



Utilisez CCM et QPM pour optimiser les performances de restauration et les plans d'exécution dans Amazon Aurora PostgreSQL

AWSConseils prescriptifs



AWSConseils prescriptifs: Utilisez CCM et QPM pour optimiser les performances de restauration et les plans d'exécution dans Amazon Aurora PostgreSQL

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

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

Table of Contents

Introduction	1
Public visé	2
Des résultats de bureau ciblés	3
Gestion des caches de clusters	4
Fonctionnement de CCM	4
Limites	6
Cas d'utilisation pour CCM	7
Gestion de plans de requêtes	8
Fonctionnement de QPM	9
Limites	10
Cas d'utilisation pour QPM	10
Ressources	11
Documentation AWS	11
AWSarticles de blog	11
AWSateliers	11
Historique du document	12
.....	xiii

Utilisez CCM et QPM pour optimiser les performances de restauration et les plans d'exécution dans Amazon Aurora PostgreSQL

Raunak Rishabh, Rohit Kapoor et Sujitha Sasikumaran, Amazon Web Services (AWS)

Janvier 2023 ([historique du document](#))

À mesure que les entreprises se développent, elles utilisent de plus en plus de données pour prendre des décisions critiques. Face à l'augmentation des volumes de données, il est important d'optimiser les performances de la base de données et de la stabiliser lors des modifications du système. Les charges de travail hautement transactionnelles, telles que celles impliquant des transactions financières ou des commandes clients, nécessitent des performances stables, cohérentes et rapides, car de mauvaises performances peuvent affecter la satisfaction des clients et les revenus de l'entreprise. Pour les bases de données qui gèrent ces charges de travail hautement transactionnelles, telles que les instances de base de données Amazon Aurora en édition compatible avec PostgreSQL, il est essentiel que vous compreniez et mettiez en œuvre les fonctionnalités d'optimisation des performances disponibles.

[Compatible avec Amazon Aurora PostgreSQL](#), il s'agit d'un moteur de base de données relationnelle entièrement géré qui vous aide à configurer, exploiter et dimensionner des déploiements PostgreSQL. Il s'agit d'un moteur de base de données largement utilisé en raison de son architecture de stockage autonome et de ses fonctionnalités, qui vous aident à optimiser les performances dans des scénarios de charge de travail réels avec une charge de maintenance minimale.

Deux de ces fonctionnalités sont la [gestion du cache des clusters \(CCM\)](#) et la [gestion des plans de requêtes \(QPM\)](#). CCM vous permet de récupérer les performances des applications et des bases de données en cas de basculement, et QPM vous permet de gérer les plans d'exécution des requêtes générés par l'optimiseur pour vos applications SQL. Ces deux fonctionnalités peuvent contribuer à optimiser les performances des requêtes SQL en fournissant un meilleur contrôle sur la base de données. Ce guide est destiné à aider les responsables, les propriétaires de produits et les architectes de bases de données (DBA) à comprendre les avantages et les résultats commerciaux potentiels de la mise en œuvre de la CCM et de la QPM.

Public visé

Ce guide s'adresse aux acteurs commerciaux qui souhaitent comprendre les fonctionnalités disponibles pour optimiser les performances des instances de base de données compatibles avec Amazon Aurora PostgreSQL et comprendre les cas d'utilisation de ces fonctionnalités.

Des résultats de bureau ciblés

Vous pouvez utiliser ce guide pour obtenir les résultats de bureau suivants avec la gestion de cache du cluster (CCM) :

- En cas de basculement, procédez à une restauration rapide afin de maintenir des performances stables et optimales de la charge de travail.
- Réduisez les pertes commerciales causées par les performances médiocres des charges de travail après un basculement.
- Évitez les coûts d'E/S inutiles après un basculement.

Vous pouvez utiliser ce guide pour obtenir les résultats commerciaux suivants grâce à la gestion des plans de requêtes (QPM) :

- Optimisez la stabilité du plan en forçant l'optimiseur à opérer sa sélection parmi une poignée de plans approuvés. Cela empêche l'optimiseur de sélectionner un plan sous-optimal pour une instruction SQL donnée après des modifications du système ou de la base de données.
- Optimisez les plans de manière centralisée, puis distribuez-les à l'échelle mondiale.
- Déterminez les index non utilisés et évaluez les effets de l'ajout ou de la suppression d'un index.
- Reconnaissez automatiquement tous les nouveaux plans à coût minimum identifiés par l'optimiseur.
- Testez les nouvelles fonctionnalités de l'optimiseur avec un minimum de risques, car vous pouvez décider de n'accepter que les changements de plans qui optimisent les performances.

