MOKGADI SELEPE

Introduction to Amazon Aurora

Here's what happened:

You've learned about some Amazon technologies:

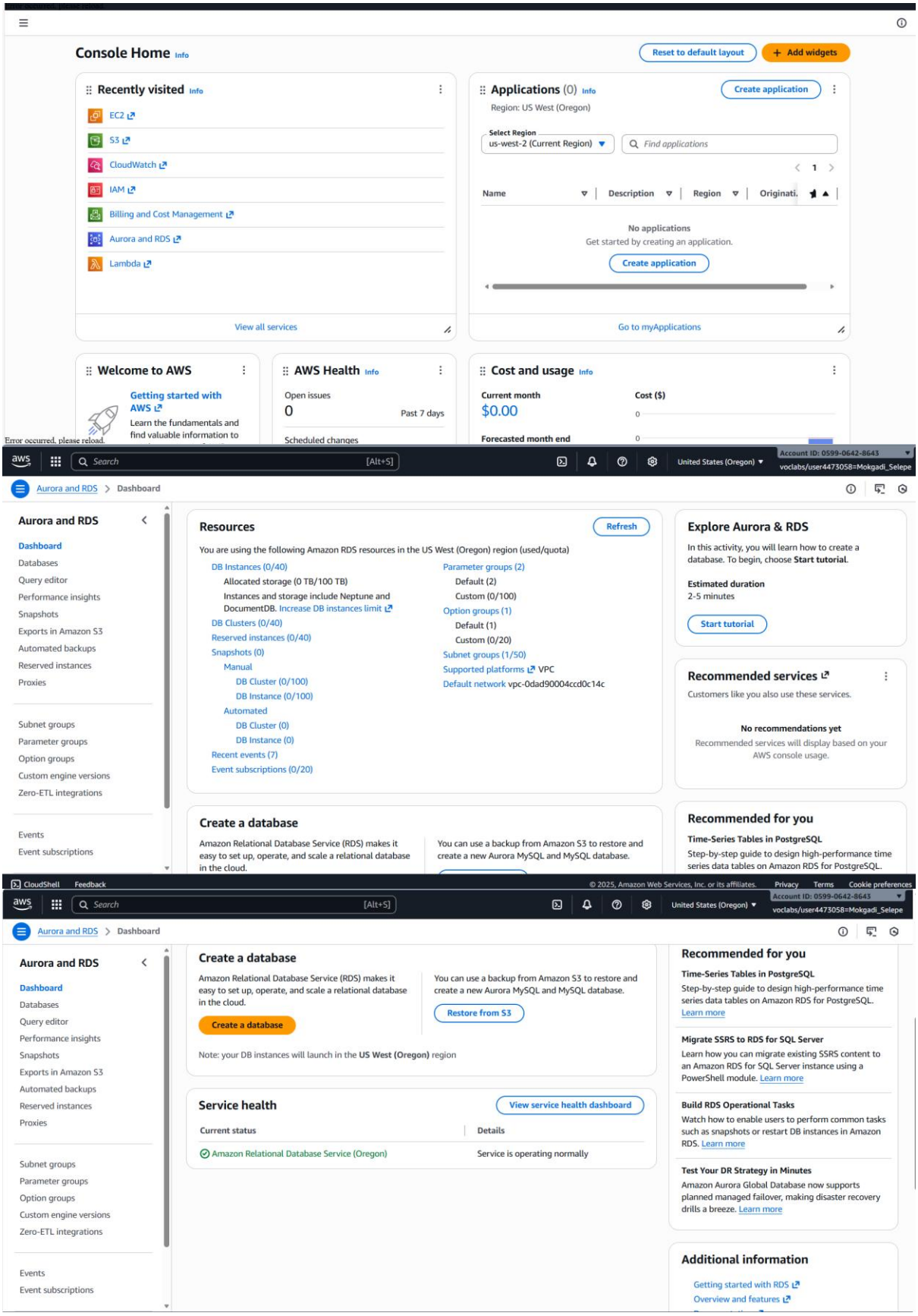
- Amazon Aurora: A fast and reliable database engine that's similar to MySQL, but more powerful and cost-effective.
- Amazon EC2: A service that lets you quickly create and manage virtual servers in the cloud.
- Amazon RDS: A service that makes it easy to create and manage databases in the cloud, including MySQL, PostgreSQL, and others.

