



Verwenden Sie CCM und QPM, um die Wiederherstellungsleistung und die Ausführungspläne in Amazon Aurora PostgreSQL zu optimieren

AWSPräskriptive Leitlinien



AWSPräskriptive Leitlinien: Verwenden Sie CCM und QPM, um die Wiederherstellungsleistung und die Ausführungspläne in Amazon Aurora PostgreSQL zu optimieren

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Marken, die nicht im Besitz von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Einführung	1
Zielgruppe	2
Gezielte Geschäftsergebnisse	3
Verwaltung des Cluster-Cachach	4
Wie funktioniert CCM?	4
Einschränkungen	6
Anwendungsfälle für CCM	7
Abfrageplanverwaltung	8
Wie funktioniert QPM?	9
Einschränkungen	9
Anwendungsfälle für QPM	10
Ressourcen	11
AWS-Dokumentation	11
AWSBlog-Beiträge	11
AWSWorkshops	11
Dokumentverlauf	12
.....	xiii

Verwenden Sie CCM und QPM, um die Wiederherstellungsleistung und die Ausführungspläne in Amazon Aurora PostgreSQL zu optimieren

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

Januar 2023 ([Dokumentengeschichte](#))

Wenn Unternehmen expandieren, verwenden sie immer mehr Daten, um wichtige Entscheidungen zu treffen. Bei zunehmenden Datenmengen ist es wichtig, die Datenbankleistung zu optimieren und sie bei Systemänderungen zu stabilisieren. Workloads mit hohem Transaktionsvolumen, wie z. B. Finanztransaktionen oder Kundenbestellungen, erfordern eine stabile, konsistente und schnelle Leistung, da eine schlechte Leistung die Kundenzufriedenheit und den Geschäftsumsatz beeinträchtigen kann. Für Datenbanken, die diese hochgradig transaktionalen Workloads verarbeiten, wie z. B. Amazon Aurora PostgreSQL-Compatible Edition-Datenbank-Instances, ist es wichtig, dass Sie die verfügbaren Funktionen zur Leistungsoptimierung verstehen und implementieren.

[Amazon Aurora PostgreSQL-Compatible](#) ist eine vollständig verwaltete relationale Datenbank-Engine, die Sie bei der Einrichtung, dem Betrieb und der Skalierung von PostgreSQL-Bereitstellungen unterstützt. Aufgrund ihrer autarken Speicherarchitektur und ihrer Funktionen, mit denen Sie die Leistung in realen Workload-Szenarien mit minimalem Wartungsaufwand optimieren können, ist es eine weit verbreitete Datenbank-Engine.

Zwei dieser Funktionen sind [Cluster Cache Management \(CCM\)](#) und [Query Plan Management \(QPM\)](#). CCM ermöglicht Ihnen, die Anwendungs- und Datenbankleistung im Falle eines Failovers wiederherzustellen, und QPM unterstützt Ihnen, die Abfrageausführungspläne zu verwalten, die vom Optimierer für Ihre SQL-Anwendungen generiert wurden. Beide Funktionen können dazu beitragen, die Leistung von SQL-Abfragen zu optimieren, indem sie mehr Kontrolle über die Datenbank bieten. Dieser Leitfaden soll Managern, Produktinhabern und Datenbankarchitekten (DBAs) helfen, die Vorteile und potenziellen Geschäftsergebnisse der Implementierung von CCM und QPM zu verstehen.

Zielgruppe

Dieses Handbuch richtet sich an Geschäftsvertreter, die sich mit den verfügbaren Funktionen zur Leistungsoptimierung von Amazon Aurora PostgreSQL-kompatiblen Datenbank-Instances vertraut machen und die Anwendungsfälle für diese Funktionen verstehen möchten.

