente.)	Effort (Utilisez la restauration instantanée, suivie de l'exportation/importation. Cependant, cette opération sera lente.)	Effort et complexité importants.

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Nom de schéma uniforme	Même nom de schéma pour chaque locataire.	Même nom de schéma pour chaque locataire.	Schéma différent pour chaque locataire.	Schéma commun
Personnalisation par locataire (par exemple, colonnes de tableau supplémentaires pour un locataire spécifique)	Possible	Possible	Possible	Complicé (car tous les locataires partagent les mêmes tables).

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Efficacité de la gestion des catalogues au niveau de la couche de mappage relationnel objet (ORM) (par exemple, Ruby)	Efficace (car la connexion client est spécifique à un locataire).	Efficace (car la connexion client est spécifique à une base de données).	Modérément efficace. (En fonction de l'ORM utilisé, du modèle de sécurité utilisateur/rôle et de la <code>search_path</code> configuration, le client met parfois en cache les métadonnées de tous les locataires, ce qui entraîne une utilisation élevée de la mémoire de connexion à la base de données.)	Efficace (car tous les locataires partagent les mêmes tables).

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Effort de reporting consolidé des locataires	Alertes d'effort (Vous devez utiliser des wrappers de données étrangers [FDW] pour consolider les données de tous les locataires ou extraire, transformer et charger [ETL] dans une autre base de données de reporting.)	Alertes d'effort (Vous devez utiliser des FDW pour consolider les données de tous les locataires ou des ETL dans une autre base de données de reporting.)	Effort (Vous pouvez agréger les données dans tous les schémas en utilisant des unions.)	Niveau d'effort (Toutes les données des locataires se trouvent dans les mêmes tableaux, ce qui simplifie la création de rapports.)
Instance en lecture seule spécifique au locataire pour les rapports (par exemple, sur la base d'un abonnement)	Alertes d'effort (Créez un réplica en lecture.)	Effort (Vous pouvez utiliser la réplication logique ou le AWS Database Migration Service [AWS DMS] pour effectuer la configuration.)	Effort (Vous pouvez utiliser la réplication logique ou AWS DMS pour configurer.)	Compliqué (car tous les locataires partagent les mêmes tables).

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Isolation des données	Meilleur	Mieux (Vous pouvez gérer les autorisations au niveau de la base de données à l'aide de rôles PostgreSQL.)	Mieux (Vous pouvez gérer les autorisations au niveau du schéma à l'aide de rôles PostgreSQL.)	Pire. (Comme tous les locataires partagent les mêmes tables, vous devez implémenter des fonctionnalités telles que la sécurité au niveau des lignes [RLS] pour isoler les locataires.)
Clé de chiffrement de stockage spécifique au locataire	Possible (Chaque cluster PostgreSQL peut avoir sa propre clé AWS Key Management Service [AWS KMS] pour le chiffrement du stockage.)	Pas possible (Tous les locataires partagent la même clé KMS pour le chiffrement du stockage.)	Pas possible (Tous les locataires partagent la même clé KMS pour le chiffrement du stockage.)	Pas possible (Tous les locataires partagent la même clé KMS pour le chiffrement du stockage.)

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Utilisation deAWS Identity and Access Management (IAM) pour l'authentification de la base de données pour chaque locataire	Possible	Possible	Possible (en ayant des utilisateurs PostgreSQL distincts pour chaque schéma).	Impossible (car les tables sont partagées par tous les locataires).
Coût d'infrastructure	Le plus élevé (car rien n'est partagé).	Modéré.	Modéré.	Au plus bas
Duplication des données et utilisation du stockage	Le total le plus élevé parmi tous les locataires. (Les tables du catalogue du système PostgreSQL et les données statiques et communes de l'application sont dupliquées sur tous les locataires.)	Le total le plus élevé parmi tous les locataires. (Les tables du catalogue du système PostgreSQL et les données statiques et communes de l'application sont dupliquées sur tous les locataires.)	Modéré. (Les données statiques et communes de l'application peuvent se trouver dans un schéma commun et être accessibles par d'autres locataires.)	Minimale. (Aucune duplication des données. Les données statiques et communes de l'application peuvent se trouver dans le même schéma.)