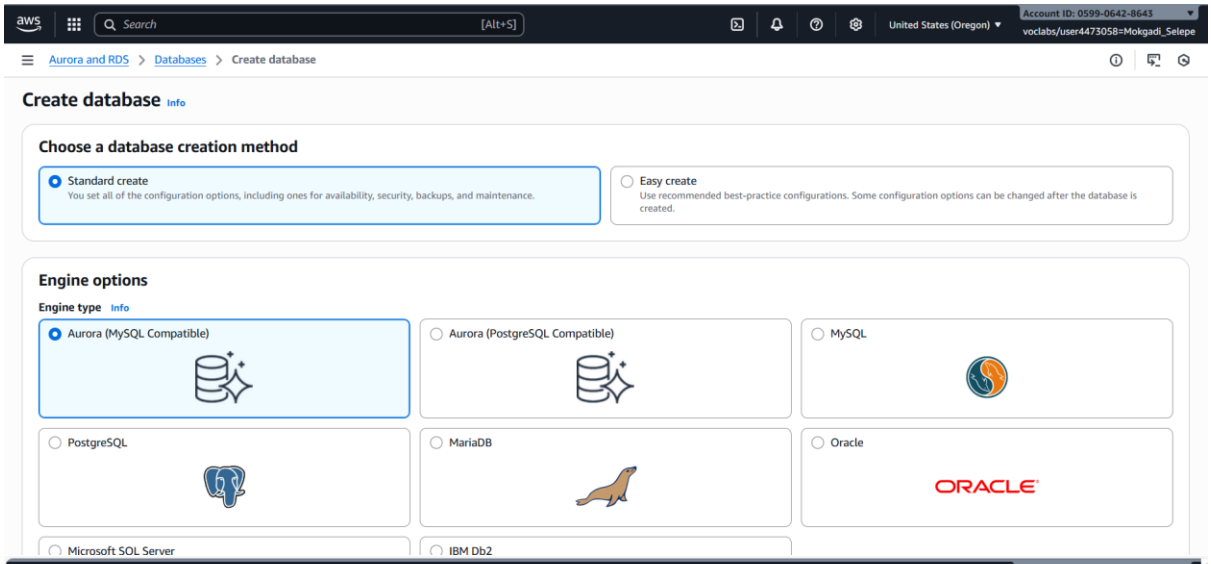
Think of it like this:

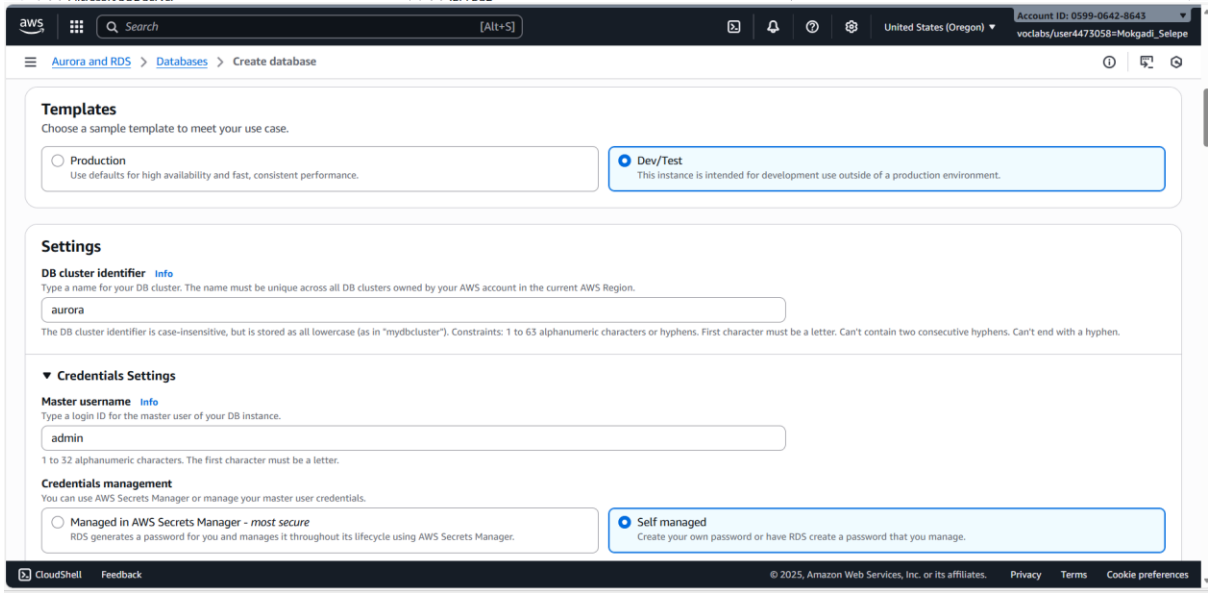
- Aurora is a high-performance database engine.
- EC2 is a virtual server that can run your applications.
- RDS is a service that helps you create and manage databases, including Aurora, with ease!