Gezielte Geschäftsergebnisse

Sie können diesen Leitfaden verwenden, um die folgenden Geschäftsergebnisse mit Cluster-Cache-Verwaltung (CCM) zu erzielen:

- Stellen Sie im Falle eines Failovers eine schnelle Wiederherstellung sicher, um eine stabile und optimale Workload-Leistung aufrechtzuerhalten.
- Reduzieren Sie Geschäftsverluste, die durch eine schlechte Workload-Leistung nach einem Failover verursacht werden.
- Helfen Sie dabei, unnötige I/O-Kosten nach einem Failover zu vermeiden.

Sie können diesen Leitfaden verwenden, um mit Query Plan Management (QPM) die folgenden Geschäftsergebnisse zu erzielen:

- Verbesserung der Planstabilität, indem der Optimierer zur Auswahl aus einer kleinen Anzahl genehmigter Pläne gezwungen wird. Dies verhindert, dass der Optimierer nach System- oder Datenbankänderungen einen suboptimalen Plan für eine bestimmte SQL-Anweisung auswählt.
- Zentrale Optimierung von Plänen, gefolgt von einer globalen Verteilung der besten Pläne.
- Ermittlung der nicht verwendeten Indizes und Bewertung der Auswirkungen einer Indexerstellung oder -löschung.
- Erkennt automatisch alle neuen Mindestkostenpläne, die der Optimierer identifiziert.
- Test neuer Optimierer-Funktionen mit minimalem Risiko, da Sie nur die Planänderungen akzeptieren können.

Verwaltung des Cluster-Cachach

Caching ist eine der wichtigsten Funktionen jeder Datenbank (DB), da es dazu beiträgt, die Festplatten-I/O zu reduzieren. Die Daten, auf die am häufigsten zugegriffen wird, werden in einem Speicherbereich gespeichert, der als Puffercache bezeichnet wird. Wenn eine Abfrage häufig ausgeführt wird, ruft sie die Daten direkt aus dem Cache statt von der Festplatte ab. Dies ist schneller und bietet eine bessere Skalierbarkeit und Anwendungsleistung. Sie konfigurieren die PostgreSQL-Cachegröße mithilfe des `shared_buffers` Parameters. Weitere Informationen finden Sie unter [Memory](#) (PostgreSQL-Dokumentation).

Nach einem Failover ist das [Cluster Cache Management \(CCM\)](#) in der Amazon Aurora PostgreSQL-Compatible Edition darauf ausgelegt, die Leistung der Anwendungs- und Datenbankwiederherstellung zu verbessern. In einer typischen Failover-Situation ohne CCM kommt es möglicherweise zu einem temporären, aber erheblichen Leistungsabfall. Der Grund ist, dass der Failover-DB-Instance der Failover-DB-Instance der Failover-DB-Instance Instance Instance Ein leerer Cache wird auch als kalter Cache bezeichnet. Die DB-Instance muss von der Festplatte lesen, was langsamer ist als das Lesen aus dem Cache.

Wenn Sie CCM implementieren, wählen Sie eine bevorzugte Reader-DB-Instance aus, und CCM synchronisiert seinen Cache-Speicher kontinuierlich mit dem der primären oder Writer-DB-Instance. Wenn ein Failover auftritt, wird die bevorzugte Reader-DB-Instance auf die neue Writer-DB-Instance befördert. Da es bereits über einen Cache-Speicher, einen sogenannten warmen Cache, verfügt, werden die Auswirkungen des Failovers auf die Anwendungsleistung minimiert.

Wie funktioniert das Cluster-Cache-Management?

Failover-DB-Instances befinden sich in anderen Availability Zones als die primäre Writer-DB-Instance. Die bevorzugte Reader-DB-Instance ist das Prioritäts-Failover-Ziel, das spezifiziert wird, indem ihr die Tier-0-Prioritätsstufe zugewiesen wird.