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Surveillance centrée sur le locataire (découvrez rapidement quel locataire est à l'origine des problèmes)	Alertes d'effort (Comme chaque locataire est surveillé séparément, il est facile de vérifier l'activité d'un locataire spécifique.)	Effort (Comme tous les locataires partagent la même ressource physique, vous devez appliquer un filtrage supplémentaire pour vérifier l'activité d'un locataire spécifique.)	Effort (Comme tous les locataires partagent la même ressource physique, vous devez appliquer un filtrage supplémentaire pour vérifier l'activité d'un locataire spécifique.)	Alertes d'effort (Comme tous les locataires partagent toutes les ressources, y compris les tables, vous devez utiliser la capture de la variable bind pour vérifier à quel locataire appartient une requête SQL spécifique.)
Gestion centralisée et surveillance de l'état de santé et des activités	Effort important (pour mettre en place une surveillance centrale et un centre de commande central).	Effort modéré (car tous les locataires partagent la même instance).	Effort modéré (car tous les locataires partagent la même instance).	Effort minimal (car tous les locataires partagent les mêmes ressources, y compris le schéma).

	Silo	Bridge avec bases de données distinctes	Pont avec schémas séparés	Piscine
Chances de concordance entre l'identifiant de l'objet (OID) et l'identifiant de la transaction (XID)	Minimale.	Haut (Parce que OID, XID est un compteur PostgreSQL unique à l'échelle du cluster et il peut y avoir des problèmes pour passer l'aspirateur efficacement entre les bases de données physiques).	Modéré. (Parce que OID, XID est un compteur PostgreSQL unique à l'échelle du cluster).	Haut (Par exemple, une seule table peut atteindre la limite TOAST OID de 4 milliards, selon le nombre de out-of-line colonnes.)

# Recommandations relatives à la sécurité au niveau des lignes

La sécurité au niveau des lignes (RLS) est requise pour maintenir l'isolation des données des locataires dans un modèle groupé avec PostgreSQL. RLS centralise l'application des politiques d'isolation au niveau de la base de données et allège la charge du maintien de cette isolation pour les développeurs de logiciels. La méthode la plus courante pour implémenter le RLS consiste à activer cette fonctionnalité dans le SGBD PostgreSQL. Le RLS implique le filtrage de l'accès aux lignes de données en fonction d'une valeur dans une colonne spécifiée. Vous pouvez utiliser deux méthodes pour filtrer l'accès aux données :

- Une colonne de données spécifiée dans un tableau est comparée à la valeur de l'utilisateur PostgreSQL actuel. Les valeurs de la colonne qui sont équivalentes à l'utilisateur PostgreSQL connecté sont accessibles à cet utilisateur.
- Une colonne de données spécifiée dans un tableau est comparée à la valeur d'une variable d'exécution définie par l'application. Les valeurs de la colonne qui sont équivalentes à la variable d'exécution sont accessibles pendant cette session.

La deuxième option est préférée, car la première nécessite la création d'un nouvel utilisateur PostgreSQL pour chaque locataire. Au lieu de cela, une application SaaS qui utilise PostgreSQL doit être chargée de définir un contexte spécifique au locataire lors de l'exécution lorsqu'elle interroge PostgreSQL. Cela aura pour effet de faire appliquer le RLS. Vous pouvez également activer le RLS sur une table-by-table base. En tant que bonne pratique, vous devez activer le RLS sur toutes les tables contenant des données sur les locataires.

L'exemple suivant crée deux tables et active le RLS. Cet exemple compare une colonne de données à la valeur de la variable d'exécution `app.current_tenant`.

```
-- Create a table for our tenants with indexes on the primary key and the tenant's name
CREATE TABLE tenant (
  tenant_id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,
  name VARCHAR(255) UNIQUE,
  status VARCHAR(64) CHECK (status IN ('active', 'suspended', 'disabled')),
  tier VARCHAR(64) CHECK (tier IN ('gold', 'silver', 'bronze'))
);
```

```
-- Create a table for users of a tenant
CREATE TABLE tenant_user (
  user_id UUID DEFAULT uuid_generate_v4() PRIMARY KEY,
  tenant_id UUID NOT NULL REFERENCES tenant (tenant_id) ON DELETE RESTRICT,
  email VARCHAR(255) NOT NULL UNIQUE,
  given_name VARCHAR(255) NOT NULL CHECK (given_name <> ''),
  family_name VARCHAR(255) NOT NULL CHECK (family_name <> '')
);

-- Turn on RLS
ALTER TABLE tenant ENABLE ROW LEVEL SECURITY;

-- Restrict read and write actions so tenants can only see their rows
-- Cast the UUID value in tenant_id to match the type current_setting
-- This policy implies a WITH CHECK that matches the USING clause
CREATE POLICY tenant_isolation_policy ON tenant
USING (tenant_id = current_setting('app.current_tenant')::UUID);

-- And do the same for the tenant users
ALTER TABLE tenant_user ENABLE ROW LEVEL SECURITY;

CREATE POLICY tenant_user_isolation_policy ON tenant_user
USING (tenant_id = current_setting('app.current_tenant')::UUID);
```