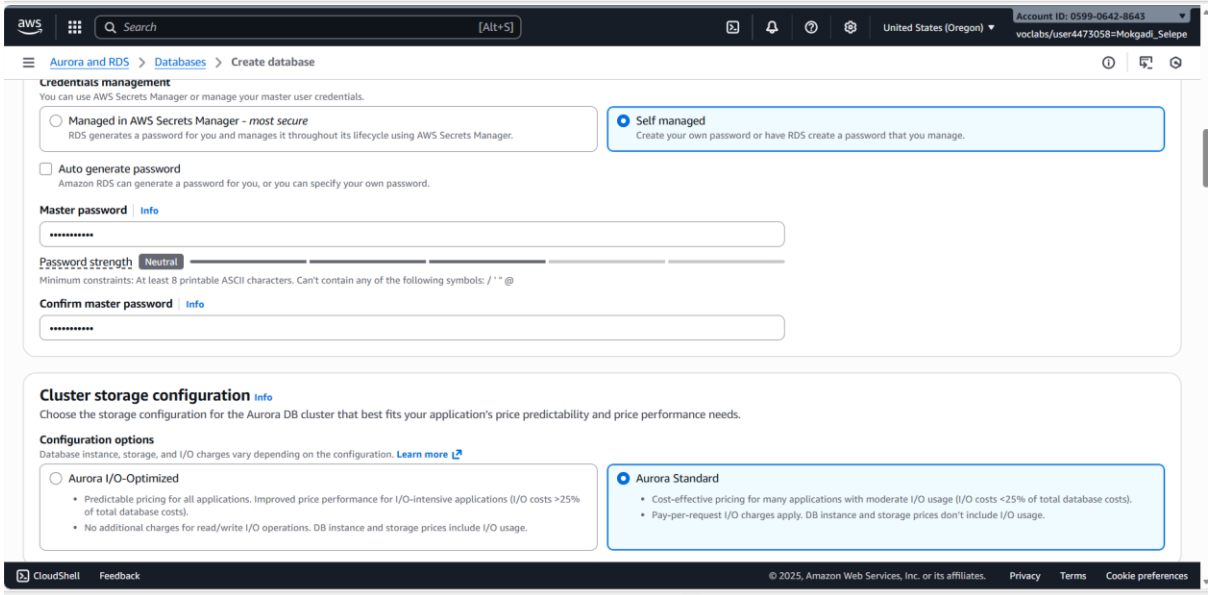
These technologies can help you build and run scalable and reliable applications in the cloud!