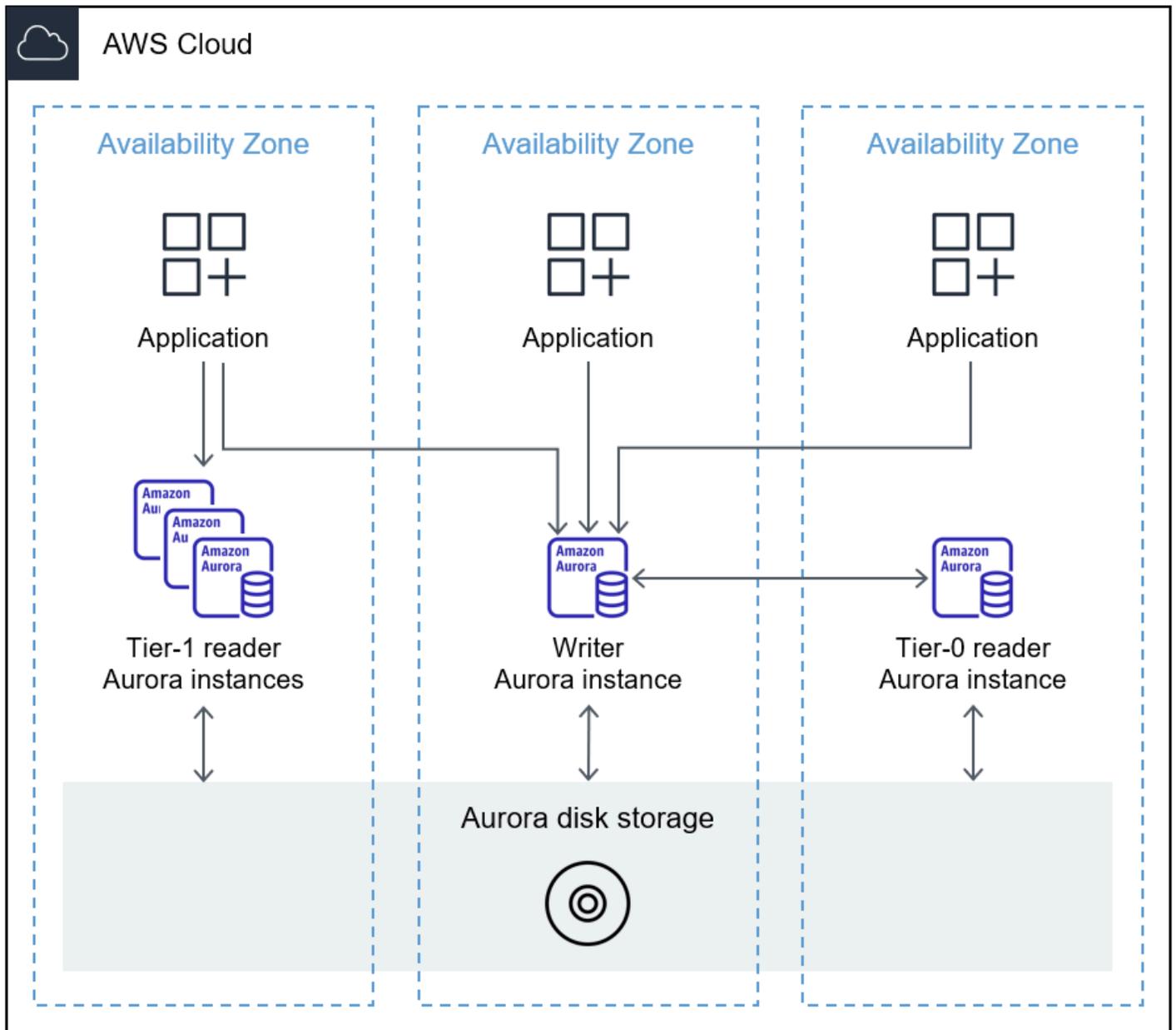
Note

Die Prioritätsstufe für das Hochstufen gibt die Reihenfolge an, in der ein Aurora-Reader nach einem Ausfall zur Writer-DB-Instance hochgestuft wird. Gültige Werte sind 0 – 15, wobei 0 die erste und 15 die letzte Prioritätsstufe darstellt. Weitere Informationen zur

Hochstufungspriorität finden Sie unter [Fehlertoleranz für einen Aurora-DB-Cluster](#). Das Ändern der Hochstufungspriorität führt nicht zu einem Nutzungsausfall.