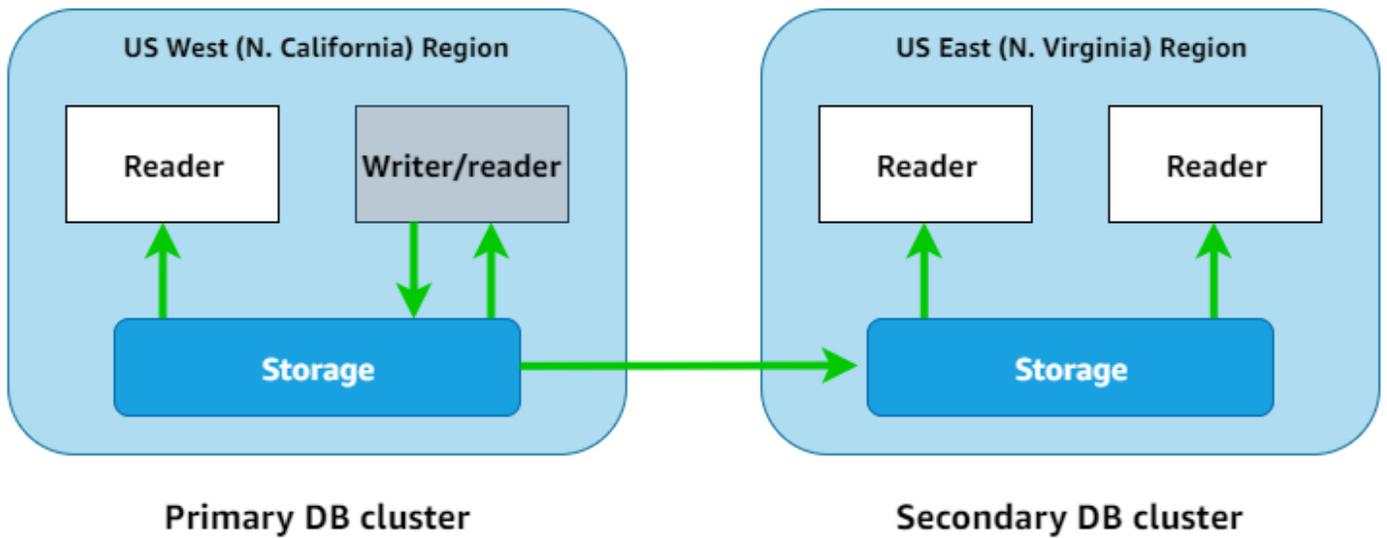
Pour de plus amples informations, veuillez consulter le billet [de blog Isolement PostgreSQL Row Level Security](#). L'équipe deAWS SaaS Factory propose également [quelques exemples GitHub](#) pour vous aider à mettre en œuvre le RLS.

## Disponibilité de PostgreSQL pour le modèle de pool

De par leur nature, les modèles de pool ne possèdent qu'une seule instance de PostgreSQL. Il est donc crucial de concevoir votre application pour une haute disponibilité. En cas de panne ou de panne d'une base de données groupée, votre application se dégrade ou devient inaccessible pour tous vos locataires.

Les instances de base de données Amazon RDS for PostgreSQL peuvent être rendues redondantes dans deux zones de disponibilité en activant la fonctionnalité de haute disponibilité. Pour plus d'informations, consultez la section [Haute disponibilité \(multi-AZ\) pour Amazon RDS](#) dans la documentation Amazon RDS. Pour le basculement entre régions, vous pouvez créer une réplique en lecture dans une autre AWS région. (Cette réplique en lecture doit être promue dans le cadre d'un processus de basculement.) En outre, vous pouvez répliquer les sauvegardes répliquées dans différentes AWS régions à des fins de restauration. Pour plus d'informations, consultez la section [Réplication de sauvegardes automatisées vers une autre AWS région](#) dans la documentation Amazon RDS.

