Working with Serverless Databases

Renato Losio

Agenda

4 x 90 minutes

Unit 1: 09:00 to 10:30

Unit 2: 11:00 to 12:30

Unit 3: 13:30 to 15:00

Unit 4: 15:30 to 17:00

Agenda

Topics?

#1 Intro to serverless databases

#2 Playing with Aurora Serverless v2

#3 Playing with DynamoDB on demand

#4 Playing with Athena

#whoami

cloudiamo.com

Berlin, Trieste

Principal Cloud Architect @ Funambol

AWS Data Hero

InfoQ Staff Editor

Source on GitHub

https://github.com/rlosio/workshop

session #1

what is a serverless DB for you?

ACID

Just marketing?

serverless DB?

SQL versus NoSQL

CAP theorem

Recent posts

How you should think about DynamoDB costs

Event-Driven Architectures vs. Event-Based Compute in Serverless Applications

Why I (Still) Like the Serverless Framework over the CDK

Key Takeaways from the DynamoDB Paper

Understanding Eventual Consistency in DynamoDB

Get the DynamoDB Book

Inconsistent thoughts on database consistency

May 12, 2022 · 19 min read

Alex DeBrie
Founder, DeBrie Advisory

So, this is a post about consistency in databases. And it comes as a result of a deep dive down a rabbit hole, with hundreds of pages of academic papers printed and countless Chrome tabs eating memory on my MacBook.

A few months ago, I set out to write a post explaining some quirks of local secondary indexes in DynamoDB. As I planned that out, I realized I needed to explain some things on eventual consistency in DynamoDB. *As part of that*, I realized the notion of consistency is pretty darn confusing and contains a bunch of overlapping concepts.

(The post on DynamoDB's eventual consistency is available here! The other post on local secondary indexes is still TBD).

In this post, I'll explain my key takeaways after reviewing previous work in this space. Hopefully it will save you some time getting up to speed.

This post has two main sections. First, I'll discuss the various definitions of the word "consistency" that are used in the distributed databases space. Then, I'll discuss some of my issues with conversations about consistency.

If this is your first time reading one of my posts or hearing of me, some quick background: I work a lot with Amazon DynamoDB and wrote a book on it. I have

https://www.alexdebrie.com/posts/database-consistency/

Serverless

What is it?

#1 HA and fault tolerance

#2 No server maintenance

#3 Charge for usage

#4 Continuous scaling


ONCE UPON
A TIME

Announcing Multi-AZ Deployments for Amazon RDS

Posted On: May 18, 2010

We are excited to announce Multi-Availability Zone (Multi-AZ) deployments for Amazon Relational Database Service (Amazon RDS). This new deployment option provides enhanced availability and data durability by automatically replicating database updates between multiple Availability Zones. Availability Zones are physically separate locations with independent infrastructure engineered to be insulated from failure in other Availability Zones. When you create or modify your DB Instance to run as a Multi-AZ deployment, Amazon RDS will automatically provision and maintain a synchronous "standby" replica in a different Availability Zone. In the event of planned database maintenance or unplanned service disruption, Amazon RDS will automatically failover to the up-to-date standby so that database operations can resume quickly without administrative intervention.

The increased availability and fault tolerance offered by Multi-AZ deployments are well suited to critical production environments. To learn more, visit the Amazon RDS product page.

What is a Serverless Database?

A serverless database is any database that embodies the core principles of the serverless computing paradigm. The exact flavor of the application doesn't matter; whether a serverless database, a cloud data warehouse or even a custom backend to a CRM app, anything calling itself serverless should be built with the following principles in mind:

1. Little to no manual server management
2. Automatic, elastic app/service scale
3. Built-in resilience and inherently fault tolerant service
4. Always available and instant access
5. Consumption-based rating or billing mechanism

There are some additional, explicit principles that help define a serverless database. Data should be accurate and of high integrity, but — and of equal importance — data must also be available everywhere, and with very low latency. The very nature of serverless is inherently multi-regional: never tied to a single region and able to deliver all of this value anywhere. These four additional principles are:

6. Survive any failure domain, including regions
7. Geographic scale
8. Transactional guarantees
9. The elegance of relational SQL

https://www.cockroachlabs.com/blog/what-is-a-serverless-database/


Relational DB
Distributed SQL
Serverless DB

https://twitter.com/tjholowaychuk/status/902999008674594816

D1: open beta is here

09/28/2023

9 min read

https://blog.cloudflare.com/d1-open-beta-is-here/

CLOUD

Google Cloud Spanner Adds PostgreSQL Interface: Is PostgreSQL Now Standard for Cloud Databases?

OCT 29, 2021 • 2 MIN READ

by

Renato Losio

InfoQ Staff Editor | Cloud Expert | AWS Data Hero

At the recent Next '21 conference, Google announced a PostgreSQL interface for Cloud Spanner, the distributed SQL database management and storage service on Google Cloud. The new interface in preview combines the scalability and reliability of Spanner with the familiarity among developers and portability of PostgreSQL.

The preview release does not yet achieve full parity between the Google Standard SQL and PostgreSQL interfaces, but longer term Google recommends to determine which dialect to use according to how familiar a development team is with the given technology. Justin Makeig, product manager at Google, explains the benefits and limitations of the new approach:

> At the lowest level, the PostgreSQL interface implements a core subset of

RELATED CONTENT

Google Cloud Unveils AlloyDB AI: Transforming PostgreSQL with Advanced Vector Embeddings and AI
SEP 06, 2023

Scale Higher Per-Database Storage Limits and Create More Databases with Cloudflare D1 Open Beta
OCT 02, 2023

Google Expands Duet AI in Google Cloud for App Development, DevOps, and More
SEP 04, 2023

https://www.infoq.com/news/2021/10/cloud-spanner-postgresql/

serverless DB

≠

FaaS

compatibility

≠

platform lock-in

When the shoulders of giants are offered, you'd do well to stand on them

Quite often, I read and hear about people who are reluctant to embrace the AWS Serverless ecosystem. Some of their reasons are valid, some are rooted in uncertainty

https://lucvandonkersgoed.com/2023/10/13/if-the-shoulders-of-giants-are-offered-youd-do-well-to-stand-on-them/

Serverless

FaaS issues?

Serverless locks you in

Serverless is too expensive

Serverless is too difficult

Serverless is too opinionated

Vintage Serverless

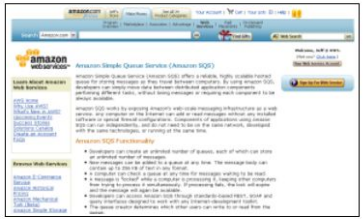
AWS News Blog

Amazon Simple Queue Service Released

by Jeff Barr | on 13 JUL 2006 | Permalink | Share

0:00 / 0:00

Voiced by Amazon Polly



A few paragraphs into yesterday's press release was an important note about the Amazon Simple Queue Service, or SQS.

SQS is now in production.

The production release allows you to have an unlimited number of queues per account, with an unlimited number of items in each queue. Each item can be up to 256KB in length. There's a complete access control model for each queue — you can independently control who is allowed to read and to write to each of your queues.

The SQS API is lean and mean, with straightforward APIs to let you create queues, send messages, receive messages, delete messages, list your queue collection, and to delete entire queues. If your queue has more than one reader, you can use the timed visibility model to ensure that each item will be processed, but that no item will be lost if a reader fails between the time that it receives and deletes a message.

Pricing is on a pay-as-you go basis, with no startup fee or minimum monthly charge. You pay 10 cents for every thousand messages sent ($0.0001 per message) and 20 cents per gigabyte of data transferred. You can learn more about pricing on the SQS Service Detail page.

Resources

Getting Started
What's New
Top Posts
Official AWS Podcast
Case Studies

Follow

Twitter
Facebook
LinkedIn
Twitch
RSS Feed
Email Updates

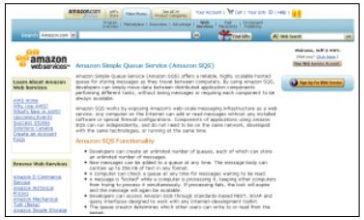
AWS News Blog

Amazon Simple Queue Service Released

by Jeff Barr | on 13 JUL 2006 | Permalink | Share

0:00 / 0:00

Voiced by Amazon Polly



A few paragraphs into yesterday's press release was an important note about the Amazon Simple Queue Service, or SQS.

SQS is now in production.

The production release allows you to have an unlimited number of queues per account, with an unlimited number of items in each queue. Each item can be up to 256KB in length. There's a complete access control model for each queue — you can independently control who is allowed to read and to write to each of your queues.

The SQS API is lean and mean, with straightforward APIs to let you create queues, send messages, receive messages, delete messages, list your queue collection, and to delete entire queues. If your queue has more than one reader, you can use the timed visibility model to ensure that each item will be processed, but that no item will be lost if a reader fails between the time that it receives and deletes a message.

Pricing is on a pay-as-you go basis, with no startup fee or minimum monthly charge. You pay 10 cents for every thousand messages sent ($0.0001 per message) and 20 cents per gigabyte of data transferred. You can learn more about pricing on the SQS Service Detail page.

Resources

Getting Started
What's New
Top Posts
Official AWS Podcast
Case Studies

Follow

Twitter
Facebook
LinkedIn
Twitch
RSS Feed
Email Updates

Search Ctrl K

TOPICS

All

Architect's Guide

Operator's Guide

Best Practices

AI/ML

Modern Data Lakes

Performance

Kubernetes

Integrations

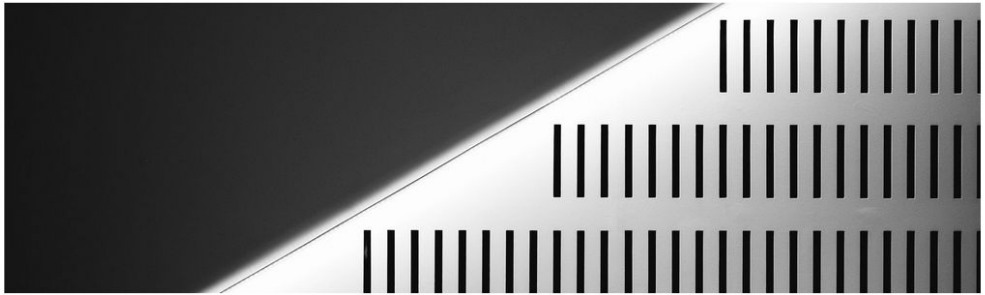
Benchmarks

Security

Multicloud

Databases on Object Storage - the New Normal

Ugur Tigli on Scalability | 19 July 2021



When you think about object storage workloads and storage types - databases are not the first thing that comes to mind. That is changing rapidly, however, driven by just two forces: the availability of true, high performance object storage and explosive growth of data and, perhaps more impactfully, its associated metadata.

Because of these two forces, almost every major database vendor now includes S3 compatible endpoints. Further, for many organizations and most workloads, this becomes the default architecture whether in the cloud or on-prem.

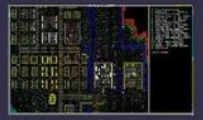
Let's explore the concepts briefly.

Performance

The storage performance requirements associated with databases have inverted in the past few years. Databases previously demanded high IOPS. This was a function of the need to make lots of small changes across the network. This was well suited to the SAN and NAS architectures and thus databases became their bread and butter. The problem is that IOPS is not

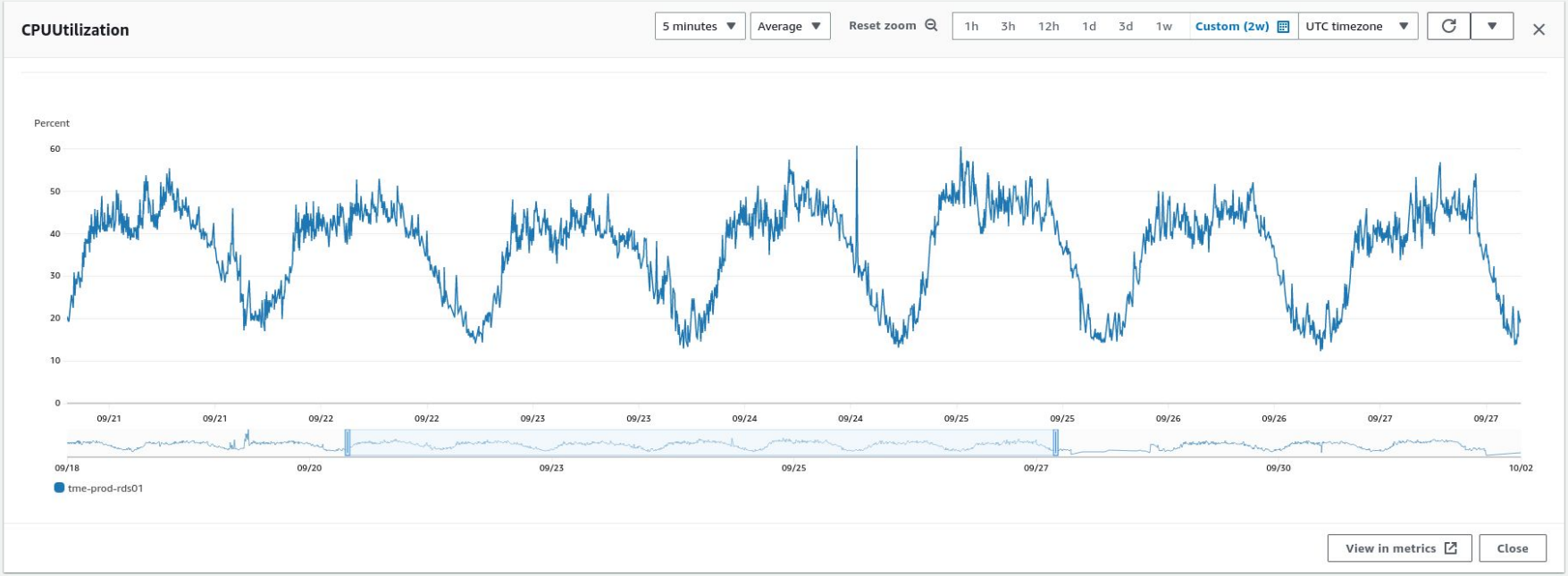
https://blog.min.io/databases-on-object-storage/