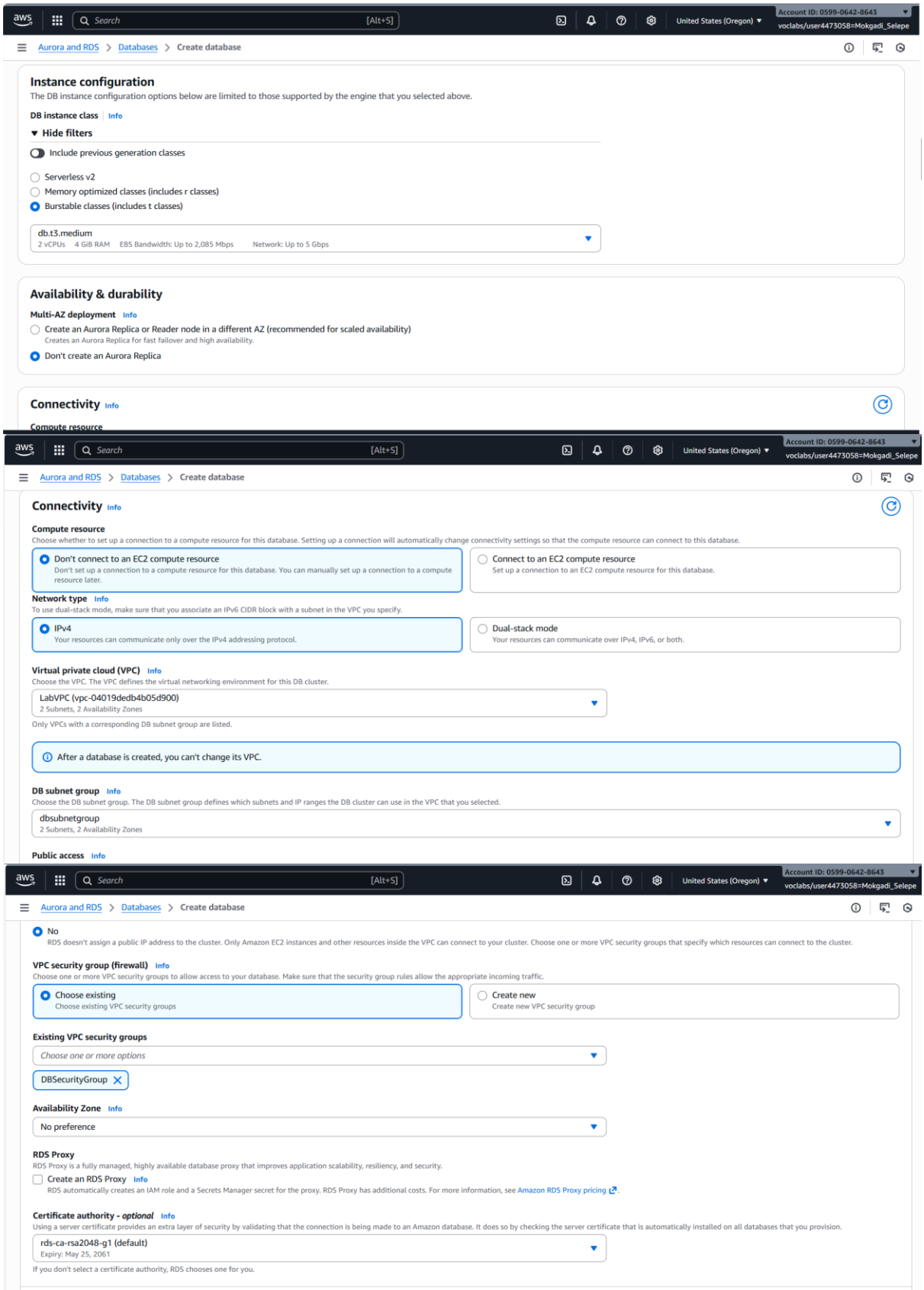
1: Create an Aurora instance

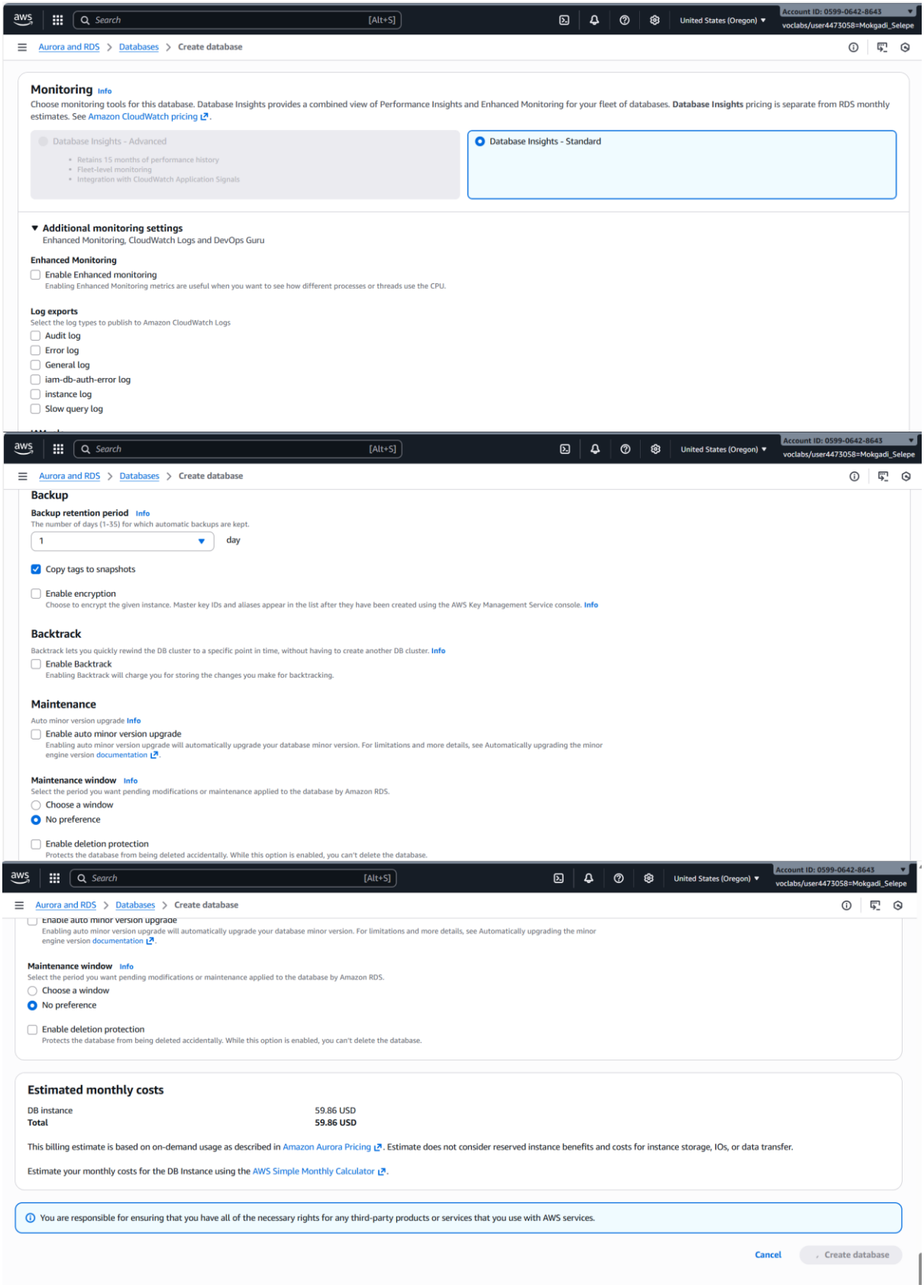

Console Home Info
Reset to default layout
+ Add widgets
Recently visited Info
EC2
S3
CloudWatch
IAM
Billing and Cost Management
Aurora and RDS
Lambda
View all services
Applications (0) Info
Region: US West (Oregon)
Create application
Select Region
us-west-2 (Current Region)
Find applications
Name
Description
Region
Originati.
No applications
Get started by creating an application.
Create application
Go to myApplications
Welcome to AWS
Getting started with AWS
Learn the fundamentals and find valuable information to
AWS Health Info
Open issues
0
Past 7 days
Scheduled changes
Cost and usage Info
Current month
$0.00
Cost ($)
Forecasted month end
Error occurred, please reload.
Search
[Alt+S]
United States (Oregon)
Account ID: 0599-0642-8643
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Dashboard
Aurora and RDS
Dashboard
Databases
Query editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations
Events
Event subscriptions
Resources
Refresh
You are using the following Amazon RDS resources in the US West (Oregon) region (used/quota)
DB Instances (0/40)
Allocated storage (0 TB/100 TB)
Instances and storage include Neptune and DocumentDB. Increase DB instances limit
DB Clusters (0/40)
Reserved instances (0/40)
Snapshots (0)
Manual
DB Cluster (0/100)
DB Instance (0/100)
Automated
DB Cluster (0)
DB Instance (0)
Recent events (7)
Event subscriptions (0/20)
Parameter groups (2)
Default (2)
Custom (0/100)
Option groups (1)
Default (1)
Custom (0/20)
Subnet groups (1/50)
Supported platforms VPC
Default network vpc-0dad90004ccd0c14c
Explore Aurora & RDS
In this activity, you will learn how to create a database. To begin, choose Start tutorial.