CCM synchronisiert den Cache von der Writer-DB-Instance mit der bevorzugten Reader-DB-Instance. Die Reader-DB-Instance sendet den Satz von Pufferadressen, die derzeit zwischengespeichert sind, als Bloom-Filter an die Writer-DB-Instance. Ein Bloomfilter ist eine probabilistische, speichereffiziente Datenstruktur, mit der getestet wird, ob ein Element Teil einer Menge ist. Die Verwendung eines Bloom-Filters verhindert, dass die Reader-DB-Instance wiederholt dieselben Pufferadressen an die Writer-DB-Instance sendet. Wenn die Writer-DB-Instance den Bloom-Filter empfängt, vergleicht sie die Blöcke in ihrem Puffercache und sendet häufig verwendete Puffer an die Reader-DB-Instance. Standardmäßig gilt ein Puffer als häufig verwendet, wenn er eine Benutzungszahl von mehr als drei hat.

Das folgende Diagramm zeigt, wie CCM den Puffercache der Writer-DB-Instance mit der bevorzugten Reader-DB-Instance synchronisiert.



Weitere Informationen zu CCM finden Sie unter [Schnelle Wiederherstellung nach einem Failover mit Cluster-Cache-Management für Aurora PostgreSQL](#) (Aurora-Dokumentation) und [Einführung in das Aurora PostgreSQL-Cluster-Cache-Management](#) (AWSBlogbeitrag). Anweisungen zur Konfiguration von CCM finden Sie unter [Konfiguration der Cluster-Cache-Verwaltung](#) (Aurora-Dokumentation).

Einschränkungen

Die CCM-Funktion hat die folgenden Einschränkungen:

-
- Die Reader-DB-Instance muss den gleichen Typ und die gleiche Größe der DB-Instance-Klasse wie die Writer-DB-Instance haben, z. B. `r5.2xlarge` oder `db.r5.xlarge`.
 - CCM wird für Aurora-PostgreSQL-DB-Cluster, die Teil der globalen Aurora-Datenbanken sind, nicht unterstützt.

Anwendungsfälle für die Cluster-CachachachVerwaltung

In einigen Branchen, wie dem Einzelhandel, dem Bank- und Finanzwesen, können Verzögerungen von nur wenigen Millisekunden zu Problemen mit der Anwendungsleistung führen und zu erheblichen Geschäftsverlusten führen. Da CCM dabei hilft, die Anwendungs- und Datenbankleistung wiederherzustellen, indem es den Puffercache der primären Datenbankinstanz mit der bevorzugten Backup-Instance synchronisiert, kann es dazu beitragen, Unternehmensverluste im Zusammenhang mit Failovers zu verhindern.

Abfrageplanverwaltung

Änderungen von Statistiken, Einschränkungen, Umgebungseinstellungen, Abfrageparameter-Bindungen und Upgrades der PostgreSQL-Datenbank-Engine können eine Abfrageplanregression verursachen. Eine Abfrageplanregression liegt vor, wenn der Optimierer einen weniger optimalen Plan als vor einer bestimmten Änderung in der Datenbankumgebung auswählt.

In der Amazon Aurora PostgreSQL-Compatible Edition wurde die Funktion [Query Plan Management \(QPM\)](#) entwickelt, um die Anpassungsfähigkeit und Stabilität von Plänen zu gewährleisten, unabhängig von Änderungen der Datenbankumgebung, die zu einer Regression des Abfrageplans führen könnten. QPM bietet eine gewisse Kontrolle über den Optimierer. Mit QPM können Sie den Abfrageausführungsplan verwalten, der vom Optimierer für Ihre SQL-Abfragen generiert wurde. Der Abfrageausführungsplan zwingt den Optimierer, aus Ihren genehmigten Plänen für kritische Abfragen auszuwählen, um deren Leistung zu optimieren.