La compatibilité avec Aurora PostgreSQL sauvegarde automatiquement les données de manière à résister à la défaillance de plusieurs zones de disponibilité. (Voir [Haute disponibilité pour Amazon Aurora](#) dans la documentation Aurora.) Pour renforcer la résilience d'Aurora et accélérer la restauration, vous pouvez créer des répliques de lecture Aurora dans d'autres zones de disponibilité. Vous pouvez utiliser les bases de données globales Aurora pour répliquer les données dans cinq AWS régions supplémentaires à des fins de restauration entre régions et de basculement automatique. (Voir [Utilisation des bases de données globales Amazon Aurora](#) dans la documentation Aurora.) En outre, vous pouvez activer le [transfert d'écriture](#) avec les bases de données globales Aurora pour garantir une haute disponibilité sur plusieurs bases de données Régions AWS.



Que vous utilisiez Amazon RDS for PostgreSQL ou que vous utilisiez une solution compatible avec Aurora PostgreSQL, nous vous recommandons de mettre en œuvre des fonctionnalités de haute disponibilité afin d'atténuer l'impact des pannes sur toutes les applications SaaS mutualisées qui utilisent un modèle de pool.

## Bonnes pratiques

Cette section répertorie certains des principaux points à retenir de ce guide. Pour des discussions détaillées sur chaque point, suivez les liens vers les sections correspondantes.

### Comparez AWS les options pour PostgreSQL géré

AWS propose deux méthodes principales pour exécuter PostgreSQL dans un environnement géré. (Dans ce contexte, géré signifie que l'infrastructure PostgreSQL et le SGBD sont partiellement ou totalement pris en charge par un AWS service.) Les options de gestion de PostgreSQL activées AWS présentent l'avantage d'automatiser les sauvegardes, le basculement, l'optimisation et une certaine administration de PostgreSQL. En tant qu'options gérées, AWS propose Amazon Aurora compatible avec PostgreSQL et Amazon Relational Database Service (Amazon RDS) pour PostgreSQL. Vous pouvez sélectionner le meilleur choix parmi ces deux modèles en analysant votre cas d'utilisation de PostgreSQL. Pour de plus amples informations, [veuillez consulter la section Aurora](#) ce guide.

### Sélectionnez un modèle de partitionnement SaaS multi-locataires

Vous pouvez choisir parmi trois modèles de partitionnement SaaS applicables à PostgreSQL : silo, bridge et pool. Chaque modèle présente des avantages et des inconvénients, et vous devez choisir le modèle le plus optimal en fonction de votre cas d'utilisation. Amazon RDS for PostgreSQL et compatible avec Aurora PostgreSQL prennent en charge les trois modèles. Le choix d'un modèle est essentiel pour maintenir l'isolation des données des locataires dans vos applications SaaS. Pour une discussion détaillée de ces modèles, consultez la section [Modèles de partitionnement SaaS multi-locataires pour PostgreSQL](#) dans ce guide.

### Utiliser la sécurité au niveau des lignes pour les modèles de partitionnement SaaS de pool

La sécurité au niveau des lignes (RLS) est requise pour maintenir l'isolation des données des locataires dans un modèle de pool avec PostgreSQL. Cela est dû au fait qu'il n'existe aucune séparation logique entre l'infrastructure, les bases de données PostgreSQL ou les schémas par locataire dans un modèle de pool. RLS centralise l'application des politiques d'isolation au niveau de la base de données et allège la charge du maintien de cette isolation pour les développeurs de logiciels. Vous pouvez utiliser le protocole RLS pour limiter les opérations de base de données à un

---

client spécifique. Pour obtenir plus d'informations et un exemple, [reportez-vous à la section](#) de ce guide.

## FAQ

Cette section fournit des réponses aux questions fréquemment posées concernant l'implémentation de PostgreSQL géré dans des applications SaaS multi-locataires.