Estimated duration
2-5 minutes
Start tutorial
Recommended services
Customers like you also use these services.
No recommendations yet
Recommended services will display based on your AWS console usage.
Create a database
Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.
You can use a backup from Amazon S3 to restore and create a new Aurora MySQL and MySQL database.
Recommended for you
Time-Series Tables in PostgreSQL
Step-by-step guide to design high-performance time series data tables on Amazon RDS for PostgreSQL.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Create a database
Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.
Create a database
You can use a backup from Amazon S3 to restore and create a new Aurora MySQL and MySQL database.
Restore from S3
Note: your DB instances will launch in the US West (Oregon) region
Service health
View service health dashboard
Current status
Details
Amazon Relational Database Service (Oregon)
Service is operating normally
Recommended for you
Time-Series Tables in PostgreSQL
Step-by-step guide to design high-performance time series data tables on Amazon RDS for PostgreSQL.
Learn more
Migrate SSRS to RDS for SQL Server
Learn how you can migrate existing SSRS content to an Amazon RDS for SQL Server instance using a PowerShell module. Learn more
Build RDS Operational Tasks
Watch how to enable users to perform common tasks such as snapshots or restart DB instances in Amazon RDS. Learn more
Test Your DR Strategy in Minutes
Amazon Aurora Global Database now supports planned managed failover, making disaster recovery drills a breeze. Learn more
Additional information
Getting started with RDS
Overview and features