Gestion des caches de clusters

La mise en cache est l'une des fonctionnalités les plus importantes de toute base de données (DB) car elle permet de réduire les E/S du disque. Les données les plus fréquemment consultées sont stockées dans une zone mémoire appelée cache tampon. Lorsqu'une requête s'exécute fréquemment, elle extrait les données directement du cache plutôt que du disque. Cette solution est plus rapide et améliore l'évolutivité et les performances des applications. Vous configurez la taille du cache PostgreSQL à l'aide du `dshared_buffers` paramètre. Pour de plus amples informations, veuillez consulter [Memory](#) (documentation PostgreSQL).

Après un basculement, la [gestion du cache de cluster \(CCM\)](#) dans l'édition compatible avec Amazon Aurora PostgreSQL est conçue pour améliorer les performances de restauration des applications et des bases de données. Dans une situation de basculement classique sans CCM, vous pouvez constater une dégradation temporaire, mais conséquente, des performances. Cela est dû au fait que le cache des caches de mémoire tampon est vide lorsque l'instance de base de données de basculement démarre. Un cache vide est également appelé cache passif. L'instance de base de données doit lire depuis le disque, ce qui est plus lent que la lecture depuis le cache.

Lorsque vous implémentez CCM, vous choisissez une instance de base de données de lecteur préférée, et CCM synchronise en permanence sa mémoire cache avec celle de l'instance de base de données principale, ou d'écriture. En cas de basculement, l'instance de base de données de lecteur préférée est promue en tant que nouvelle instance de base de données de rédacteur. Comme il possède déjà une mémoire cache, connue sous le nom de cache chaud, cela minimise l'impact du basculement sur les performances de l'application.

Comment fonctionne la gestion du cache du cluster ?

Les instances de base de données de basculement se situent dans des zones de disponibilité différentes de celles de l'instance de base de données d'écriture principale. L'instance de base de données de lecteur préférée est la cible de basculement prioritaire, qui est spécifiée en lui affectant le niveau de priorité de niveau 0.