Unternehmen stellen Anwendungen und Datenbanken in der Regel global bereit oder verwalten mehrere Umgebungen für jede Anwendungsdatenbank, z. B. für Entwicklung, Qualitätssicherung, Staging, Vorproduktion, Test und Produktion. Die Verwaltung der Abfrageausführungspläne für jede Datenbank, in jeder Umgebung und für alle AWS-Regionen kann komplex und zeitaufwändig sein. QPM kann Amazon Aurora PostgreSQL-kompatible verwaltete Pläne von einer Datenbank in eine andere exportieren und importieren. Auf diese Weise können Sie den Abfrageausführungsplan zentral verwalten und Datenbanken global bereitstellen. Mit dieser Funktion können Sie eine Reihe von Plänen in einer Vorproduktionsdatenbank untersuchen, überprüfen, ob sie gut funktionieren, und sie dann in die Produktionsumgebung laden.

QPM bietet auch mehrere andere Vorteile. Beispielsweise können Sie QPM verwenden, um Ausführungspläne zu verbessern, die in Anwendungen nicht geändert werden können oder wenn der Anweisung keine Hinweise hinzugefügt werden können. QPM erkennt auch automatisch neue Pläne mit minimalen Kosten, die der Optimierer entdeckt, sodass Sie neben der Leistung auch weiterhin die Kosten optimieren können.

Es wird empfohlen, QPM zu aktivieren. Wenn QPM aktiviert ist, verwendet der Optimierer den von Ihnen genehmigten Mindestkostenplan. Dies hilft, Regressionen zu verhindern und reduziert den Zeitaufwand für die Verwaltung und Behebung suboptimaler Pläne.

Es gibt zwei verschiedene Ansätze für die Verwendung der QPM-Funktion: proaktiv und reaktiv. Der proaktive Ansatz soll dazu beitragen, Leistungsregressionen zu verhindern, und der reaktive Ansatz ist darauf ausgelegt, Leistungsregressionen zu erkennen und zu reparieren, nachdem sie aufgetreten

sind. Sie können Ihren Ansatz für jede Abfrage auswählen. Für komplexe Abfragen, die anfällig für Regressionen sein könnten, oder für geschäftskritische Abfragen können Sie einen proaktiven Ansatz verwenden und die optimalen Pläne für diese Abfragen genehmigen. Wenn bei anderen Abfragen während der Laufzeit eine Regression des Abfrageplans auftritt, können Sie einen reaktiven Ansatz verwenden. Wenn Sie die Regression feststellen, ändern Sie den Status dieses Plans in `rejected` sodass der Optimierer einen anderen, genehmigten Plan auswählt. Weitere Informationen finden Sie unter [Bewährte Methoden für die Verwaltung von Aurora PostgreSQL-Abfrageplänen](#) (Aurora-Dokumentation).

Wie funktioniert die Verwaltung von Abfrageplänen?

Plänen wird einer der folgenden Status zugewiesen: `approved`, `unapprovedpreferred`, oder `rejected`. Der Optimierer setzt den ersten generierten Plan für jede verwaltete Anweisung auf `approved` und legt dann den Status zusätzlicher Pläne auf `unapproved`. Später können Sie die `unapproved` Pläne überprüfen und ihren Status in `approvedpreferred`, oder ändern `rejected`. Weitere Informationen finden Sie unter [Grundlegendes zur Verwaltung von Aurora PostgreSQL-Abfrageplänen](#) (Aurora-Dokumentation).

Verwaltete Pläne können manuell oder automatisch erfasst werden. Der gebräuchlichste Ansatz besteht darin, Pläne für alle Anweisungen, die zwei- oder mehrmals ausgeführt werden, automatisch zu erfassen. Sie können Pläne für einen bestimmten Satz von Kontoauszügen jedoch auch manuell erfassen. Weitere Informationen finden Sie unter [Capturing Aurora PostgreSQL-Ausführungspläne](#) (Aurora-Dokumentation).