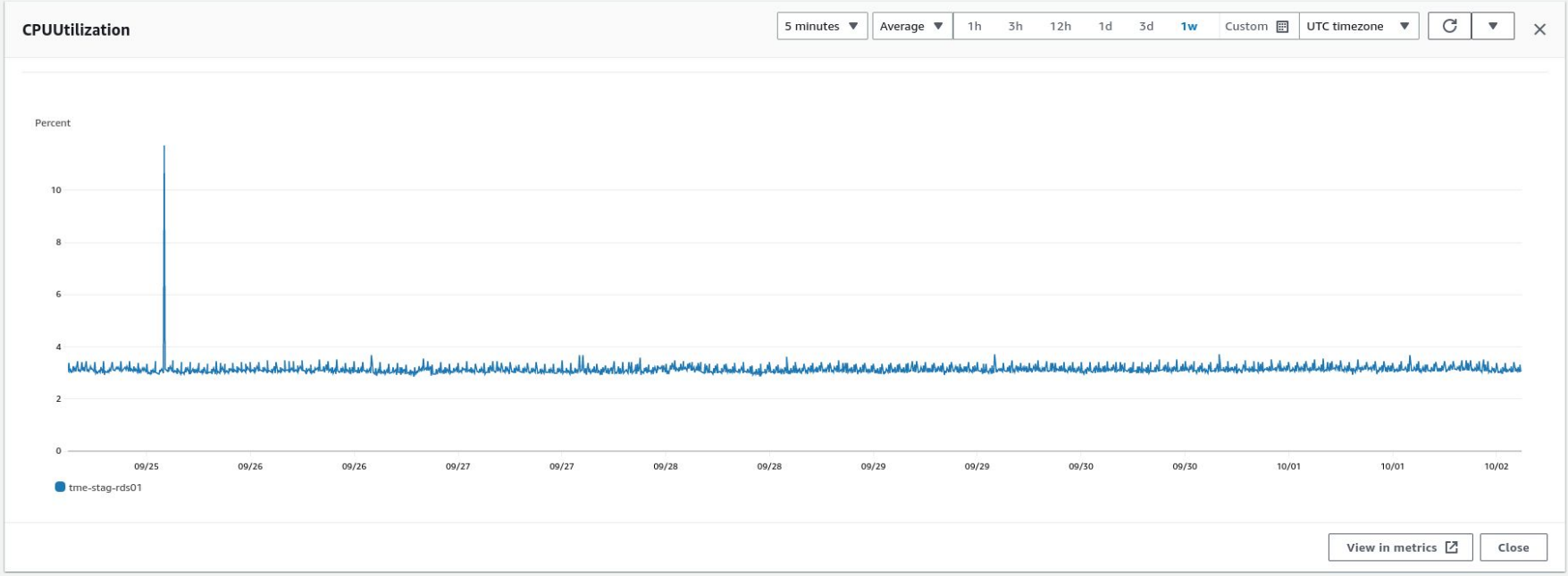
Database Alignment Chart

	Access Purist (must be queryable with a query language)	Access Neutral (must be queryable with a language)	Access Radical (queryable in any way)
Function Purist (must contain digital data)	 PostgresSQL is a database	 Excel is a database	 Dwarf Fortress is a database
Function Neutral (must contain information)	 A library is a database	 A senior engineer is a database	 A file cabinet is a database
Function Radical (can contain anything)	 Battleship is a database	 Subway checkout counter is a database	 A fridge is a database

https://www.reddit.com/r/ProgrammerHumor/comments/y3ykga/database_alignment_chart/

unused capacity?


CPUUtilization
5 minutes
Average
Reset zoom
1h
3h
12h
1d
3d
1w
Custom (2w)
UTC timezone
Percent
60
50
40
30
20
10
0
09/21
09/21
09/22
09/22
09/23
09/23
09/24
09/24
09/25
09/25
09/26
09/26
09/27
09/27
09/18
09/20
09/23
09/25
09/27
09/30
10/02
tme-prod-rds01
View in metrics
Close


CPUUtilization
5 minutes
Average
1h
3h
12h
1d
3d
1w
Custom
UTC timezone
Percent
10
8
6
4
2
0
09/25
09/26
09/26
09/27
09/27
09/28
09/28
09/29
09/29
09/30
09/30
10/01
10/01
10/02
tme-stag-rds01
View in metrics
Close

connect to *cname*
as before

< demo >

ideology

fine-grained

Pricing

who cares!

better a micro!

Use Cases

1. Intermittent load
2. Development/test environments
3. Serverless deployments
4. Unpredictable load
5. Handling bugs


bugs, indexes &
performances

< demo >

The Dream

A relational (MySQL) DB cluster that automatically starts up, shuts down, and scales up or down capacity based on my application's needs.

Challenges of RDBMS

Connection limits

Networking issues

Provisioning issues

Scaling issues

Infrastructure as code issues

AWS Cloud Databases

Modernize your data infrastructure with fully managed, purpose-built databases

Choose the right purpose-built engine

Build use case-driven, highly scalable, distributed applications suited to your specific needs. AWS offers 15+ purpose-built engines to support diverse data models, including relational, key-value, document, in-memory, graph, time series, wide column, and ledger databases.

Achieve performance at scale

Start small and scale as your applications grow with relational databases that are 3-5X faster than popular alternatives, or non-relational databases that give you microsecond to sub-millisecond latency. Match your storage and compute needs easily, often with no downtime.

Run fully managed databases

Free your teams from time-consuming database tasks like server provisioning, patching, and backups. AWS fully managed database services provide continuous monitoring, self-healing storage, and automated scaling to help you focus on application development.

Rely on high availability and security

Support multi-region, multi-primary replication, and provide full data oversight with multiple levels of security, including network isolation and end-to-end encryption. AWS databases deliver the high availability, reliability, and security you need for business-critical, enterprise workloads.

AWS Databases: Break Free to Save, Grow, and Innovate Faster (2:02)

Database services

Built for no purpose databases.

Rick Houlihan, inventor of DynamoDB


Trident
Trident
Trident
Trident
Trident
Trident
Trident

The Paradox of Choice: Jam Experiment

PERSONALITY PROCESSES AND INDIVIDUAL DIFFERENCES

When Choice is Demotivating: Can One Desire Too Much of a Good Thing?

Sheena S. Iyengar
Columbia University

Mark R. Lepper
Stanford University

Current psychological theory and research affirm the positive affective and motivational consequences of having personal choice. These findings have led to the popular notion that the more choice, the better—that the human ability to manage, and the human desire for, choice is unlimited. Findings from 3 experimental studies starkly challenge this implicit assumption that having more choices is necessarily more intrinsically motivating than having fewer. These experiments, which were conducted in both field and laboratory settings, show that people are more likely to purchase gourmet jams or chocolates or to undertake optional class essay assignments when offered a limited array of 6 choices rather than a more extensive array of 24 or 30 choices. Moreover, participants actually reported greater subsequent satisfaction with their selections and wrote better essays when their original set of options had been limited. Implications for future research are discussed.

https://faculty.washington.edu/jdb/345/345%20Articles/Iyengar%20%26%20Lepper%20(2000).pdf

Aurora, DynamoDB and Athena on AWS

MySQL (or PostgreSQL) on AWS

“EC2” #flexibility	“RDS” #managed
“Aurora” #enterprise	“Serverless” ” #simplicity

Auto Scaling a MySQL Database to Meet Fluctuating Application Demands with Amazon Aurora

TUTORIAL

Overview

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases. In this tutorial, you will learn how to create an Amazon Aurora database and configure it to scale automatically, by adding or removing read replicas, to meet fluctuating demands of your application.

This tutorial is not within the free tier and will cost you less than $1 provided you follow the steps in the tutorial and terminate your resources at the close of the tutorial.

What you will accomplish

In this tutorial, you will:

- create an Amazon Aurora database
- configure it to scale automatically to meet fluctuating demand by adding or removing read replicas

Prerequisites

✔ AWS experience	Intermediate
Time to complete	10-20 minutes
$ Cost to complete	Less than $1, provided you follow the steps to terminate your resources at the end of the tutorial
Requires	AWS account* *Accounts created within the past 24 hours might not yet have access to the services required for this tutorial.
Services used	Amazon Aurora, Amazon RDS
Last updated	February 3, 2023

Aurora Serverless v1

vintage auto-scaling database

Endpoint without instance size

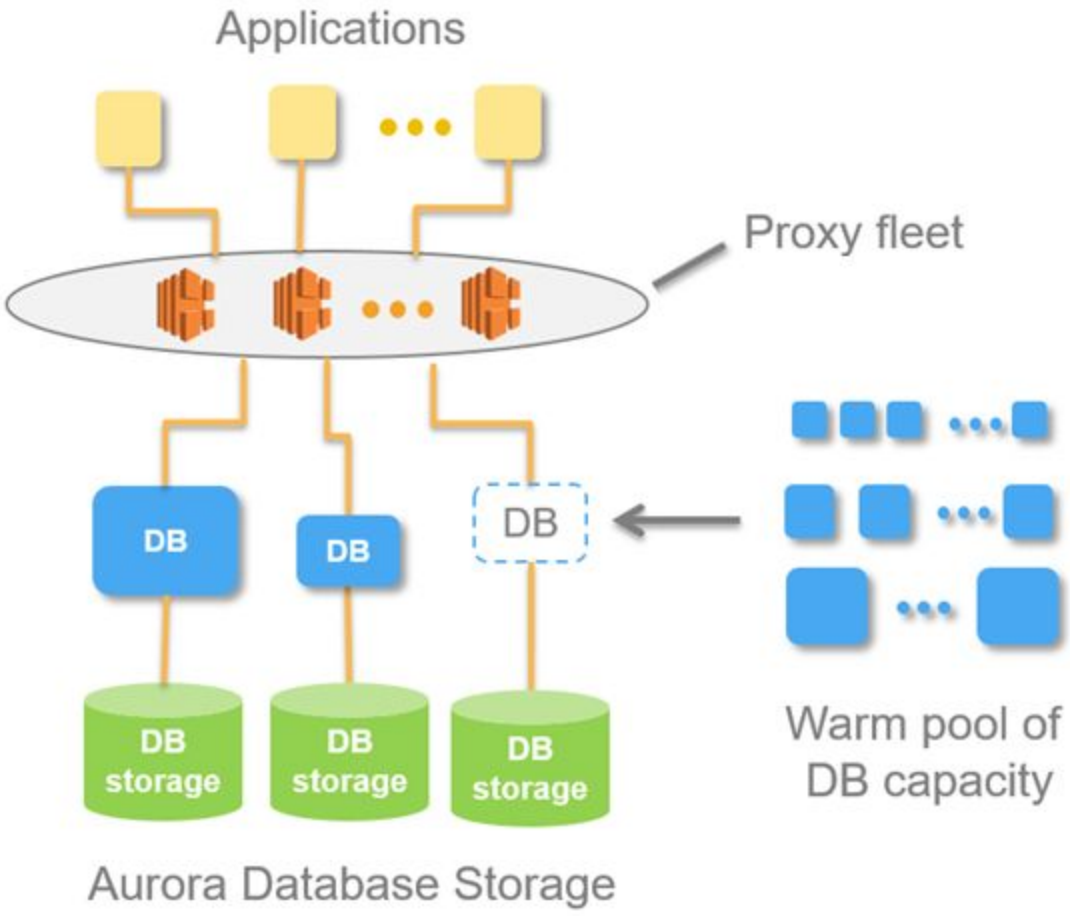
Easy scaling

Pay-per-use

MySQL 5.6 5.7/PostgreSQL

GA 5 years

TCP connection / API


Applications
Proxy fleet
DB
DB
DB
DB storage
DB storage
DB storage
Warm pool of DB capacity
Aurora Database Storage

Serverless Aurora or
Autoscaling Aurora?

Aurora Serverless Data API

no read replica

No Multi AZ

limitations

max 0.5 TB

MySQL 5.6 - 5.7
only

vertical scaling

#elasticity

Killed by Google

Search

All (291)

Google Jamboard

December 2024

App

Like a fork stuck in the outlet in about 1 year, Google Jamboard was a web and native whiteboard app that offered a rich collaborative experience. It will be about 8 years old.

Google Podcasts

September 2024

App

Fading into darkness in 12 months, Google Podcasts was a podcast hosting platform and an Android podcast listening app. It will be over 6 years old.

Jamboard

September 2024

Hardware

Done for in 12 months, Jamboard was a digital 4K touchscreen whiteboard device that allowed to collaborate using Google Workspace services. It will be over 7 years old.

Google Domains

2014 - 2023

Service

Killed 6 days ago, Google Domains was a domain name registrar operated by Google. It was over 9 years old.

Google Optimize

2012 - 2023

Service

Killed 6 days ago, Google Optimize was a web analytics and testing tool that allowed users to run experiments aimed at increasing visitor conversion rates and overall satisfaction. It was over 11 years old.

Google Cloud IoT Core

2018 - 2023

Service

Killed about 2 months ago, Google Cloud IoT Core was a managed service designed to let customers securely connect, manage, and ingest data from globally dispersed devices. It was over 5 years old.

Google Album Archive

2016 - 2023

Service

Killed 3 months ago, Google Album Archive was a platform that allowed users to access and manage their archived photos and videos from various Google services, such as Hangouts and Picasa Web Albums. It was almost 7 years old.

YouTube Stories

2017 - 2023

Service

Killed 3 months ago, YouTube Stories (originally YouTube Reels) allowed creators to post temporary videos that would expire after seven days. It was over 5 years old.

Amazon S3 Reduced Redundancy Storage

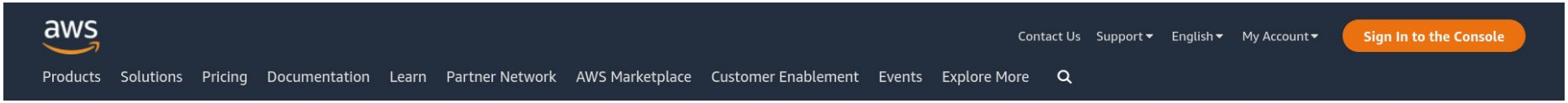
Reduced Redundancy Storage (RRS) is an Amazon S3 storage option that enables customers to store noncritical, reproducible data at lower levels of redundancy than Amazon S3's standard storage. It provides a highly available solution for distributing or sharing content that is durably stored elsewhere, or for storing thumbnails, transcoded media, or other processed data that can be easily reproduced. The RRS option stores objects on multiple devices across multiple facilities, providing 400 times the durability of a typical disk drive, but does not replicate objects as many times as standard Amazon S3 storage.

Reduced Redundancy Storage is:

- Backed with the Amazon S3 Service Level Agreement for availability.
- Designed to provide 99.99% durability and 99.99% availability of objects over a given year. This durability level corresponds to an average annual expected loss of 0.01% of objects.
- Designed to sustain the loss of data in a single facility.

	Standard	Standard - Infrequent Access	Reduced Redundancy Storage
Durability	99.999999999%	99.999999999%	99.99%

https://aws.amazon.com/s3/reduced-redundancy/



PRODUCTS & SERVICES

- Amazon SimpleDB
- Product Details
- Pricing
- Getting Started
- Developer Resources
- FAQs

RELATED LINKS

- Documentation
- Release Notes
- Discussion Forum

Get Started for Free

Create Free Account

Amazon SimpleDB

Create Free Account »

Amazon SimpleDB is a highly available NoSQL data store that offloads the work of database administration. Developers simply store and query data items via web services requests and Amazon SimpleDB does the rest.