### Quelles sont les options de gestion de PostgreSQL AWS proposées ?

AWS propose une solution [compatible avec Amazon Aurora PostgreSQL](#) et [Amazon Relational Database Service \(Amazon RDS\) pour PostgreSQL](#). AWS dispose également d'un [vaste catalogue d'offres de bases de données gérées](#).

### Quel service est optimal pour les applications SaaS ?

Vous pouvez utiliser à la fois des applications compatibles avec Aurora PostgreSQL et Amazon RDS for PostgreSQL pour les applications SaaS et tous les modèles de partitionnement SaaS décrits dans ce guide. Ces deux services présentent des différences en termes d'évolutivité, de reprise après incident, de basculement, d'options de stockage, de haute disponibilité, de reprise après sinistre, de sauvegarde et de classes d'instances disponibles pour chaque option. Le choix optimal dépendra de votre cas d'utilisation spécifique. Utilisez la [matrice de décision](#) présentée dans ce guide pour choisir l'option la mieux adaptée à votre cas d'utilisation.

### Quelles exigences uniques dois-je prendre en compte si je décide d'utiliser une base de données PostgreSQL avec une application SaaS multi-tenant ?

Comme pour tout magasin de données utilisé avec une application SaaS, la considération la plus importante est la méthode de maintien de l'isolation des données des locataires. Comme indiqué dans ce guide, il existe plusieurs manières d'isoler les données des locataires grâce aux offres PostgreSQL AWS gérées. En outre, vous devez envisager l'isolation des performances par locataire pour toutes les implémentations de PostgreSQL.

## Quels modèles puis-je utiliser pour maintenir l'isolation des données des locataires avec PostgreSQL ?

Vous pouvez utiliser les modèles de silo, de pont et de pool comme stratégies de partitionnement SaaS afin de maintenir l'isolation des données des locataires. Pour en savoir plus sur ces modèles et sur la manière dont ils peuvent être appliqués à PostgreSQL, consultez la section [Modèles de partitionnement SaaS multi-locataires pour PostgreSQL](#) dans ce guide.

## Comment maintenir l'isolation des données des locataires avec une base de données PostgreSQL unique partagée entre plusieurs locataires ?

PostgreSQL prend en charge une fonctionnalité de sécurité au niveau des lignes (RLS) que vous pouvez utiliser pour renforcer l'isolation des données des locataires dans une seule base de données ou instance PostgreSQL. En outre, vous pouvez mettre en service des bases de données PostgreSQL distinctes par locataire dans une seule instance, ou créer des schémas par locataire pour atteindre cet objectif. Pour connaître les avantages et les inconvénients de ces approches, consultez la section [Recommandations de sécurité au niveau](#) des lignes de ce guide.

## Étapes suivantes

AWS propose deux options pour faire fonctionner PostgreSQL géré : Aurora PostgreSQL compatible et Amazon RDS for PostgreSQL. Nous vous recommandons d'évaluer les deux services et de choisir l'option qui convient le mieux à votre cas d'utilisation spécifique pour vos applications SaaS multi-locataires. La conformité à un modèle de partitionnement SaaS peut garantir qu'une application SaaS qui utilise PostgreSQL respecte strictement les meilleures pratiques en matière de gestion de la location. Les modèles SaaS de cloisonnement de silo, de pont et de pool prennent en charge de nombreux cas d'utilisation du SaaS. Ces modèles offrent divers avantages en termes de facteurs tels que l'isolation des performances, les frais opérationnels et la sécurité des locataires.

Prochaines étapes :

- [Évaluez la compatibilité avec Aurora PostgreSQL et Amazon RDS for PostgreSQL](#), et choisissez la meilleure option pour votre application SaaS.
- [Sélectionnez un modèle de partitionnement SaaS](#) qui répond aux exigences de votre application : silo, pont ou pool.
- Implémentez PostgreSQL conformément au modèle de partitionnement SaaS que vous avez sélectionné.