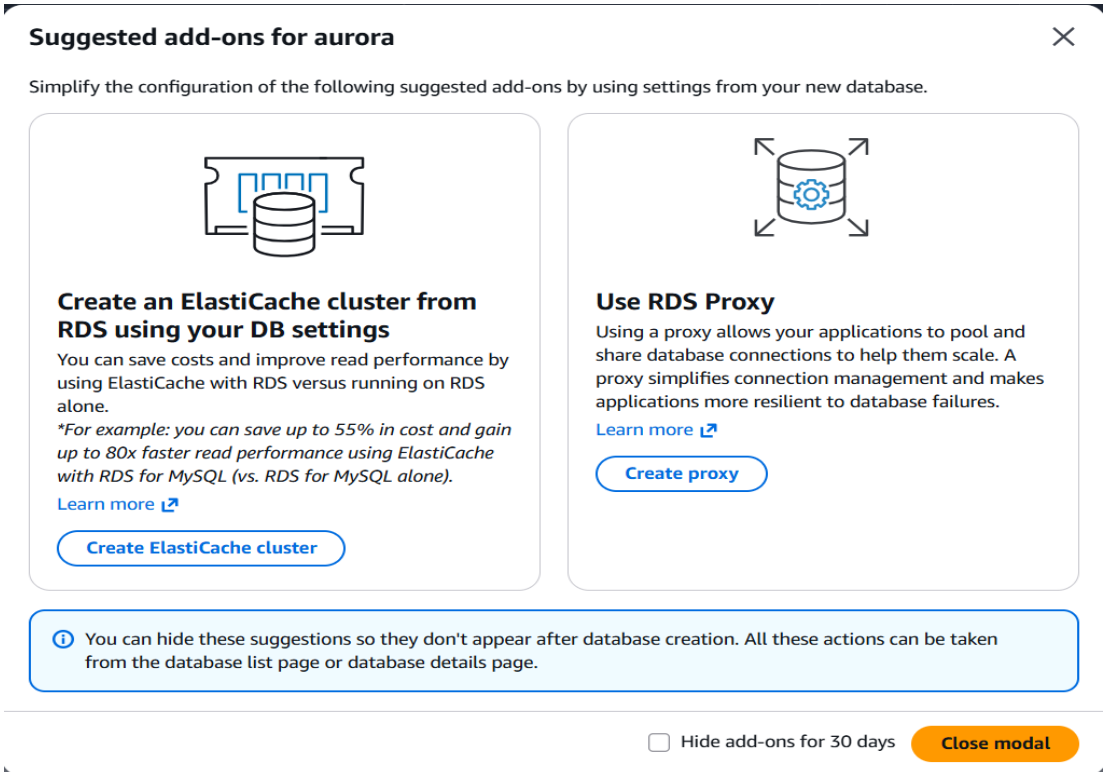

Search [Alt+S]
United States (Oregon)
Account ID: 0599-0642-8643
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Databases > Create database
Create database Info
Choose a database creation method
Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.
Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.
Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
PostgreSQL
MariaDB
Oracle
Microsoft SQL Server
IBM Db2

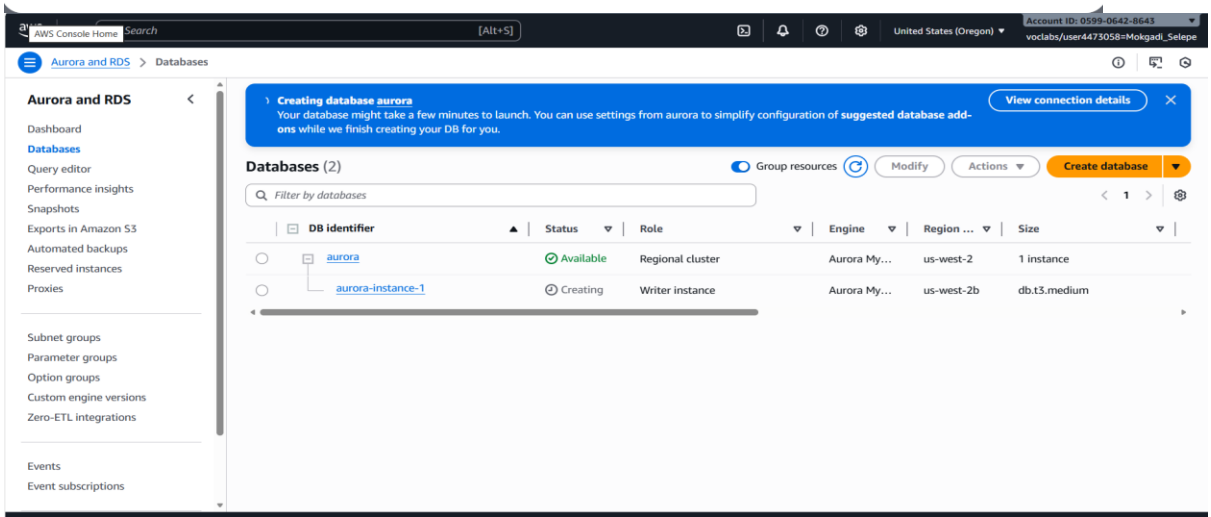

Search [Alt+S]
United States (Oregon)
Account ID: 0599-0642-8643
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Databases > Create database
Templates
Choose a sample template to meet your use case.
Production
Use defaults for high availability and fast, consistent performance.
Dev/Test
This instance is intended for development use outside of a production environment.
Settings
DB cluster identifier Info
Type a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.
aurora
The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.
Credentials Settings
Master username Info
Type a login ID for the master user of your DB instance.
admin
1 to 32 alphanumeric characters. The first character must be a letter.
Credentials management
You can use AWS Secrets Manager or manage your master user credentials.
Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
Self managed
Create your own password or have RDS create a password that you manage.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