Unbound by the strict requirements of a relational database, Amazon SimpleDB is optimized to provide high availability and flexibility, with little or no administrative burden. Behind the scenes, Amazon SimpleDB creates and manages multiple geographically distributed replicas of your data automatically to enable high availability and data durability. The service charges you only for the resources actually consumed in storing your data and serving your requests. You can change your data model on the fly, and data is automatically indexed for you. With Amazon SimpleDB, you can focus on application development without worrying about infrastructure provisioning, high availability, software maintenance, schema and index management, or performance tuning.

Benefits

Low touch

The service allows you to focus fully on value-added application development, rather than arduous and time-consuming database administration. Amazon SimpleDB automatically manages infrastructure provisioning, hardware and software maintenance, replication and indexing of data items, and performance tuning.

https://aws.amazon.com/simpledb/

COM202

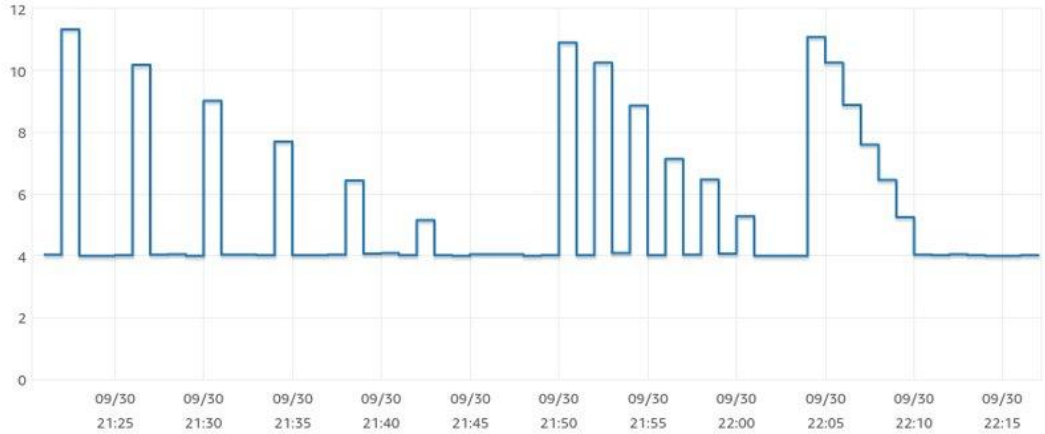
Drawing the New York City skyline with Amazon Aurora Serverless v2

Renato Losio (he/him)
Principal Cloud Architect, AWS Data Hero
Funambol

Metrics

Metric: Serverless ... | Statistic: Average | Time Range: Last 24 Ho... | Period: 1 Minute

Legend: test-aurora-v2

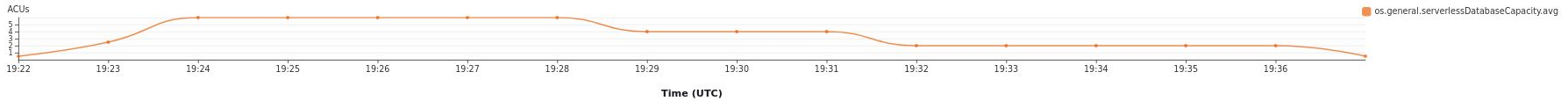


milano-instance-1
Aurora MySQL 8.0.mysql_aurora.3.03.1 db.serverless

2023-07-05T21:21:55+02:00 — 2023-07-05T21:36:56+02:00 | Auto refresh

Counter metrics

Manage metrics

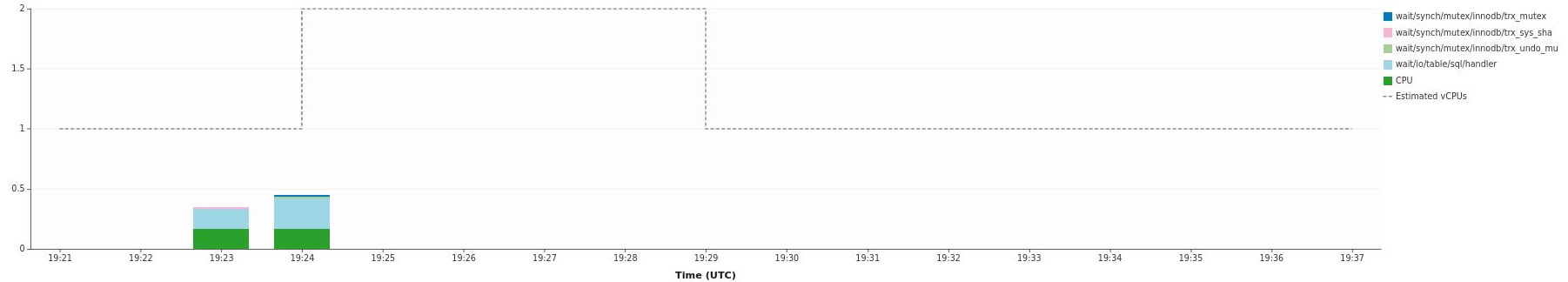


Database load

Load is measured in average active sessions (AAS). Info

Chart type: Bar | Slice by: Waits

Show estimated vCPU



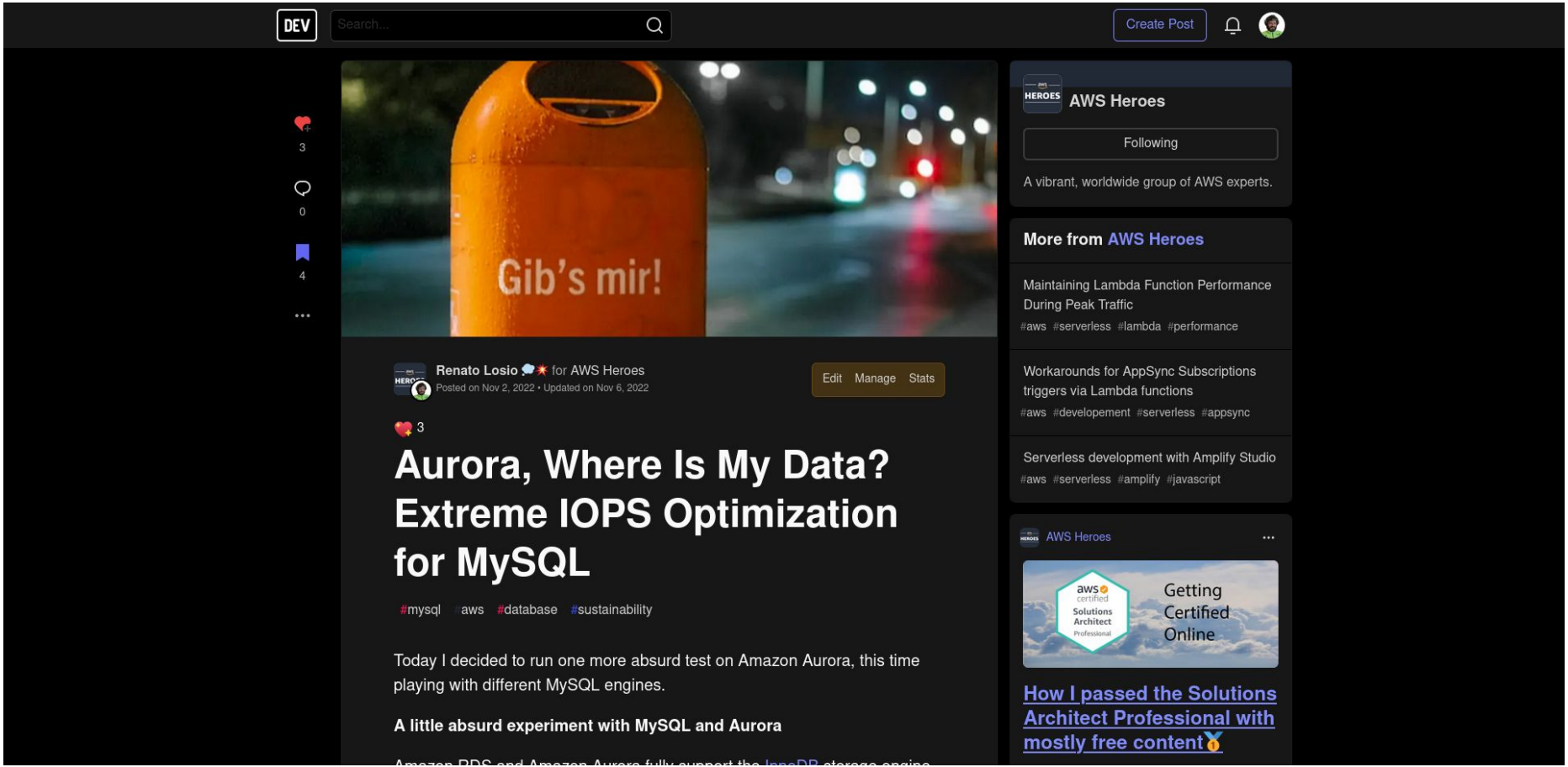
Load 100K items

one at a time
single thread
ACID

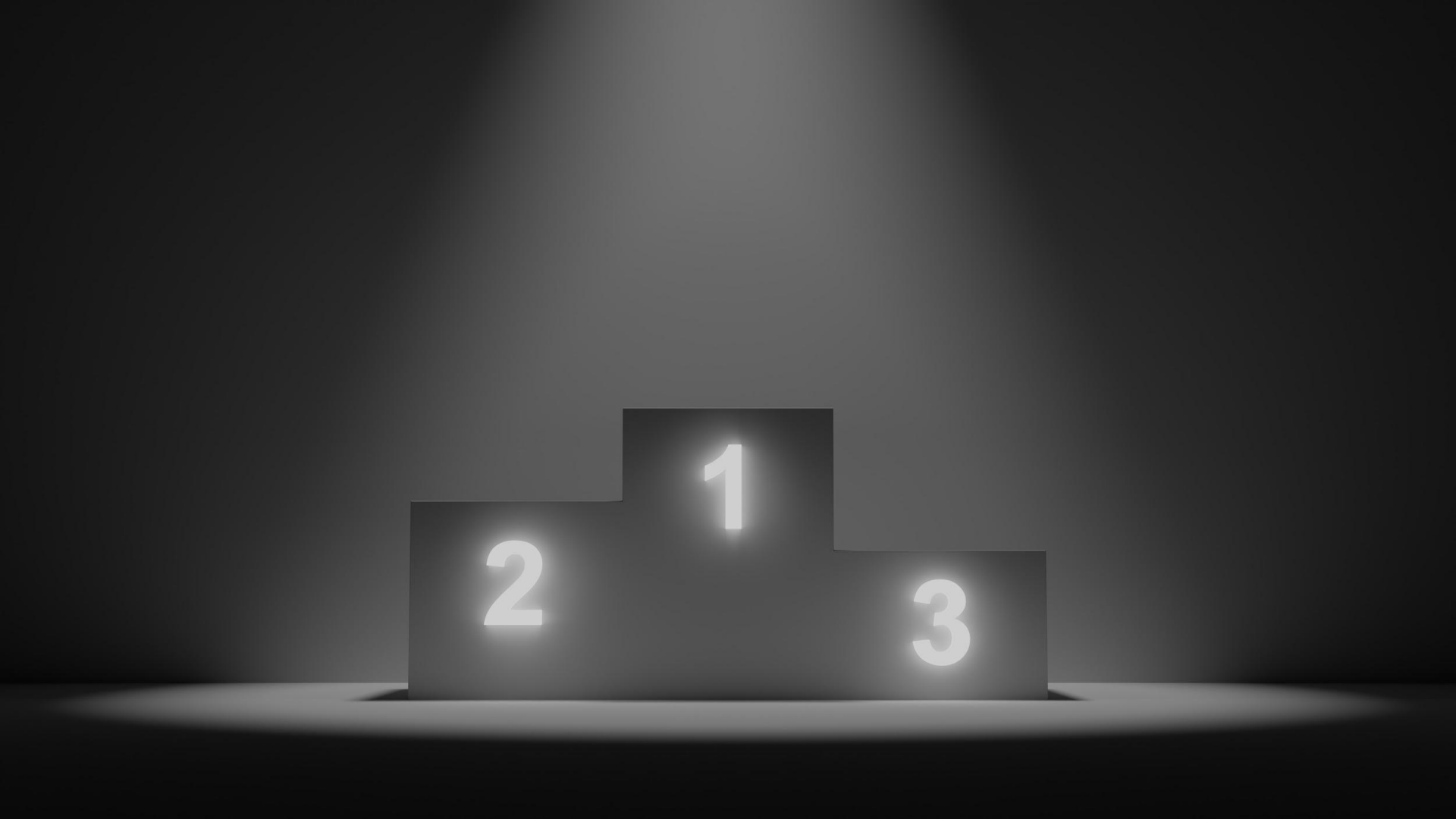
- Aurora Serverless (Min 4ACU)
- Aurora Serverless (Min 0.5 ACU)
- Aurora db.t4g.medium
- RDS t4g.medium (100G, GP3)
- RDS t4g.medium (AZ, 100G, GP3)
- ....

```
CREATE TABLE `workshopdemo`
(`id` bigint(20) NOT NULL AUTO_INCREMENT,
`datetime` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
`value` float DEFAULT NULL,
PRIMARY KEY (`id`));


DELIMITER $$
CREATE PROCEDURE load_workshopdemo()
BEGIN
  DECLARE i INT DEFAULT 0;
  WHILE i < 100000 DO
      INSERT INTO `workshopdemo` (`datetime`,`value`) VALUES (
      FROM_UNIXTIME(UNIX_TIMESTAMP('2023-10-01 01:00:00')+FLOOR(RAND()*31536000)),
      ROUND(RAND()*100,2));
      SET i = i + 1;
  END WHILE;
END$$
DELIMITER ;

CALL load_workshopdemo();
SELECT sleep(5);
CALL load_workshopdemo();
```

< demo >



https://dev.to/aws-heroes/aurora-where-is-my-data-extreme-iops-optimization-for-mysql-2i0p


1
2
3

1) Aurora **Serverless** (Min 4 ACU) 7,03 sec

2) Aurora **Serverless** (Min 0.5 ACU) 15,97 sec, then 9,19 sec

3) **Aurora** db.t4g.medium 6,77 sec

4) **Aurora** db.r5.large 6,66 sec

5) **RDS** db.t4g.medium (100G) 7 min 22,54 sec

6) **RDS** db.t4g.medium (400G) 7 min 20,17 sec

7) **RDS** db.t4g.medium (400G innodb_flush_log_at_trx_commit=2) 4 min 32,82

Auto Scaling a MySQL Database to Meet Fluctuating Application Demands with Amazon Aurora

TUTORIAL

Overview

Amazon Aurora is a MySQL and PostgreSQL-compatible relational database that combines the performance and availability of traditional enterprise databases with the simplicity and cost-effectiveness of open source databases. In this tutorial, you will learn how to create an Amazon Aurora database and configure it to scale automatically, by adding or removing read replicas, to meet fluctuating demands of your application.