# Ressources

## Références

- [Stratégies de stockage SaaS : création d'un modèle de stockage mutualisé sur AWS](#) (AWSlivre blanc)
- [Reprise après sinistre entre régions à l'aide de la base de données globale Amazon Aurora pour Amazon Aurora PostgreSQL](#) (article de blogAWS)
- [Isolation des données mutualisées avec PostgreSQL Row Level Security](#) (article de blog) AWS
- [Utilisation d'Amazon Aurora PostgreSQL](#) (documentation Aurora)
- [PostgreSQL sur Amazon RDS](#) (documentation Amazon RDS)

## Partenaires

- [Amazon Aurora pour les partenaires PostgreSQL](#)
- [Partenaires Amazon RDS pour PostgreSQL](#)

## 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="#">Mettre à jour</a>	Mises à jour pour refléter la disponibilité du transfert d'écriture dans Aurora.	29 avril 2024
<a href="#">Mettre à jour</a>	Mise à jour du <a href="#">tableau de comparaison entre Amazon RDS et Aurora</a> .	21 octobre 2022
<a href="#">=</a>	Publication initiale	30 septembre 2021

# 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 Amazon Aurora Édition compatible avec 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 in the Cloud. AWS
- **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. Ce scénario de migration est spécifique à VMware Cloud on AWS, qui prend en charge la compatibilité des machines virtuelles (VM) et la portabilité de la charge de travail entre votre environnement sur site et AWS. Vous pouvez utiliser les technologies VMware Cloud Foundation à partir de vos centres de données sur site lorsque vous migrez votre infrastructure vers VMware Cloud on AWS. Exemple : déplacez l'hyperviseur hébergeant votre base de données Oracle vers VMware Cloud on. AWS

- **Retenir** : conservez les applications dans votre environnement source. Il peut s'agir d'applications nécessitant une refactorisation majeure, que vous souhaitez retarder, et d'applications existantes que vous souhaitez retenir, car rien ne justifie leur migration sur le plan commercial.
- **Retirer** : mettez hors service ou supprimez les applications dont vous n'avez plus besoin dans votre environnement source.

## A

### ABAC

Voir contrôle [d'accès basé sur les attributs](#).

services abstraits

Consultez la section [Services gérés](#).

### ACIDE

Voir [atomicité, consistance, isolation, durabilité](#).

migration active-active

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

migration active-passive

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

fonction d'agrégation

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

## AI

Voir [intelligence artificielle](#).

## AIOps

Voir les [opérations d'intelligence artificielle](#).

## anonymisation

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

## anti-motif

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

## contrôle des applications

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

## portefeuille d'applications

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

## intelligence artificielle (IA)

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

## opérations d'intelligence artificielle (AIOps)

Processus consistant à utiliser des techniques de machine learning pour résoudre les problèmes opérationnels, réduire les incidents opérationnels et les interventions humaines, mais aussi améliorer la qualité du service. Pour plus d'informations sur la façon dont les AIOps sont utilisées dans la stratégie de migration AWS, veuillez consulter le [guide d'intégration des opérations](#).

## chiffrement asymétrique

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

## atomicité, cohérence, isolement, durabilité (ACID)

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

## contrôle d'accès par attributs (ABAC)

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

## source de données faisant autorité

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

## Zone de disponibilité

Un emplacement distinct au sein d'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.

## CCoE

Voir [le Centre d'excellence du cloud](#).

## CDC

Consultez la section [Capture des données de modification](#).

## capture des données de modification (CDC)

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

## ingénierie du chaos

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

## CI/CD

Découvrez [l'intégration continue et la livraison continue](#).

## classification

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

## chiffrement côté client

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

## Centre d'excellence 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 [articles du CCoE](#) sur le blog AWS Cloud Enterprise Strategy.

## 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 le 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 mettre à l'échelle l'adoption du cloud (par exemple, en créant une zone de destination, en définissant un CCoE ou en établissant 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 on the AWS Cloud Enterprise Strategy](#) blog. 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 AWS CodeCommit. 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, AWS

Panorama propose des appareils qui ajoutent des CV aux réseaux de caméras locaux, et Amazon SageMaker 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 source, de génération, de test, intermédiaire 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

Consultez la section [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](#).

### 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.

### chiffrement

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

### clé de chiffrement

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

### endianisme

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

### point de terminaison

Voir [point de terminaison de service](#).

### service de point de terminaison

Service que vous pouvez héberger sur un cloud privé virtuel (VPC) pour le partager avec d'autres utilisateurs. Vous pouvez créer un service de point de terminaison avec AWS PrivateLink et accorder des autorisations à d'autres Comptes AWS ou à AWS Identity and Access Management (IAM) principaux. Ces comptes ou principaux peuvent se connecter à votre service de point de

terminaison de manière privée en créant des points de terminaison d'un VPC d'interface. Pour plus d'informations, veuillez consulter [Création d'un service de point de terminaison](#) dans la documentation Amazon Virtual Private Cloud (Amazon VPC).