Nachdem Sie einen verwalteten Plan eingerichtet haben, verwendet der Optimierer für jede verwaltete Anweisung jenen kostengünstigsten `preferred` oder aktivierten `approved` Plan. Ausführliche Informationen finden Sie unter Erfahren Sie, [wie der Optimierer bestimmt, welche Pläne ausgeführt werden](#) (Aurora-Dokumentation).

Anweisungen zur Konfiguration der QPM-Funktion in Amazon Aurora PostgreSQL-kompatiblen Geräten finden Sie unter [Verwaltung von Abfrageausführungsplänen für Aurora PostgreSQL](#) (Aurora-Dokumentation).

Einschränkungen

Um QPM verwenden zu können, müssen Sie sicherstellen, dass Sie die Anforderungen für unterstützte SQL-Anweisungen erfüllen, dass Ihre Anweisungen keine Systembeziehungen

referenzieren und dass Ihre DB-Instance-Klasse über ausreichend vCPUs verfügt. Weitere Informationen finden Sie unter [Unterstützte SQL-Anweisungen](#) und [Einschränkungen der Abfrageplanverwaltung](#) (Aurora-Dokumentation).

Anwendungsfälle für die Abfrageplanverwaltung

- Verhinderung einer Regression des Abfrageplans — Die Aktualisierung Ihrer Datenbankversion bietet viele Vorteile, z. B. verbesserte Leistung und Sicherheit, Zugriff auf neue Funktionen, Behebung bekannter Probleme und Einhaltung gesetzlicher Anforderungen. Es besteht jedoch das Risiko, dass Datenbankaktualisierungen bei einigen Abfragen zu Leistungseinbußen führen können. Dieses Risiko ist bei größeren Versionsupgrades höher, da sie Änderungen enthalten können, die möglicherweise nicht mit vorhandenen Anwendungsabfragen abwärtskompatibel sind. Die Implementierung von QPM kann dazu beitragen, Regressionen zu verhindern und die Leistung bei Systemänderungen zu stabilisieren. Wenn Sie Statistiken aktualisieren, einen Index hinzufügen, Parameter ändern oder auf eine neue Version von Amazon Aurora PostgreSQL Compatible upgraden, erkennt QPM einen neuen Plan, verwendet aber weiterhin den genehmigten Plan, wodurch die Planstabilität gewahrt wird.
- Testfunktionen — Sie können den Planverlauf für alle verwalteten SQL-Anweisungen einsehen und beurteilen, ob neue PostgreSQL-Funktionen oder Planänderungen die Leistung verbessern. Sie können dann entscheiden, ob Sie diese Funktionen oder neue Pläne implementieren möchten. Weitere Informationen finden Sie unter [Untersuchen von Aura-PostgreSQL-PostgreSQL-Abfrageplänen in der dba_plans-Ansicht \(Aura-PostgreSQL-Abfrageplänen\)](#).
- Verbesserung eines Plans — In einigen Fällen ist es möglicherweise sinnvoll, einen nicht optimalen Plan zu reparieren, anstatt ihn abzulehnen, zu deaktivieren oder zu löschen. Weitere Informationen finden Sie unter [Pläne mithilfe von pg_hint_plan korrigieren](#) (Aurora-Dokumentation).

Ressourcen

AWS-Dokumentation

- [Cluster-Cache-Verwaltung \(CCM\)](#)
- [Abfrageplanverwaltung \(QPM\)](#)

AWSBlog-Beiträge

- [Einführung in Aurora PostgreSQL CCM](#)
- [Einführung in Aurora PostgreSQL QPM](#)

AWSWorkshops

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

Dokumentverlauf

In der folgenden Tabelle werden die wichtigsten Änderungen an diesem Handbuch beschrieben.
Wenn Sie über future Updates informiert werden möchten, können Sie einen [RSS-Feed](#) abonnieren.

Änderung	Beschreibung	Datum
Erste Veröffentlichung	—	20. Januar 2023

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.