This tutorial is not within the free tier and will cost you less than $1 provided you follow the steps in the tutorial and terminate your resources at the close of the tutorial.

✔ AWS experience	Intermediate
Time to complete	10-20 minutes
$ Cost to complete	Less than $1, provided you follow the steps to terminate your resources at the end of the tutorial
Requires	AWS account* *Accounts created within the past 24 hours might not yet have access to the services required for this tutorial.
Services used	Amazon Aurora, Amazon RDS
Last updated	February 3, 2023

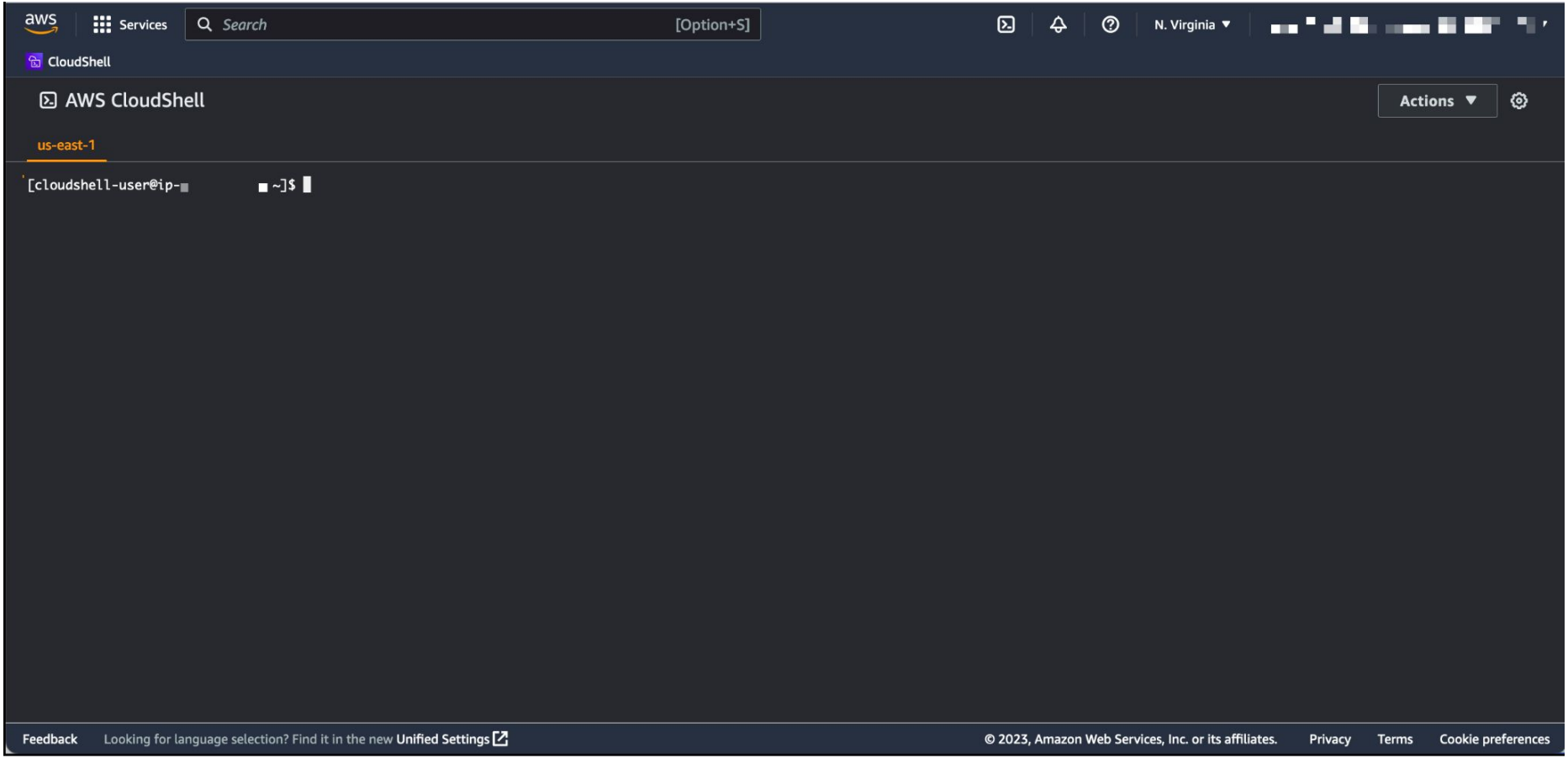
What you will accomplish

In this tutorial, you will:

- create an Amazon Aurora database
- configure it to scale automatically to meet fluctuating demand by adding or removing read replicas

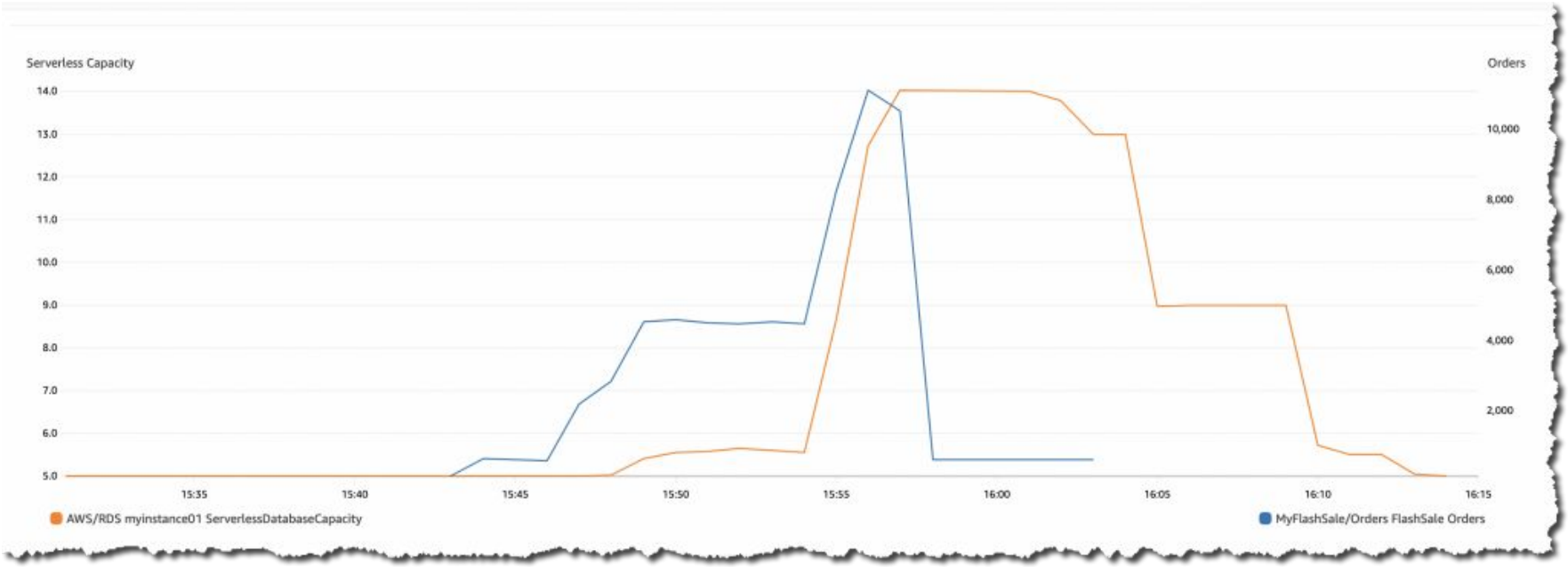
Prerequisites

https://docs.aws.amazon.com/cli/



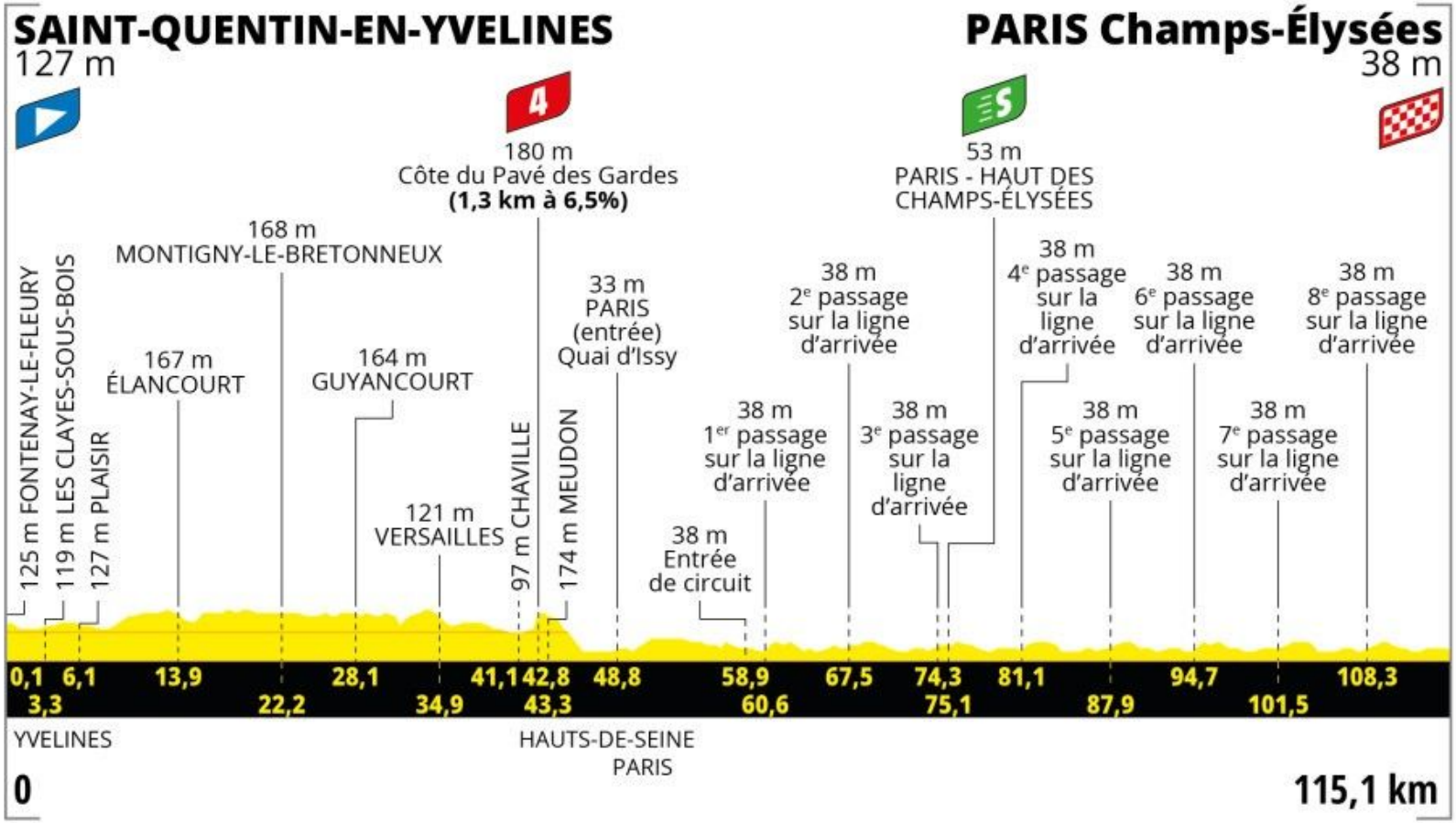
https://docs.aws.amazon.com/cloudshell/latest/userguide/welcome.html

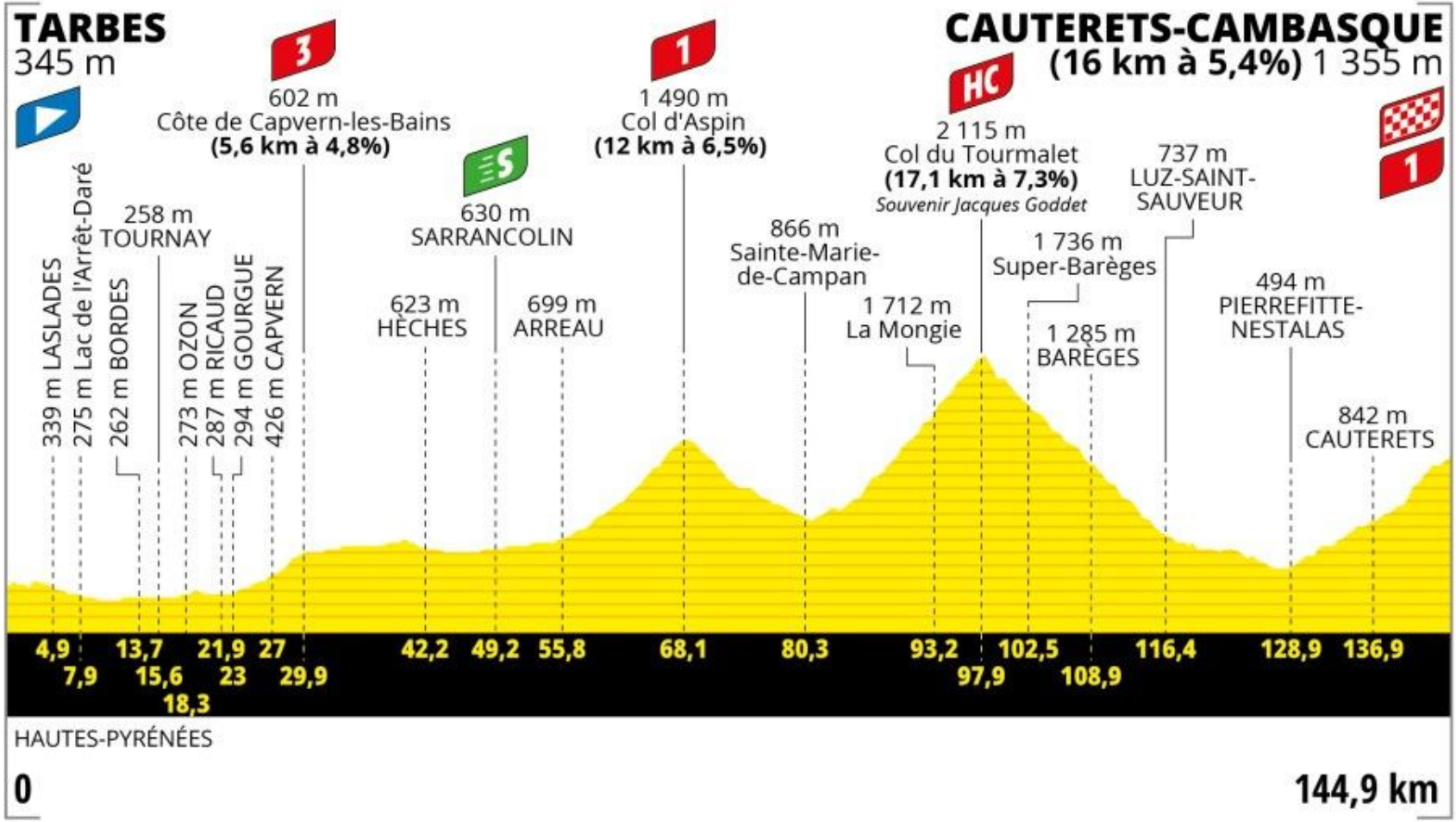
Sale Event E-commerce Site



https://aws.amazon.com/blogs/aws/amazon-aurora-serverless-v2-is-generally-available-instant-scaling-for-demanding-workloads/