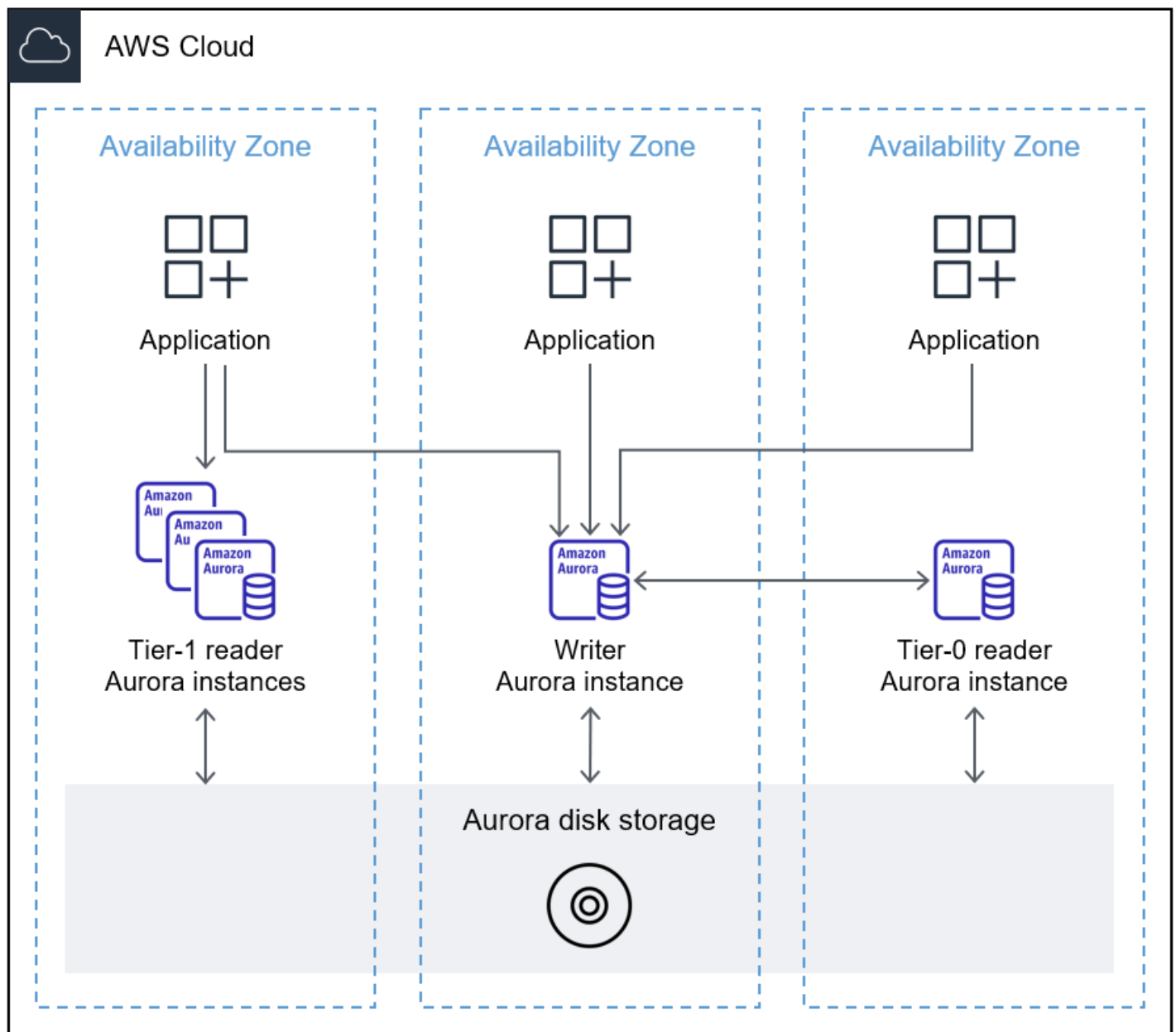
Note

La priorité du niveau de promotion est une valeur qui spécifie l'ordre dans lequel un lecteur Aurora est promu comme instance de base de données de l'enregistreur après un échec. Les

valeurs valides sont comprises dans la plage 0–15, 0 étant la priorité la plus élevée et 15 la priorité la plus faible. Pour plus d'informations sur le niveau de promotion, veuillez consulter [Tolérance aux pannes d'un cluster de base de données Aurora](#). La modification du niveau de promotion n'entraîne pas d'interruption de service.

CCM synchronise le cache de l'instance de base de données de rédacteur à l'instance de base de données de lecteur préférée. L'instance de base de données de lecteur envoie l'ensemble des adresses de mémoire tampon actuellement mises en cache à l'instance de base de données d'écriture sous la forme d'un filtre bloquant. Un filtre Bloom est une structure de données probabiliste économe en mémoire qui est utilisée pour vérifier si un élément fait partie d'un ensemble. L'utilisation d'un filtre Bloom empêche l'instance de base de données de lecteur d'envoyer à plusieurs reprises les mêmes adresses de mémoire tampon à l'instance de base de données d'écriture. Lorsque l'instance de base de données d'écriture reçoit le filtre Bloom, elle compare les blocs de son cache tampon et envoie les tampons fréquemment utilisés à l'instance de base de données de lecture. Par défaut, une mémoire tampon est considérée comme fréquemment utilisée si son nombre d'utilisations est supérieur à trois.

Le schéma suivant montre comment CCM synchronise le cache tampon de l'instance de base de données d'écriture avec l'instance de base de données de lecteur préférée.



Pour plus d'informations sur CCM, voir [Restauration rapide après basculement grâce à la gestion du cache de cluster pour Aurora PostgreSQL](#) (documentation Aurora) et [Introduction à la gestion du cache de cluster Aurora PostgreSQL](#) (article de AWS blog). Pour obtenir des instructions sur la configuration du CCM, consultez la section [Configuration de la gestion du cache du cluster](#) (documentation Aurora).

Limites

La fonction CCM présente les restrictions suivantes :

- L'instance de base de données de lecteur doit avoir le même type et la même taille de classe d'instance de base de données de base de données que l'instance de base de données de rédacteur, par exemple `r5.2xlarge` ou `db.r5.xlarge`.
- CCM n'est pas pris en charge pour les clusters de base de données Aurora PostgreSQL qui font partie des bases de données globales Aurora.