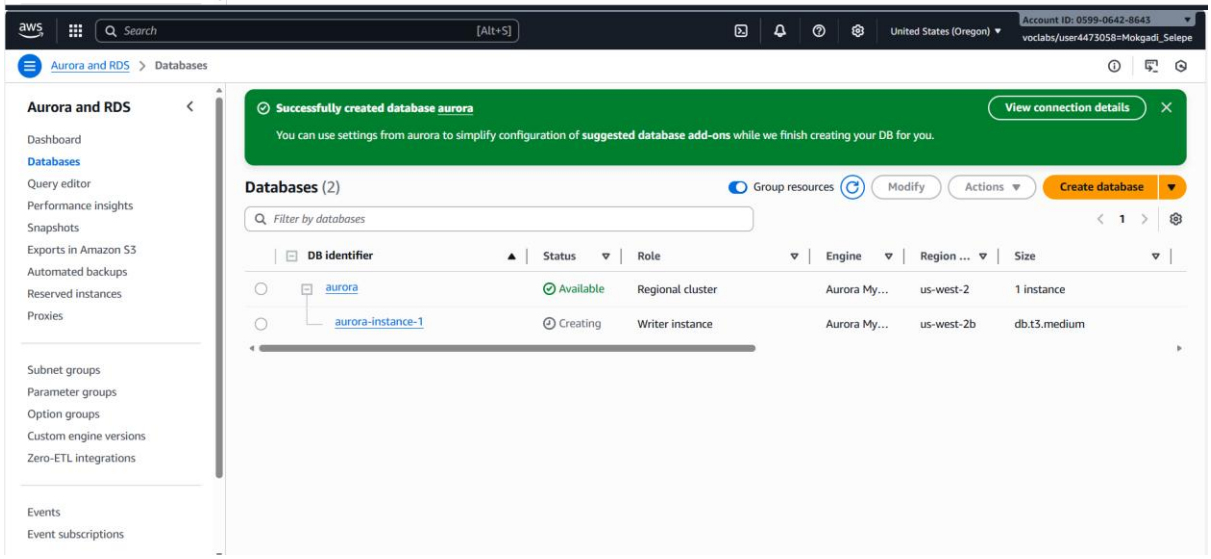

Search [Alt+S]
United States (Oregon)
Account ID: 0599-0642-8643
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Databases > Create database
Credentials management
You can use AWS Secrets Manager or manage your master user credentials.
Managed in AWS Secrets Manager - most secure
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
Self managed
Create your own password or have RDS create a password that you manage.
Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.
Master password Info
Password strength
Neutral
Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @
Confirm master password Info
Cluster storage configuration Info
Choose the storage configuration for the Aurora DB cluster that best fits your application's price predictability and price performance needs.
Configuration options
Database instance, storage, and I/O charges vary depending on the configuration. Learn more
Aurora I/O-Optimized
Predictable pricing for all applications. Improved price performance for I/O-intensive applications (I/O costs >25% of total database costs).
No additional charges for read/write I/O operations. DB instance and storage prices include I/O usage.
Aurora Standard
Cost-effective pricing for many applications with moderate I/O usage (I/O costs <25% of total database costs).
Pay-per-request I/O charges apply. DB instance and storage prices don't include I/O usage.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
United States (Oregon)
Account ID: 0599-0642-8643
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Databases > Create database
Instance configuration
The DB instance configuration options below are limited to those supported by the engine that you selected above.
DB instance class Info
Hide filters
Include previous generation classes
Serverless v2
Memory optimized classes (includes r classes)
Burstable classes (includes t classes)
db.t3.medium
2 vCPUs 4 GiB RAM EBS Bandwidth: Up to 2,085 Mbps Network: Up to 5 Gbps
Availability & durability
Multi-AZ deployment Info
Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)
Creates an Aurora Replica for fast failover and high availability.
Don't create an Aurora Replica
Connectivity Info
Compute resource
Connectivity Info
Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.
Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.
Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.
Network type Info
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.
IPv4
Your resources can communicate only over the IPv4 addressing protocol.
Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.
Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB cluster.
LabVPC (vpc-04019dedb4b05d900)
2 Subnets, 2 Availability Zones
Only VPCs with a corresponding DB subnet group are listed.
After a database is created, you can't change its VPC.
DB subnet group Info
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB cluster can use in the VPC that you selected.
dbsubnetgroup
2 Subnets, 2 Availability Zones
Public access Info
No
RDS doesn't assign a public IP address to the cluster. Only Amazon EC2 instances and other resources inside the VPC can connect to your cluster. Choose one or more VPC security groups that specify which resources can connect to the cluster.
VPC security group (firewall) Info
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.
Choose existing
Choose existing VPC security groups
Create new
Create new VPC security group
Existing VPC security groups
Choose one or more options
DBSecurityGroup
Availability Zone Info
No preference
RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.
Create an RDS Proxy Info
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see Amazon RDS Proxy pricing
Certificate authority - optional Info
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.
rds-ca-rsa2048-g1 (default)
Expiry: May 25, 2061
If you don't select a certificate authority, RDS chooses one for you.