## planification des ressources d'entreprise (ERP)

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

## chiffrement d'enveloppe

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

## 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 pipeline CI/CD, 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 épépées 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 [la 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.

## 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 via 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.

## G

### 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.

### 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 d'organisation (UO). 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 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 de l'historien

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.

#### migration de base de données homogène

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

#### données chaudes

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

#### correctif

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

#### période de soins intensifs

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

I

IaC

Considérez [l'infrastructure comme un code](#).

politique basée sur l'identité

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

application inactive

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

IIoT

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 de référence de sécuritéAWS](#) recommande de configurer votre compte réseau avec des VPC entrants, sortants et d'inspection afin de protéger l'interface bidirectionnelle entre votre application et 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

I

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, veuillez consulter [Building an industrial Internet of Things \(IIoT\) digital transformation strategy](#).

## VPC d'inspection

Dans une architecture AWS multi-comptes, un VPC centralisé qui gère les inspections du trafic réseau entre les VPC (identiques ou Régions AWS différents), Internet et les réseaux sur site. L'[architecture de référence de sécurité AWS](#) recommande de configurer votre compte réseau avec des VPC entrants, sortants et d'inspection afin de protéger l'interface bidirectionnelle entre votre application et 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, veuillez consulter [Machine learning model interpretability with 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](#).

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](#).

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 [la 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 qui AWS gère 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 au fur et à mesure de son fonctionnement. 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/abonnement, pour les appareils IoT aux ressources limitées.](#)

## microservice

Petit service indépendant qui communique via des API bien définies 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 à l'aide d'API légères. 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 les opérations, les analystes commerciaux et les propriétaires, les ingénieurs de migration, les développeurs et les DevOps 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)

Un 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

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, veuillez consulter [Evaluating modernization readiness for applications in the AWS Cloud](#).

## applications monolithiques (monolithes)

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

## MPA

Voir [Évaluation du portefeuille de migration](#).

## MQTT

Voir [Message Queuing Telemetry Transport](#).

## classification multi-classes

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

## infrastructure mutable

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

## O

### OAC

Voir [Contrôle d'accès à l'origine](#).

### OAI

Voir [l'identité d'accès à l'origine](#).

### OCM

Voir [gestion du changement organisationnel](#).

### migration hors ligne

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

### OI

Consultez la section [Intégration des opérations](#).

### OLA

Voir l'accord [au niveau opérationnel](#).

### migration en ligne

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

### OPC-UA

Voir [Open Process Communications - Architecture unifiée](#).

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

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

## accord au niveau opérationnel (OLA)

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

## examen de l'état de préparation opérationnelle (ORR)

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

## technologie opérationnelle (OT)

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

## intégration des opérations (OI)

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

## journal de suivi d'organisation

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

## gestion du changement organisationnel (OCM)

Cadre pour gérer les transformations métier majeures et perturbatrices du point de vue des personnes, de la culture et du leadership. L'OCM aide les organisations à se préparer et à effectuer la transition vers de nouveaux systèmes et de nouvelles politiques en accélérant l'adoption des changements, en abordant les problèmes de transition et en favorisant des changements culturels et organisationnels. Dans la stratégie de AWS migration, ce cadre est appelé accélération du personnel, en raison de la rapidité du changement requise dans les projets d'adoption du cloud. Pour plus d'informations, veuillez consulter le [guide OCM](#).

## contrôle d'accès d'origine (OAC)

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

## identité d'accès d'origine (OAI)

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

## OU

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 de référence de sécuritéAWS](#) recommande de configurer votre compte réseau avec des VPC entrants, sortants et d'inspection afin de protéger l'interface bidirectionnelle entre votre application et 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](#).

## politique

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

## persistance polyglotte

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

## évaluation du portefeuille

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

## predicate

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

## prédicat pushdown

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

## contrôle préventif

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

## principal

Entité capable d'effectuer AWS des actions et d'accéder à des 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 de l'ingénierie des systèmes qui prend en compte la confidentialité tout au long du processus d'ingénierie.