Cas d'utilisation pour la gestion des caches de clusters

Dans certains secteurs, tels que le commerce de détail, la banque et la finance, des retards de quelques millisecondes seulement peuvent entraîner des problèmes de performance des applications et entraîner une perte d'activité significative. Comme le CCM permet de restaurer les performances des applications et des bases de données en synchronisant en permanence le cache tampon de l'instance de base de données principale avec l'instance de sauvegarde préférée, il peut aider à prévenir les pertes des entreprises liées aux basculements.

Gestion de plans de requêtes

Les changements en termes de statistiques, de contraintes, de paramètres d'environnement, de liaisons de paramètres de requêtes et de mises à niveau du moteur de base de données PostgreSQL peuvent tous provoquer une régression de plan de requêtes. La régression de plan de requêtes se produit lorsque l'optimiseur choisit un plan moins optimal qu'avant une modification donnée de l'environnement de base de données.

Dans l'édition compatible avec Amazon Aurora PostgreSQL, la fonctionnalité de [gestion des plans de requêtes \(QPM\)](#) est conçue pour garantir l'adaptabilité et la stabilité du plan, quelles que soient les modifications de l'environnement de base de données susceptibles d'entraîner une régression du plan de requête. QPM permet de contrôler dans une certaine mesure l'optimiseur. QPM vous permet de gérer le plan d'exécution des requêtes généré par l'optimiseur pour vos requêtes SQL. Le plan d'exécution des requêtes oblige l'optimiseur à choisir parmi vos plans approuvés pour les requêtes critiques, afin d'optimiser leurs performances.

Les entreprises déploient généralement des applications et des bases de données dans le monde entier ou gèrent plusieurs environnements pour chaque base de données d'applications, tels que le développement, l'assurance qualité, le test et la production. La gestion des plans d'exécution des requêtes pour chaque base de données, dans chaque environnement et dans l'ensemble des Régions AWS peut s'avérer complexe et fastidieuse. QPM peut exporter et importer des plans gérés compatibles avec Amazon Aurora PostgreSQL d'une base de données à une autre. Cela vous permet de gérer le plan d'exécution des requêtes de manière centralisée et de déployer des bases de données dans le monde entier. Vous pouvez utiliser cette fonctionnalité pour étudier un ensemble de plans dans une base de données de préproduction, vérifier qu'ils fonctionnent correctement, puis les charger dans un environnement de production.

QPM offre également plusieurs autres avantages. Par exemple, vous pouvez utiliser QPM pour améliorer les plans d'exécution qui ne peuvent pas être modifiés dans les applications ou lorsqu'il est impossible d'ajouter des indices à la déclaration. QPM détecte également automatiquement les nouveaux plans à coût minimum découverts par l'optimiseur, afin que vous puissiez continuer à optimiser les coûts en plus des performances.

Nous vous recommandons d'activer QPM. Lorsque QPM est activé, l'optimiseur utilise le plan à coût minimum que vous avez approuvé. Cela permet d'éviter la régression et de réduire le temps nécessaire à la gestion et à la correction des plans sous-optimaux.

Il existe deux approches différentes pour utiliser la fonctionnalité QPM : proactive et réactive. L'approche proactive est conçue pour empêcher toute régression des performances, tandis que l'approche réactive est conçue pour détecter et réparer les régressions de performances une fois qu'elles se sont produites. Vous pouvez sélectionner votre approche par requête. Pour les requêtes complexes susceptibles d'être sujettes à des régressions ou pour les requêtes critiques pour l'entreprise, vous pouvez utiliser une approche proactive et approuver les plans optimaux pour ces requêtes. Si d'autres requêtes subissent une régression du plan de requêtes pendant l'exécution, vous pouvez utiliser une approche réactive. Lorsque vous détectez la régression, modifiez le statut de ce plan pour `rejected` que l'optimiseur choisisse un autre plan approuvé. Pour plus d'informations, voir [Meilleures pratiques pour la gestion des plans de requêtes Aurora PostgreSQL](#) (documentation Aurora).

Comment fonctionne la gestion des plans de requêtes ?