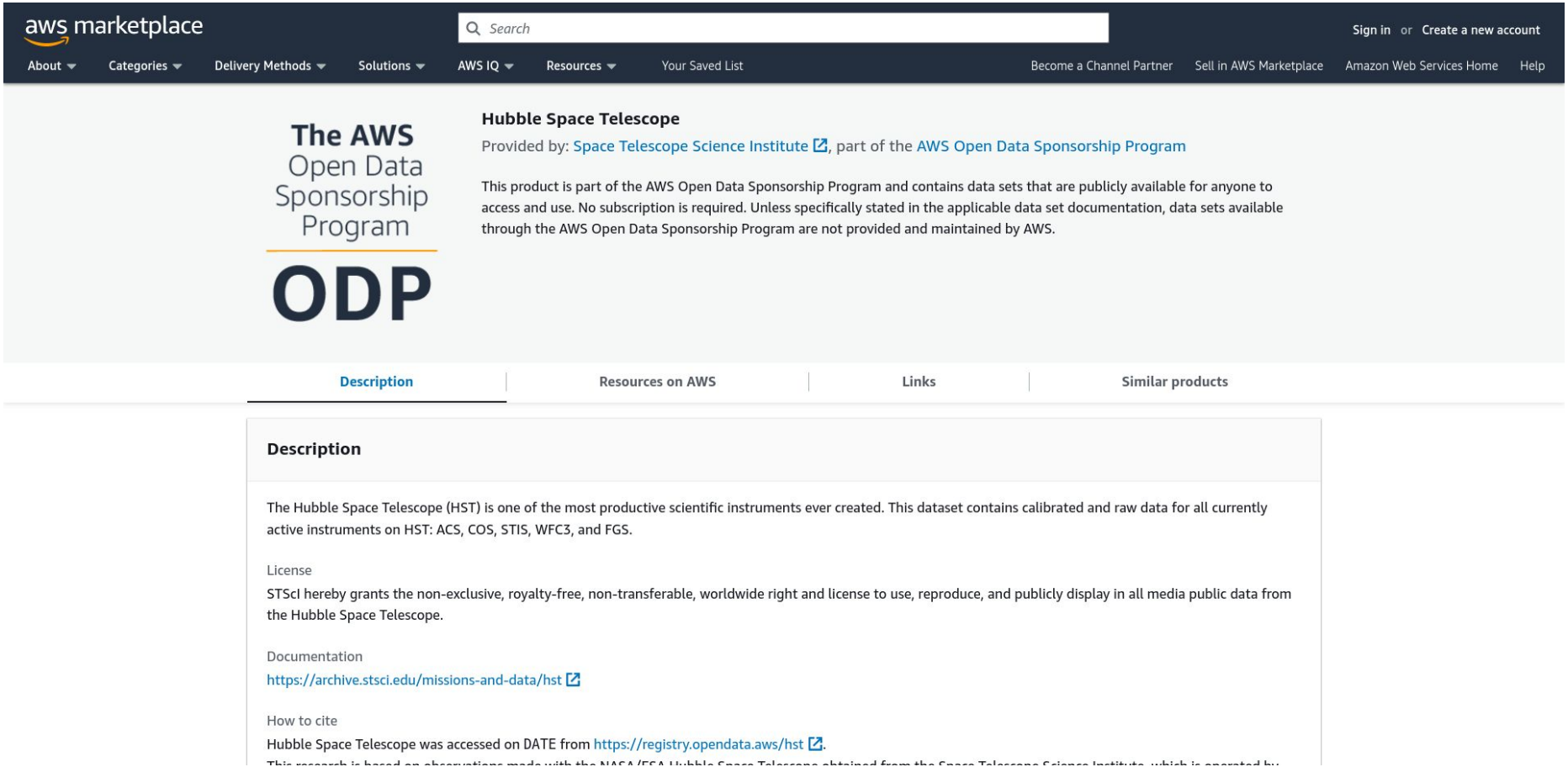
https://www.letour.fr/


SAINT-QUENTIN-EN-YVELINES
127 m
PARIS Champs-Élysées
38 m
4
5
180 m
Côte du Pavé des Gardes
(1,3 km à 6,5%)
53 m
PARIS - HAUT DES
CHAMPS-ÉLYSÉES
168 m
MONTIGNY-LE-BRETONNEUX
33 m
PARIS
(entrée)
Quai d'Issy
38 m
2e passage
sur la ligne
d'arrivée
38 m
4e passage
sur la
ligne
d'arrivée
38 m
6e passage
sur la ligne
d'arrivée
38 m
8e passage
sur la ligne
d'arrivée
167 m
ÉLANCOURT
164 m
GUYANCOURT
38 m
1er passage
sur la ligne
d'arrivée
38 m
3e passage
sur la
ligne
d'arrivée
38 m
5e passage
sur la ligne
d'arrivée
38 m
7e passage
sur la ligne
d'arrivée
125 m FONTENAY-LE-FLEURY
119 m LES CLAYES-SOUS-BOIS
127 m PLAISIR
121 m
VERSAILLES
97 m CHAVILLE
174 m MEUDON
38 m
Entrée
de circuit
0,1
6,1
13,9
28,1
41,1
42,8
48,8
58,9
67,5
74,3
81,1
94,7
108,3
3,3
22,2
34,9
43,3
60,6
75,1
87,9
101,5
YVELINES
HAUTS-DE-SEINE
PARIS
0
115,1 km

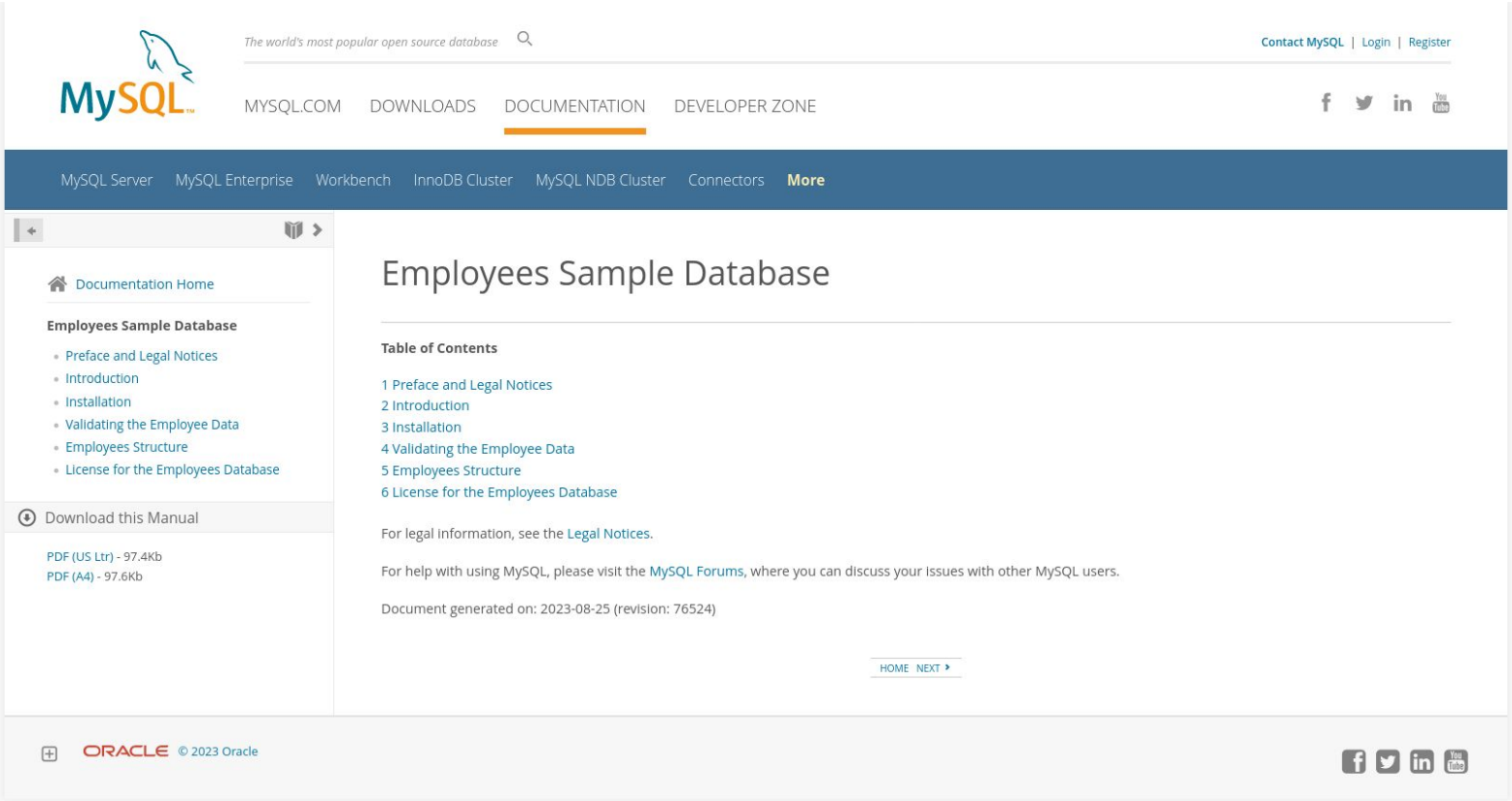

TARBES
345 m
CAUTERETS-CAMBASQUE
(16 km à 5,4%) 1 355 m
602 m
Côte de Capvern-les-Bains
(5,6 km à 4,8%)
1 490 m
Col d'Aspin
(12 km à 6,5%)
2 115 m
Col du Tourmalet
(17,1 km à 7,3%)
Souvenir Jacques Goddet
630 m
SARRANCOLIN
737 m
LUZ-SAINT-
SAUVEUR
866 m
Sainte-Marie-
de-Campan
1 736 m
Super-Barèges
494 m
PIERREFITTE-
NESTALAS
339 m LASLADES
275 m Lac de l'Arrêt-Darré
262 m BORDES
258 m
TOURNAY
273 m OZON
287 m RICAUD
294 m GOURGUE
426 m CAPVERN
623 m
HÈCHES
699 m
ARREAU
1 712 m
La Mongie
1 285 m
BARÈGES
842 m
CAUTERETS
4,9
7,9
13,7
15,6
18,3
21,9
23
27
29,9
42,2
49,2
55,8
68,1
80,3
93,2
97,9
102,5
108,9
116,4
128,9
136,9
HAUTES-PYRÉNÉES
0
144,9 km

datasets

Registry of Open Data on AWS / AWS Data Exchange



https://registry.opendata.aws/hst/



https://dev.mysql.com/doc/employee/en/

Q&A

session #2

Aurora Serverless v2

Aurora but serverless

Endpoint without instance size

Incremental scaling

Pay-per-use (2x)/ API

Multi AZ

MySQL 8.0 / PostgreSQL TCP connection

fast?

auto-scale in milliseconds based on application load

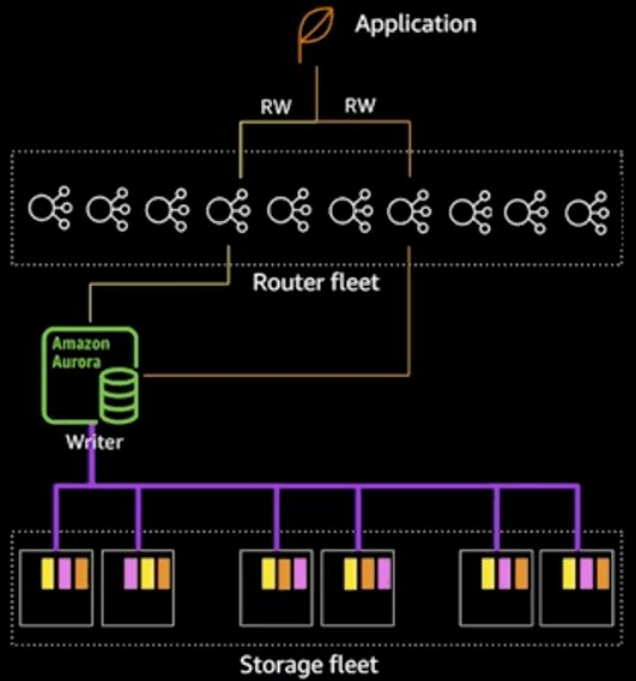
Aurora capacity units (ACU)

v1: 1 2 4 8 16 32 64 128 256
v2: 0.5 1 1.5 2 2.5...