## zones hébergées privées

Conteneur qui contient des informations concernant la façon dont vous souhaitez qu'Amazon Route 53 réponde aux requêtes DNS pour un domaine et ses sous-domaines dans un ou plusieurs VPC. 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.

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.

publier/souscrire (pub/sub)

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

## Q

plan de requête

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

régression du plan de requêtes

Le cas où un optimiseur de service de base de données choisit un plan moins optimal qu'avant une modification donnée de l'environnement de base de données. Cela peut être dû à des

changements en termes de statistiques, de contraintes, de paramètres d'environnement, de liaisons de paramètres de requêtes et de mises à jour du moteur de base de données.

## R

### Matrice RACI

Voir [responsable, responsable, consulté, informé \(RACI\)](#).

### 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. Cela permet de déterminer ce qui est considéré comme une perte de données acceptable entre le dernier point de restauration 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. Chacune Région AWS est isolée et indépendante 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 la section [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](#).

## 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 d' 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, consultez la section [Secret](#) dans la documentation de Secrets Manager.

## contrôle de sécurité

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

## renforcement de la sécurité

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

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

Outils et services qui associent les systèmes de gestion des informations de sécurité (SIM) et de gestion des événements de sécurité (SEM). Un système SIEM collecte, surveille et analyse les

données provenant de serveurs, de réseaux, d'appareils et d'autres sources afin de détecter les menaces et les failles de sécurité, mais aussi de générer des alertes.

#### automatisation des réponses de sécurité

Action prédéfinie et programmée conçue pour répondre automatiquement à un événement de sécurité ou y remédier. Ces automatisations servent de contrôles de sécurité [détectifs ou réactifs](#) qui vous aident à mettre en œuvre les meilleures pratiques AWS de sécurité. Parmi les actions de réponse automatique, citons la modification d'un groupe de sécurité VPC, l'application de correctifs à une 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 propose un contrôle centralisé des autorisations pour tous les comptes d'une organisation dans AWS Organizations. Les SCP définissent des barrières de protection ou des limites aux actions qu'un administrateur peut déléguer à des utilisateurs ou à des rôles. Vous pouvez utiliser les SCP comme listes d'autorisation ou de refus, pour indiquer les services ou les actions autorisés ou interdits. Pour plus d'informations, consultez la section [Politiques de contrôle des services](#) dans la AWS Organizations documentation.

#### point de terminaison du service

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

#### contrat de niveau de service (SLA)

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

#### indicateur de niveau de service (SLI)

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

#### objectif de niveau de service (SLO)

Mesure cible qui représente l'état d'un service, tel que mesuré par un indicateur de [niveau de service](#).

## modèle de responsabilité partagée

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

## SIEM

Consultez les [informations de sécurité et le système de gestion des événements](#).

## point de défaillance unique (SPOF)

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

## SLA

Voir le contrat [de niveau de service](#).

## SLI

Voir l'indicateur de [niveau de service](#).

## SLO

Voir l'objectif de [niveau de service](#).

## split-and-seed modèle

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

## SPOF

Voir [point de défaillance unique](#).

## schéma en étoile

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

## modèle de figuier étrangleur

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

## sous-réseau

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

## contrôle de supervision et acquisition de données (SCADA)

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

## chiffrement symétrique

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

## tests synthétiques

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

# T

## balises

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

## variable cible

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

## liste de tâches

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

## environnement de test

Voir [environnement](#).

## entraînement

Pour fournir des données à partir desquelles votre modèle de ML peut apprendre. Les données d'entraînement doivent contenir la bonne réponse. L'algorithme d'apprentissage identifie des modèles dans les données d'entraînement, qui mettent en correspondance les attributs des données d'entrée avec la cible (la réponse que vous souhaitez prédire). Il fournit un modèle de ML qui capture ces modèles. Vous pouvez alors utiliser le modèle de ML pour obtenir des prédictions sur de nouvelles données pour lesquelles vous ne connaissez pas la cible.

## passerelle de transit

Hub de transit de réseau que vous pouvez utiliser pour relier vos VPC et vos réseaux 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

Connexion entre deux VPC qui vous permet d'acheminer le trafic à l'aide d'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

Consultez le [cadre de qualification des charges de travail AWS](#).

## 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 de type « 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.

### 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.