Les plans se voient attribuer l'un des statuts suivants : `approved`, `unapproved`, `preferred`, ou `rejected`. L'optimiseur définit le premier plan généré pour chaque instruction gérée sur `approved` puis définit le statut des plans supplémentaires sur `unapproved`. Vous pourrez ensuite évaluer les `unapproved` plans et modifier leur statut en `approved`, `preferred`, ou `rejected`. Pour plus d'informations, consultez [la section Présentation de la gestion des plans de requêtes Aurora PostgreSQL](#) (documentation Aurora).

Les plans gérés peuvent être capturés manuellement ou automatiquement. L'approche la plus courante consiste à capturer automatiquement les plans pour tous les relevés exécutés deux fois ou plus. Toutefois, vous pouvez également capturer manuellement des plans pour un ensemble spécifique de relevés. Pour plus d'informations, consultez [Capture des plans d'exécution d'Aurora PostgreSQL](#) (documentation Aurora).

Une fois que vous avez configuré un plan géré, l'optimiseur utilise le coût minimal `preferred` ou le `approved` plan qui est valide et activé pour chaque instruction gérée. Pour des informations détaillées, voir [Sélection du plan à exécuter par Aurora optimiseur](#).

Pour obtenir des instructions sur la configuration de la fonctionnalité QPM dans un environnement compatible avec Amazon Aurora PostgreSQL, consultez [la section Gestion des plans d'exécution de requêtes pour Aurora PostgreSQL](#) (documentation Aurora).

Limites

Pour utiliser QPM, vous devez vous assurer que vous répondez aux exigences relatives aux instructions SQL prises en charge, que vos instructions ne font pas référence aux relations système et que votre classe d'instance de base de données possède suffisamment de vCPUs. Pour plus d'informations, consultez les sections [Instructions SQL prises en charge](#) et [Limitations relatives à la gestion des plans de requêtes](#) (documentation Aurora).

Cas d'utilisation pour la gestion des plans de requêtes

- Empêcher la régression du plan de requête : la mise à jour de la version de votre base de données présente de nombreux avantages, tels que l'amélioration des performances et de la sécurité, l'accès à de nouvelles fonctionnalités, la résolution de problèmes connus et la conformité aux exigences réglementaires. Toutefois, les mises à jour des bases de données risquent d'entraîner une régression des performances de certaines requêtes. Ce risque est plus élevé avec les mises à niveau de versions majeures, car elles peuvent contenir des modifications qui peuvent ne pas être rétrocompatibles avec les requêtes d'applications existantes. La mise en œuvre de la QPM peut aider à prévenir la régression et à stabiliser les performances lors des modifications du système. Si vous actualisez les statistiques, ajoutez un index, modifiez des paramètres ou passez à une nouvelle version d'Amazon Aurora compatible avec PostgreSQL, QPM détecte un nouveau plan mais continue à utiliser le plan approuvé, préservant ainsi la stabilité du plan.
- Fonctionnalités de test : vous pouvez consulter l'historique des plans pour toutes les instructions SQL gérées et déterminer si les nouvelles fonctionnalités de PostgreSQL ou les modifications apportées au plan améliorent les performances. Vous pouvez ensuite décider d'implémenter ces fonctionnalités ou de nouveaux plans. Pour plus d'informations, voir [Examen des plans de requête d'Aurora PostgreSQL dans la vue dba_plans](#) (documentation d'Aurora).
- Amélioration d'un plan : dans certains cas, vous préférerez peut-être corriger un plan non optimal plutôt que de le rejeter, de le désactiver ou de le supprimer. Pour plus d'informations, consultez la section [Corriger des plans à l'aide de pg_hint_plan](#) (documentation Aurora).

Ressources

Documentation AWS

- [Gestion du cache du cluster \(CCM\)](#)
- [Gestion de plans de requêtes \(QPM\)](#)

AWSarticles de blog

- [Présentation d'Aurora PostgreSQL CCM](#)
- [Présentation d'Aurora PostgreSQL QPM](#)

AWSateliers

- [Amazon Aurora Labs pour PostgreSQL : CCM](#)
- [Amazon Aurora Labs pour PostgreSQL : QPM](#)

Historique du document

Le tableau suivant décrit les modifications importantes apportées à ce guide. Si vous souhaitez être informé des futures mises à jour, vous pouvez vous abonner à un [fil RSS](#).

Modification	Description	Date
Publication initiale	—	20 janvier 2023

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.