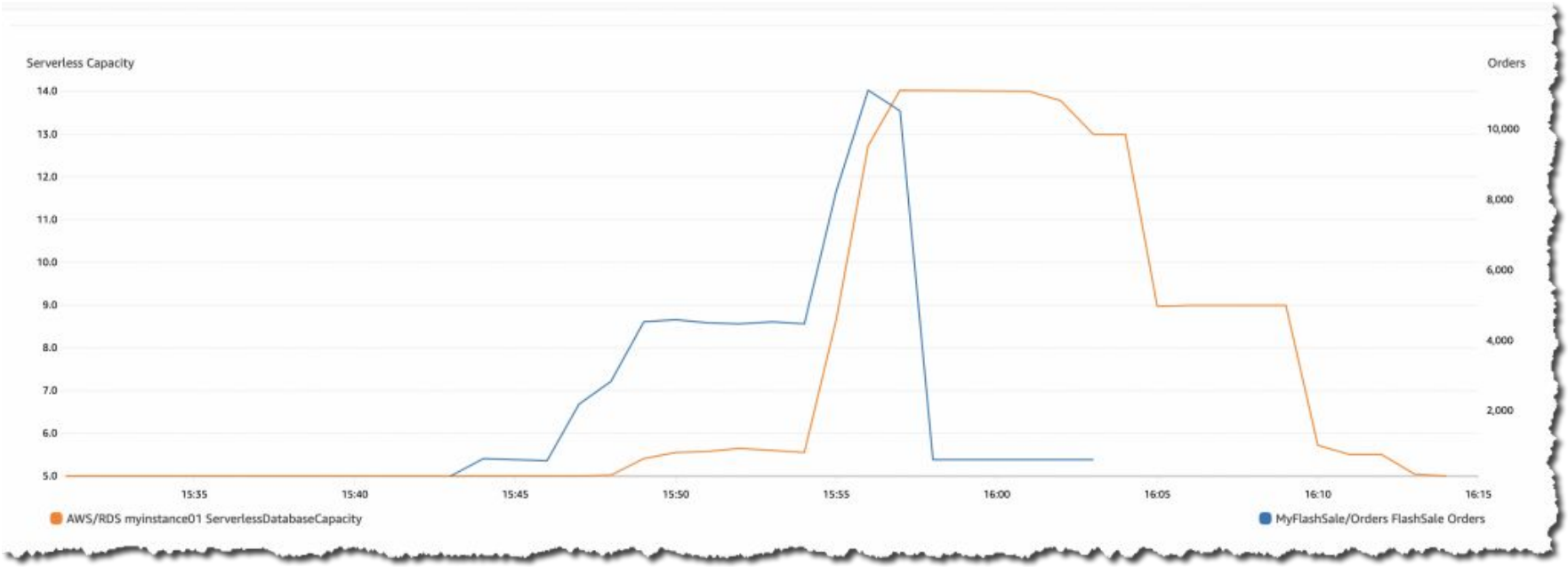
Aurora Serverless v2: Read-write scaling

INSTANTANEOUS SCALING IN FINER GRANULARITY

- Router fleet holds the connections from application
- Database capacity is scaled up from zero to as low as 0.5 Aurora Capacity Units (ACU)
  - 1 ACU has approximately 2 GiB of memory with corresponding CPU and networking
- As more load increases based on CPU / memory usage, required ACUs are instantly available to the database
  - Smallest increment of ACU addition is 0.5 ACU, and the maximum supported is 256 ACUs
- As load vanishes, database capacity is scaled down to zero but with connections still held active with router and storage available for immediate re-attach



Sale Event E-commerce Site



https://aws.amazon.com/blogs/aws/amazon-aurora-serverless-v2-is-generally-available-instant-scaling-for-demanding-workloads/

Introducing the next version of Amazon Aurora Serverless in preview

Posted On: Dec 1, 2020

Aurora Serverless v2 scales to hundreds of thousands of transactions in a fraction of a second, delivering up to 90% cost savings compared to provisioning for peak capacity.

Aurora Serverless is an on-demand, auto-scaling configuration for Amazon Aurora. It automatically starts up, shuts down, and scales capacity up or down based on your application's needs. Amazon Aurora Serverless v2 provides the ability to scale database workloads to hundreds of thousands of transactions in a fraction of a second. Instead of doubling capacity every time a workload needs to scale, it adjusts capacity in fine-grained increments to provide just the right amount of database resources for an application's needs. There is no database capacity for you to manage, you pay only for the capacity your application consumes, and you can save up to 90% of your database cost compared to the cost of provisioning capacity for peak load.

Amazon Aurora Serverless v2 also provides the full breadth of Amazon Aurora's capabilities, including Multi-AZ support, Global Database, and read replicas. Amazon Aurora Serverless is now ideal for a much broader set of applications. For example, Amazon Aurora Serverless can now support enterprises that have hundreds of thousands of applications and want to manage database capacity across the entire fleet and Software as a Service (SaaS) vendors that have multi-tenant environments with hundreds or thousands of databases that each support a different customer. To learn more, visit the Aurora Serverless product page and read the documentation.

Amazon Aurora Serverless v2 is available in preview today for the MySQL 5.7-compatible edition of Amazon Aurora.

Fill out the sign-up form to apply for access to the preview.

Amazon Aurora Serverless v2 is generally available

Posted On: Apr 21, 2022

Amazon Aurora Serverless v2, the next version of Aurora Serverless, is now generally available. Aurora Serverless v2 scales instantly to support even the most demanding applications, delivering up to 90% cost savings compared to provisioning for peak capacity.

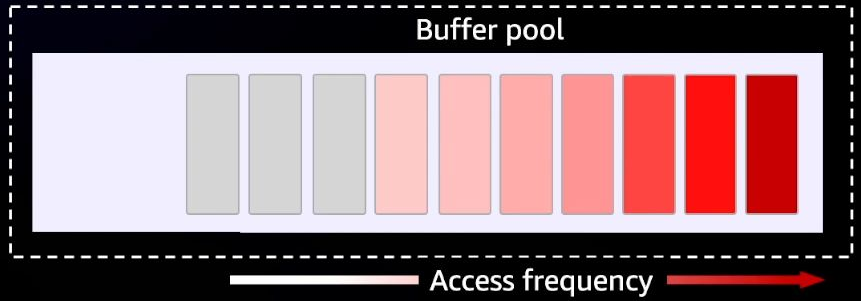
Aurora Serverless is an on-demand, automatic scaling configuration for Amazon Aurora. Aurora Serverless v2 scales database workloads to hundreds of thousands of transactions in a fraction of a second. It adjusts capacity in fine-grained increments to provide just the right amount of database resources for an application's needs. You don't need to manage database capacity, and you pay for only the resources consumed by your application.

Aurora Serverless v2 provides the full breadth of Amazon Aurora capabilities, including Multi-AZ support, Global Database, RDS Proxy, and read replicas. Amazon Aurora Serverless v2 is ideal for a broad set of applications. For example, enterprises that have hundreds of thousands of applications, or software as a service (SaaS) vendors that have multi-tenant environments with hundreds or thousands of databases, can use Aurora Serverless v2 to manage database capacity across the entire fleet.

Aurora Serverless v2 is available for the MySQL 8.0- and PostgreSQL 13-compatible editions of Amazon Aurora. For pricing details and Region availability, visit Amazon Aurora Pricing. To learn more, read the documentation, and get started by creating an Aurora Serverless v2 database using only a few steps in the AWS Management Console.

Apr 27, 2022: *This post has been updated to reflect support of Amazon RDS Proxy.*

Buffer pool resizing



Feature	Aurora Serverless v2 scaling and high availability behavior	Aurora Serverless v1 scaling and high availability behavior
Minimum Aurora capacity units (ACUs) (Aurora MySQL)	0.5	1 when the cluster is running, 0 when the cluster is paused.
Minimum ACUs (Aurora PostgreSQL)	0.5	2 when the cluster is running, 0 when the cluster is paused.
Maximum ACUs (Aurora MySQL)	128	256
Maximum ACUs (Aurora PostgreSQL)	128	384
Stopping a cluster	You can manually stop and start the cluster by using the same cluster stop and start feature as provisioned clusters.	The cluster pauses automatically after a timeout. It takes some time to become available when activity resumes.
Scaling for DB instances	Scale up and down with minimum increment of 0.5 ACUs.	Scale up and down by doubling or halving the ACUs.
Number of DB instances	Same as a provisioned cluster: 1 writer DB instance, up to 15 reader DB instances.	1 DB instance handling both reads and writes.
Scaling can happen while SQL statements are running?	Yes. Aurora Serverless v2 doesn't require waiting for a quiet point.	No. For example, scaling waits for completion of long-running transactions, temporary tables, and table locks.
Reader DB instances scale along with writer	Optional.	Not applicable.
Maximum storage	128 TiB	128 TiB or 64 TiB, depending on database engine and version.
Buffer cache preserved when scaling	Yes. Buffer cache is resized dynamically.	No. Buffer cache is rewarmed after scaling.
Failover	Yes, same as for provisioned clusters.	Best effort only, subject to capacity availability. Slower than in Aurora Serverless v2.
Multi-AZ capability	Yes, same as for provisioned. A Multi-AZ cluster requires a reader DB instance in a second Availability Zone (AZ). For a Multi-AZ cluster, Aurora performs Multi-AZ failover in case of an AZ failure.	Aurora Serverless v1 clusters run all their compute in a single AZ. Recovery in case of AZ failure is best effort only and subject to capacity availability.
Aurora global databases	Yes	No
Scaling based on memory pressure	Yes	No
Scaling based on CPU load	Yes	Yes
Scaling based on network traffic	Yes, based on memory and CPU overhead of network traffic. The `max_connections` parameter remains constant to avoid dropping connections	Yes, based on number of connections.

https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/aurora-serverless-v2.upgrade.html#Serverless.v1-v2-requirements

AWS CLI - Aurora Serverless v2

```
aws rds create-db-instance --db-instance-identifier benchmark
--region eu-central-1 --db-cluster-identifier benchmark --engine
aurora-mysql --engine-version 8.0.mysql_aurora.3.02.0
--serverless-v2-scaling-configuration MinCapacity=1,
MaxCapacity=4 --master-username myuser --master-user-password
mypassword --back-retention 1
```

< demo >

"If the database still doesn't scale down to the minimum capacity configured, then **stop and restart** the database to reclaim any memory fragments that might have built up over time. Stopping and starting a database results in downtime, so we **recommend doing this sparingly**."

Aurora Serverless v2 (AWS official doc)

Q&A

session #3

DynamoDB

Fully managed NoSQL database service

Provisioned read/write capacity

Autoscaling read/write capacity

On-demand read/write capacity

DynamoDB on-demand

No Capacity Planning and Pay-Per-Request Pricing

No changes to your code

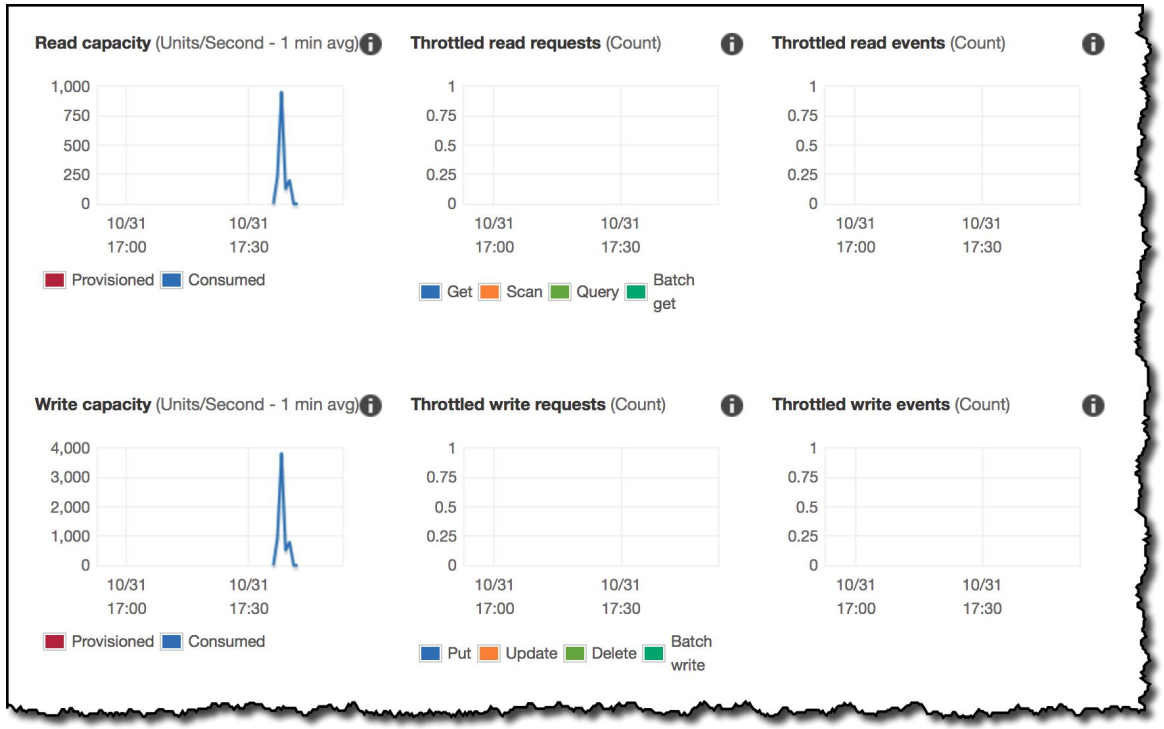
New applications, workloads complex to forecast

Serverless stacks with pay-per-use pricing

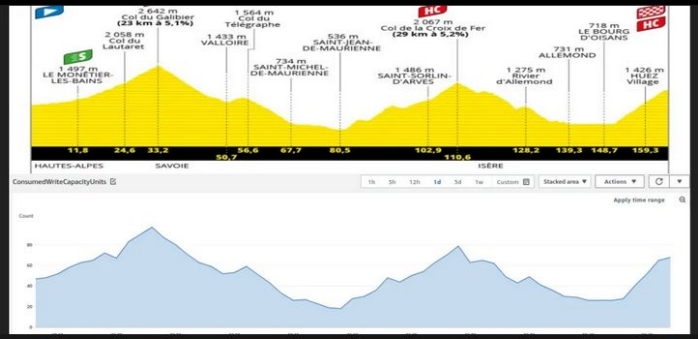
API requests

Single Table Design

Scaling of the Serverless Stack



https://aws.amazon.com/blogs/aws/amazon-aurora-serverless-v2-is-generally-available-instant-scaling-for-demanding-workloads/



Renato Losio for AWS Heroes
Posted on Sep 18, 2022 • Originally published at cloudiamo.com

Edit Manage Stats

4 3

An Absurd Way to Try Amazon DynamoDB On-Demand

#beginners #aws #database #cloud

There are different ways to learn and test Amazon DynamoDB on-demand and test how the database processes requests without capacity planning. There are conventional ones, where you follow the AWS documentation and *"start 1,000 Lambda functions in parallel to generate load on the API*

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

Upload from

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

putItemDynamoDB

lambda_function.py

lambda_function × Execution results ×

```python
import boto3
import random

client = boto3.client('dynamodb')
def lambda_handler(event, context):


  r = random.randint(0,10000);

  data = client.put_item(

    TableName='renato-workshop',
    Item={
        'userid': {
          'S': 'workshop-' + str(r)
        },
        'date': {
          'N': str(r)
        },
        'name': {
          'S': 'Yeezys'
        }
    }
  )

  response = {
      'statusCode': 200,
      'body': 'successfully created item: ' + str(r),
      'headers': {
        'Content-Type': 'application/json',
        'Access-Control-Allow-Origin': '*'
      },
  }
```

26:15 Python Spaces: 2

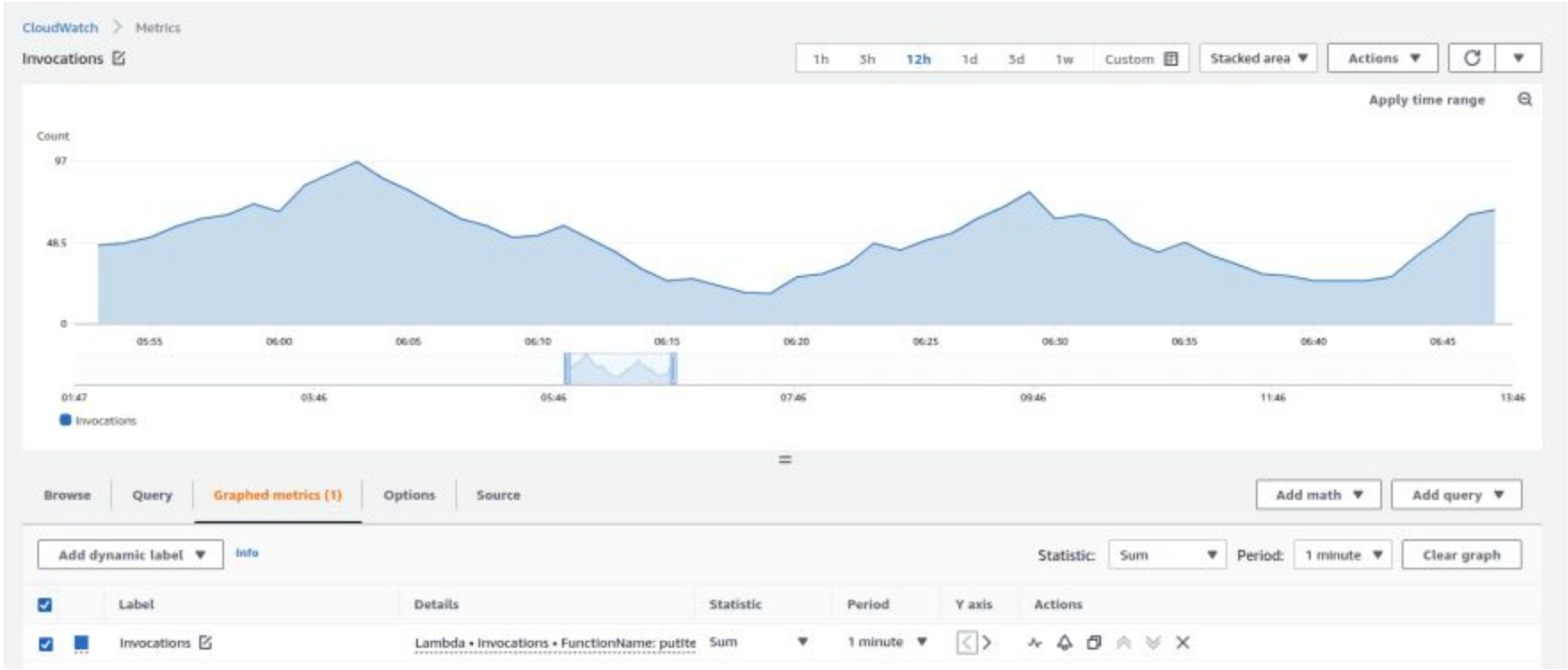
Code properties

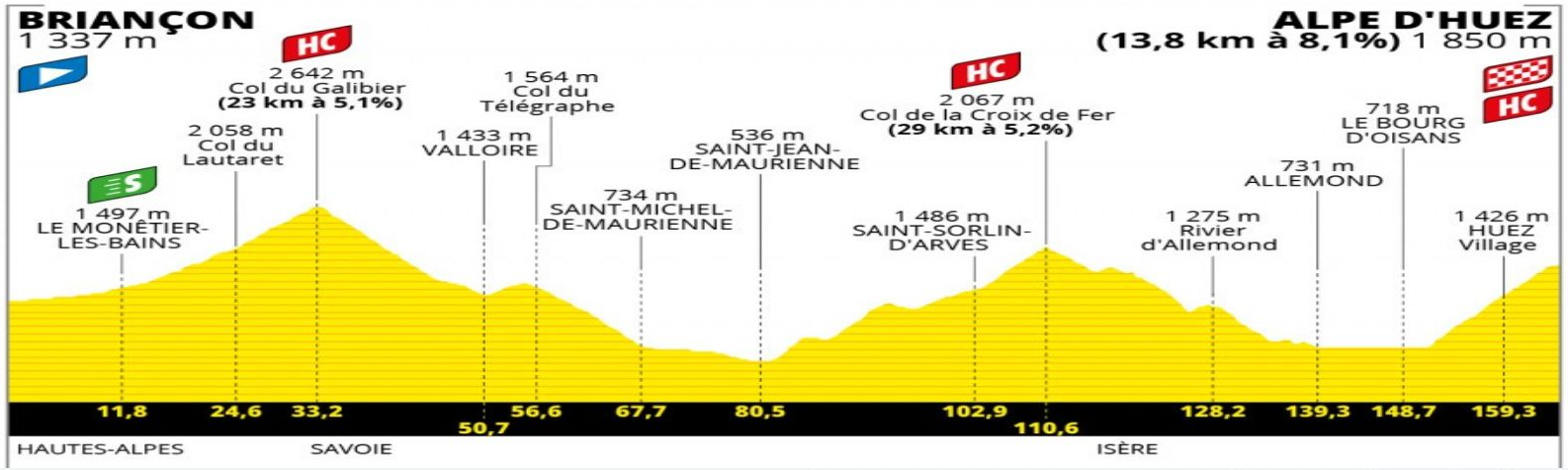
Package size	SHA256 hash	Last modified
470.0 byte	ThMMs3Jy630Jl0HuZfGHQHolhb520IrAvXPs8p6MzVQ=	September 15, 2022 at 08:33 AM GMT+2

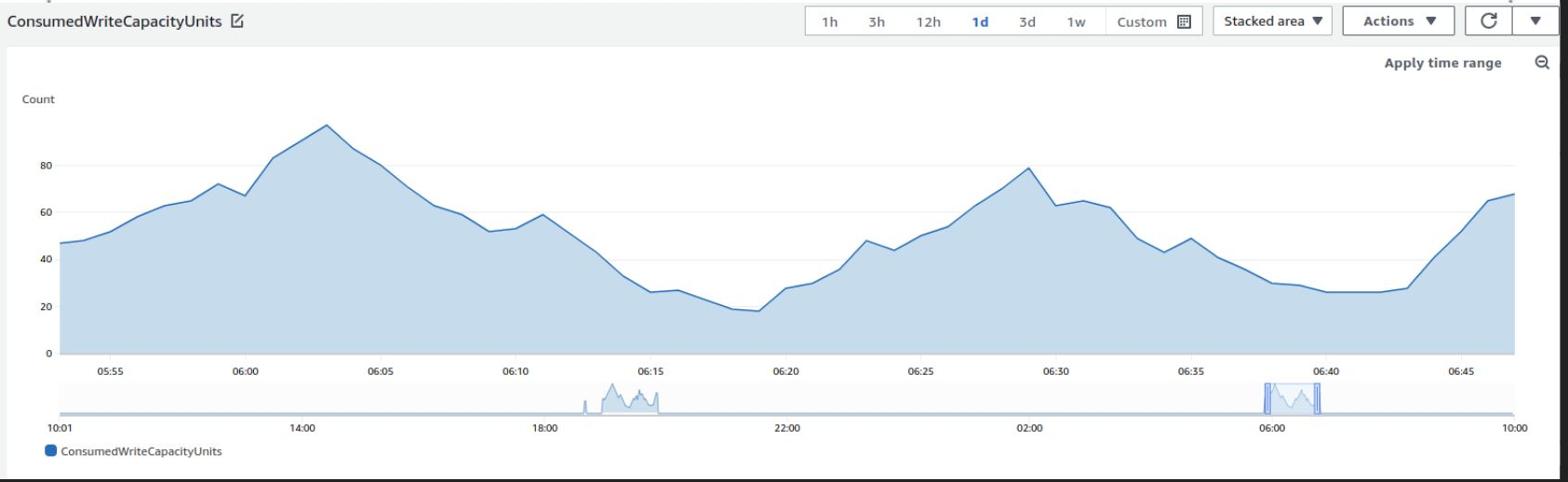
Runtime settings Info

Actions Edit

Runtime Handler Info Architecture Info


CloudWatch
Metrics
Invocations
1h
3h
12h
1d
3d
1w
Custom
Stacked area
Actions
Apply time range
Count
97
48.5
0
05:55
06:00
06:05
06:10
06:15
06:20
06:25
06:30
06:35
06:40
06:45
01:47
03:46
05:46
07:46
09:46
11:46
13:46
Invocations
Browse
Query
Graphed metrics (1)
Options
Source
Add math
Add query
Add dynamic label
Info
Statistic:
Sum
Period:
1 minute
Clear graph
Label
Details
Statistic
Period
Y axis
Actions
Invocations
Lambda • Invocations • FunctionName: putIte
Sum
1 minute


BRIANÇON
1 337 m
ALPE D'HUEZ
(13,8 km à 8,1%) 1 850 m
HC
2 642 m
Col du Galibier
(23 km à 5,1%)
2 058 m
Col du
Lautaret
1 564 m
Col du
Télégraphe
1 433 m
VALLOIRE
536 m
SAINT-JEAN-
DE-MAURIENNE
734 m
SAINT-MICHEL-
DE-MAURIENNE
HC
2 067 m
Col de la Croix de Fer
(29 km à 5,2%)
718 m
LE BOURG
D'OISANS
731 m
ALLEMOND
1 497 m
LE MONÊTIER-
LES-BAINS
1 486 m
SAINT-SORLIN-
D'ARVES
1 275 m
Rivier
d'Allemond
1 426 m
HUEZ
Village
11,8
24,6
33,2
50,7
56,6
67,7
80,5
102,9
110,6
128,2
139,3
148,7
159,3
HAUTES-ALPES
SAVOIE
ISÈRE


ConsumedWriteCapacityUnits
1h
3h
12h
1d
3d
1w
Custom
Stacked area
Actions
Apply time range
Count
80
60
40
20
0
05:55
06:00
06:05
06:10
06:15
06:20
06:25
06:30
06:35
06:40
06:45
10:01
14:00
18:00
22:00
02:00
06:00
10:00
ConsumedWriteCapacityUnits

AWS CLI - Dynamo DB

```
aws dynamodb create-table \
--table-name MusicCollection \
--attribute-definitions AttributeName=Artist,AttributeType=S
AttributeName=SongTitle,AttirbuteType=S \
--key-schema AttributeName=Artist,KeyType=HASH
AttributeName=SongTitle,KeyType=RANGE \
--billing-mode PAY_PER_REQUEST
```

< demo >

NoSQL vs SQL

elasticity?

Storage is cheap, CPU is expensive

RDBMS is not deprecated by NoSQL

NoSQL != not relational

NoSQL great fit OLTP

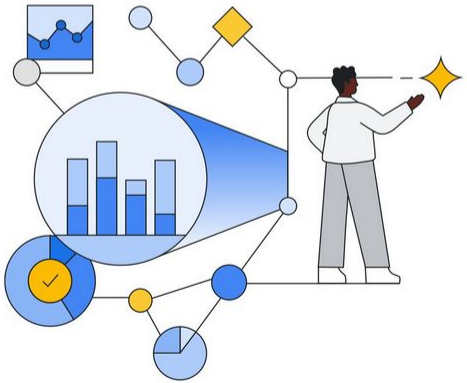
RDBMS high velocity OLAP

NewSQL: scalability for OLTP + ACID

Databases

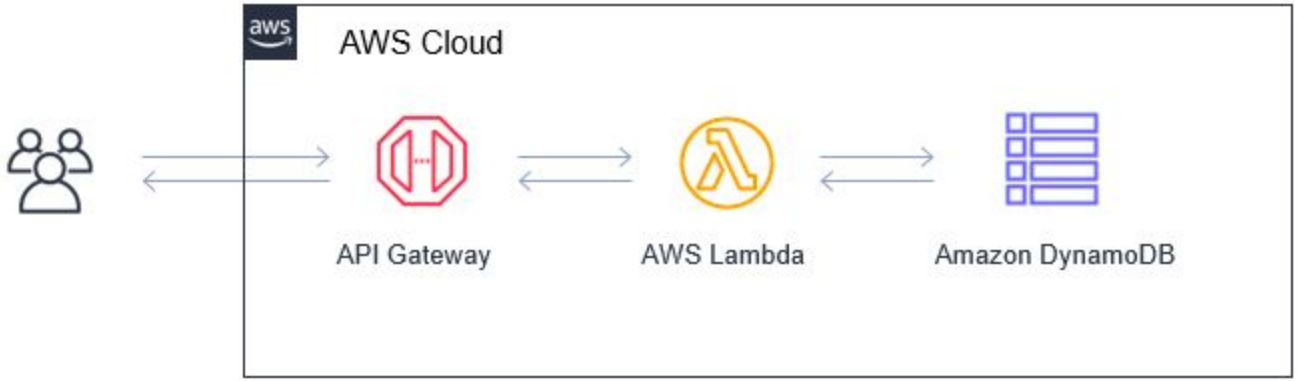
IT prediction: Barriers between transactional and analytical workloads fall away

December 20, 2022

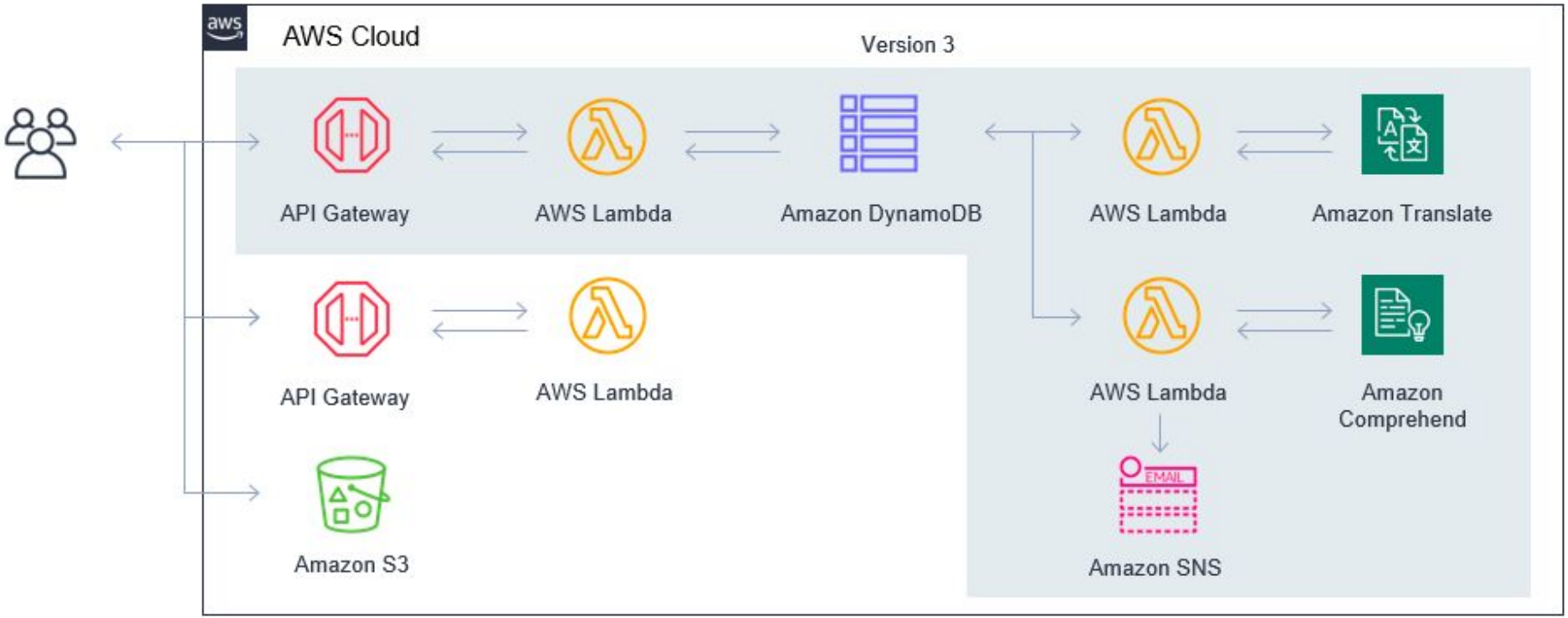


https://cloud.google.com/blog/products/databases/transactional-and-analytical-workloads-unite

Modeling business logic flows in serverless applications: Storing customer reviews



https://aws.amazon.com/blogs/compute/modeling-business-logic-flows-in-serverless-applications/


AWS Cloud
Version 3
API Gateway
AWS Lambda
Amazon DynamoDB
AWS Lambda
Amazon Translate
API Gateway
AWS Lambda
AWS Lambda
Amazon Comprehend
EMAIL
Amazon S3
Amazon SNS

Q&A

session #4

Athena + S3

Serverless Interactive Query Service

- Analyzing data on S3 using SQL
- Pay only for the query you run
- Model database (CSV, JSON*, etc)
- Integrated with AWS Glue Data Catalog

Amazon Athena announces upgraded query engine

Posted On: Oct 13, 2022

Amazon Athena has upgraded its SQL query engine to include the latest features from the Trino open source project. Athena engine version 3 includes over 50 new SQL functions, 30 new features, and more than 90 query performance improvements. With today's launch, Athena is also introducing a continuous integration approach to open source software management that will improve currency with the Trino and Presto projects so that you get faster access to community improvements, integrated and tuned within the Athena engine.

Athena engine version 3 includes all the features of version 2 while bringing numerous enhancements, such as T-Digest functions that can be used to approximate rank-based statistics with high accuracy, new Geospatial functions to run optimized Geospatial queries, and new query syntaxes such as MATCH_RECOGNIZE for identifying data patterns in applications such as fraud detection and sensor data analysis. With our simplified engine upgrade process, you can configure existing workgroups to be automatically upgraded to engine version 3 without requiring manual review or intervention.

Start using the new Athena engine today by creating or configuring a workgroup and selecting the recommended "Athena Engine Version 3". For more details about selecting engine versions, see Changing Athena engine versions. To learn more about the features of Athena engine version 3, see Athena engine version 3 reference and Upgrade to Athena engine version 3 to increase query performance and access more analytics features.

Athena engine version 3 is available today in all regions supporting Athena except for AWS China (Beijing) region, operated by Sinnet and AWS China (Ningxia) region, operated by NWCD.

Learn how to use AWS re:Post's new question subfilters!

AWS re:Post recently launched new question subfilters as part of the enhanced search experience. Read this article to learn how to optimize your time on re:Post whether you're looking for answers or looking to contribute knowledge.

How do I analyze my Amazon S3 server access logs using Athena?

6 minute read

1

I want to query Amazon Simple Storage Service (Amazon S3) server access logs in Amazon Athena.

Resolution

Amazon S3 stores server access logs as objects in an S3 bucket. Use Athena to quickly analyze and query server access logs.

1. Turn on server access logging for your S3 bucket, if you haven't already. Note the values for **Target bucket** and **Target prefix**. You need both to specify the Amazon S3 location in an Athena query.

2. Open the Amazon Athena console. You may need to perform the Athena first time setup before running your first query.

3. In the **Query editor**, run a DDL statement to create a database. It's a best practice to create the database in the same AWS Region as your S3 bucket.

```
create database s3_access_logs_db
```

4. Create a table schema in the database. In the following example, the **STRING** and **BIGINT** data type values are the access log properties. You can query these properties in Athena. For **LOCATION**, enter the S3 bucket and prefix path from Step 1. Make sure to include a forward slash (/) at the end of the prefix (for example, **s3://doc-example-bucket/prefix/**). If you're not using a prefix, then

Topics

Analytics

Tags

Amazon Athena

Language

English

Related videos

Watch Bukola's video to learn more (5:43)

AWS OFFICIAL

Updated 4 months ago

FEEDBACK

https://repost.aws/knowledge-center/analyze-logs-athena

< demo >

Q&A

recap

Open Source?

AWS Open Source Blog

Behind the Scenes on AWS Contributions to Open Source Databases

by Libby Clark, Colleen Betik, and Madelyn Olson | on 04 OCT 2023 | in Amazon RDS, Customer Solutions, Open Source, RDS For MariaDB, RDS For MySQL, RDS For PostgreSQL | Permalink | Comments | Share

AWS managed open source services make it easier for customers to set up and operate their favorite open source projects on AWS. We help reduce the overhead of self-managing open source and provide better integrations with AWS services. In the process, we hire maintainers or develop engineers to become experts and leaders in the open source projects that

Resources

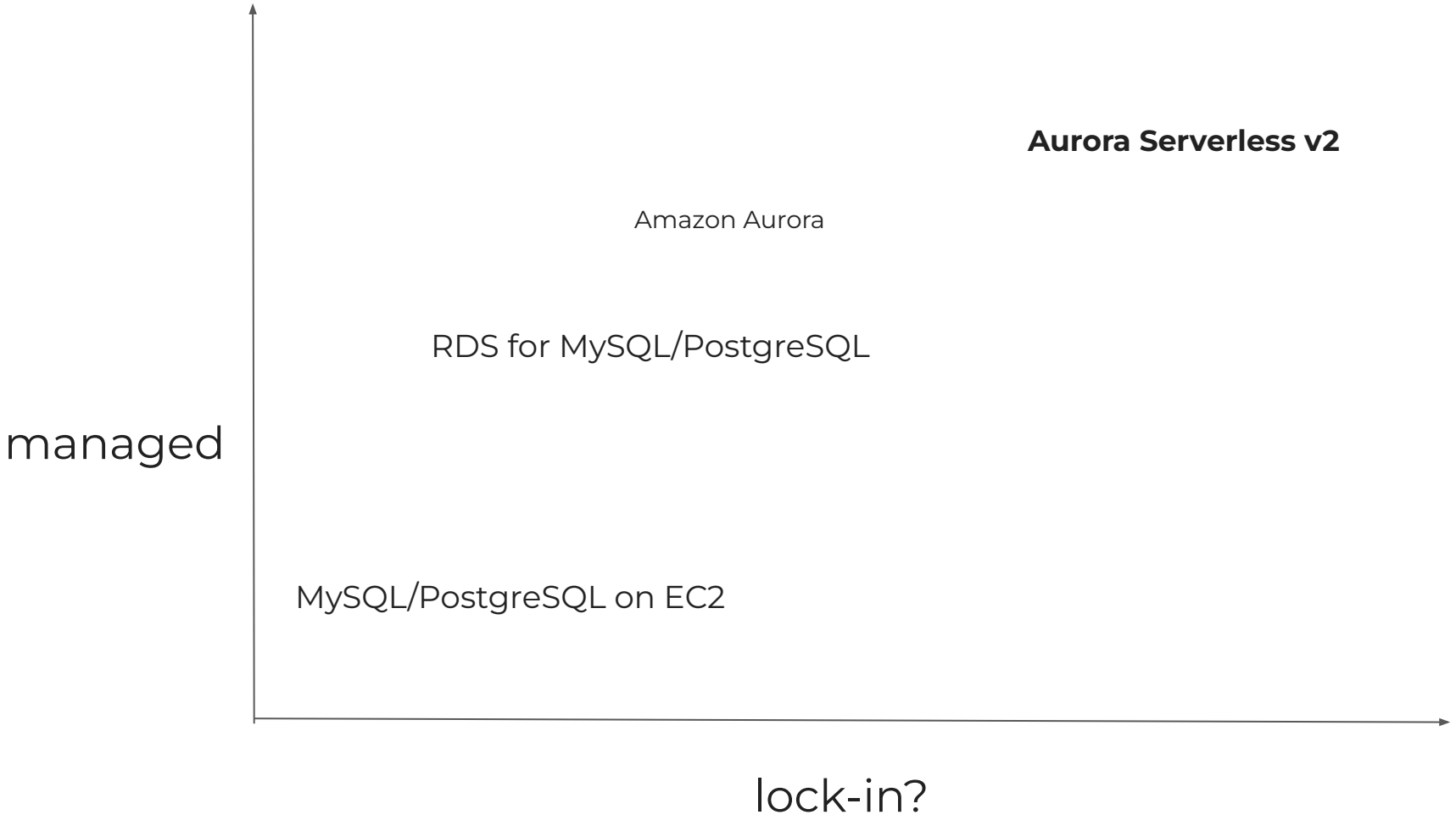
Open Source at AWS
Projects on GitHub

Follow

AWS Open Source Twitter
Facebook
LinkedIn
Twitch
Open Source RSS Feed
Email Updates

https://aws.amazon.com/blogs/opensource/behind-the-scenes-on-aws-contributions-to-open-source-databases/


NO


Aurora Serverless v2
Amazon Aurora
RDS for MySQL/PostgreSQL
managed
MySQL/PostgreSQL on EC2
lock-in?

Costs?


THE UNITED STATES OF AMERICA
ONE DOLLAR
THIS NOTE IS LEGAL TENDER FOR ALL DEBTS, PUBLIC AND PRIVATE
WASHINGTON, D.C.

However, **no system provides 100% availability**, so the pragmatic question is whether or not Spanner delivers availability that is so high that most users don't worry about its outages. For example, given there are many sources of outages for an application, if Spanner is an insignificant contributor to its downtime, then users are correct to not worry about it.

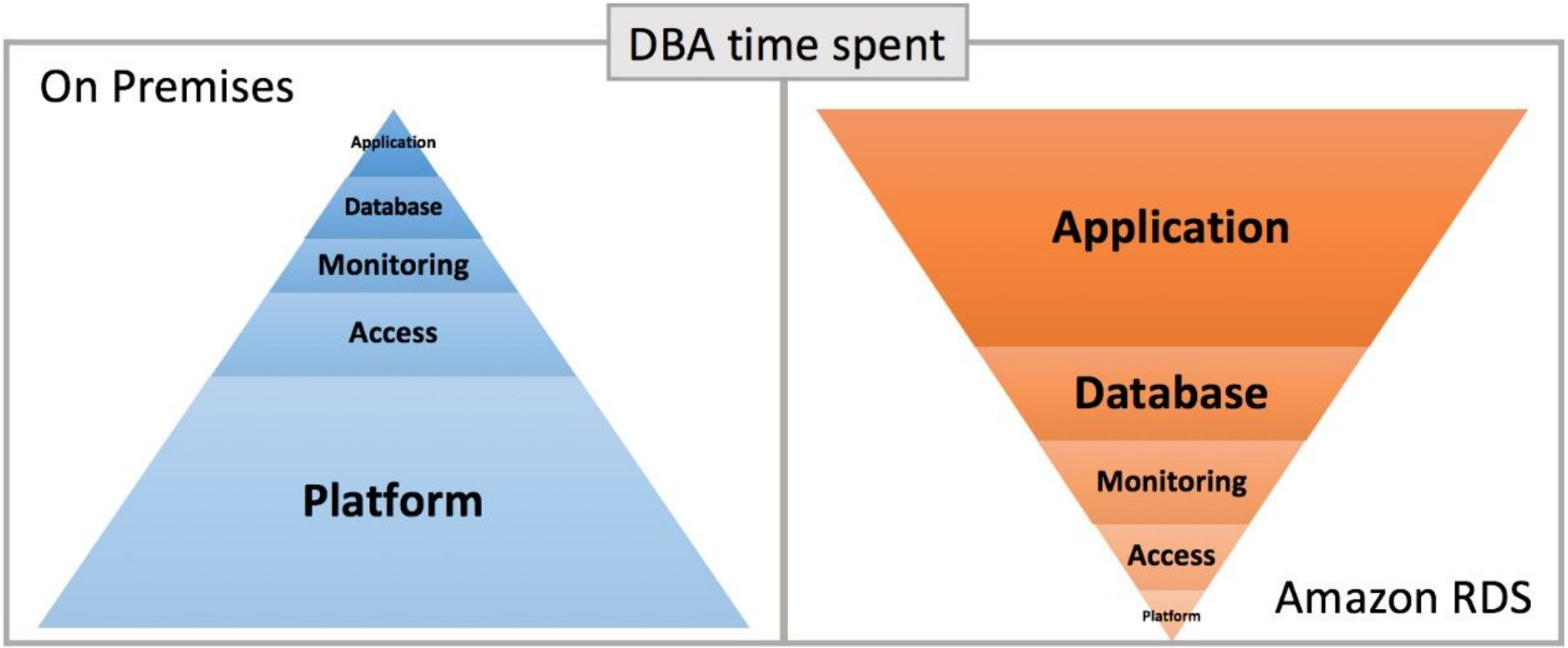
https://cloud.google.com/blog/products/databases/inside-cloud-spanner-and-the-cap-theorem

One single server?

One provisioned managed server?

A serverless database?

A k8s deployment?



https://aws.amazon.com/blogs/database/part-1-role-of-the-dba-when-moving-to-amazon-rds-responsibilities/

all's well that ends well?

Default
configuration?

Databases are
complex beasts

CPU or
Storage?

Conference effect (**99%**)

Aim for **good enough**

Peter Zaitsev • 1st
Founder, CEO | Driving Success with MySQL, MariaDB, MongoDB & PostgreS...
1w

Auto Scaling can convert some performance bugs driven downtime to large bills. This does not always apply to Databases where bad query is often so bad no auto scaling can make it tolerable

Thank You!

cloudiamo.com