Search [Alt+S]
United States (Oregon)
Account ID: 0599-0642-8643
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Databases > Create database
Monitoring Info
Choose monitoring tools for this database. Database Insights provides a combined view of Performance Insights and Enhanced Monitoring for your fleet of databases. Database Insights pricing is separate from RDS monthly estimates. See Amazon CloudWatch pricing.
Database Insights - Advanced
Retains 15 months of performance history
Fleet-level monitoring
Integration with CloudWatch Application Signals
Database Insights - Standard
Additional monitoring settings
Enhanced Monitoring, CloudWatch Logs and DevOps Guru
Enhanced Monitoring
Enable Enhanced monitoring
Enabling Enhanced Monitoring metrics are useful when you want to see how different processes or threads use the CPU.
Log exports
Select the log types to publish to Amazon CloudWatch Logs
Audit log
Error log
General log
iam-db-auth-error log
instance log
Slow query log
Backup
Backup retention period Info
The number of days (1-35) for which automatic backups are kept.
1
day
Copy tags to snapshots
Enable encryption
Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. Info
Backtrack
Backtrack lets you quickly rewind the DB cluster to a specific point in time, without having to create another DB cluster. Info
Enable Backtrack
Enabling Backtrack will charge you for storing the changes you make for backtracking.
Maintenance
Auto minor version upgrade Info
Enable auto minor version upgrade
Enabling auto minor version upgrade will automatically upgrade your database minor version. For limitations and more details, see Automatically upgrading the minor engine version documentation.
Maintenance window Info
Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.
Choose a window
No preference
Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.
Estimated monthly costs
DB instance 59.86 USD
Total 59.86 USD
This billing estimate is based on on-demand usage as described in Amazon Aurora Pricing. Estimate does not consider reserved instance benefits and costs for instance storage, IOs, or data transfer.
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator.
You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.
Cancel
Create database


Suggested add-ons for aurora
Simplify the configuration of the following suggested add-ons by using settings from your new database.
Create an ElastiCache cluster from RDS using your DB settings
You can save costs and improve read performance by using ElastiCache with RDS versus running on RDS alone.
*For example: you can save up to 55% in cost and gain up to 80x faster read performance using ElastiCache with RDS for MySQL (vs. RDS for MySQL alone).
Learn more
Create ElastiCache cluster
Use RDS Proxy
Using a proxy allows your applications to pool and share database connections to help them scale. A proxy simplifies connection management and makes applications more resilient to database failures.
Learn more
Create proxy
You can hide these suggestions so they don't appear after database creation. All these actions can be taken from the database list page or database details page.
Hide add-ons for 30 days
Close modal


Aurora and RDS
Databases
Creating database aurora
Your database might take a few minutes to launch. You can use settings from aurora to simplify configuration of suggested database add-ons while we finish creating your DB for you.
View connection details
Databases (2)
Group resources
Modify
Actions
Create database
Filter by databases
DB identifier
Status
Role
Engine
Region ...
Size
aurora
Available
Regional cluster
Aurora My...
us-west-2
1 instance
aurora-instance-1
Creating
Writer instance
Aurora My...
us-west-2b
db.t3.medium


Aurora and RDS
Databases
Successfully created database aurora
You can use settings from aurora to simplify configuration of suggested database add-ons while we finish creating your DB for you.
View connection details
Databases (2)
Group resources
Modify
Actions
Create database
Filter by databases
DB identifier
Status
Role
Engine
Region ...
Size
aurora
Available
Regional cluster
Aurora My...
us-west-2
1 instance
aurora-instance-1
Creating
Writer instance
Aurora My...
us-west-2b
db.t3.medium

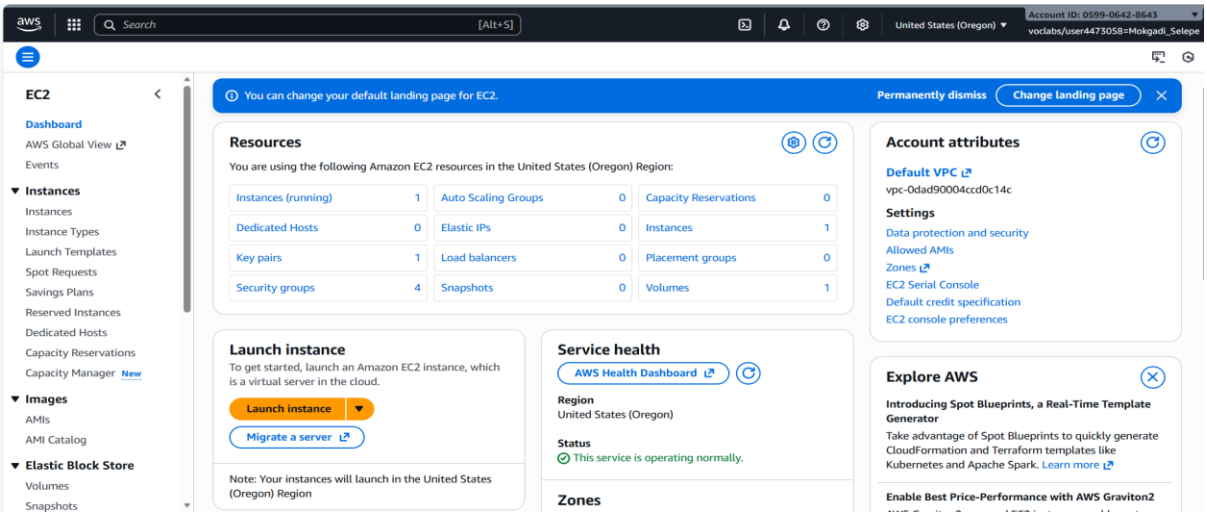
I've created an Amazon Aurora database instance! Here's what I did:

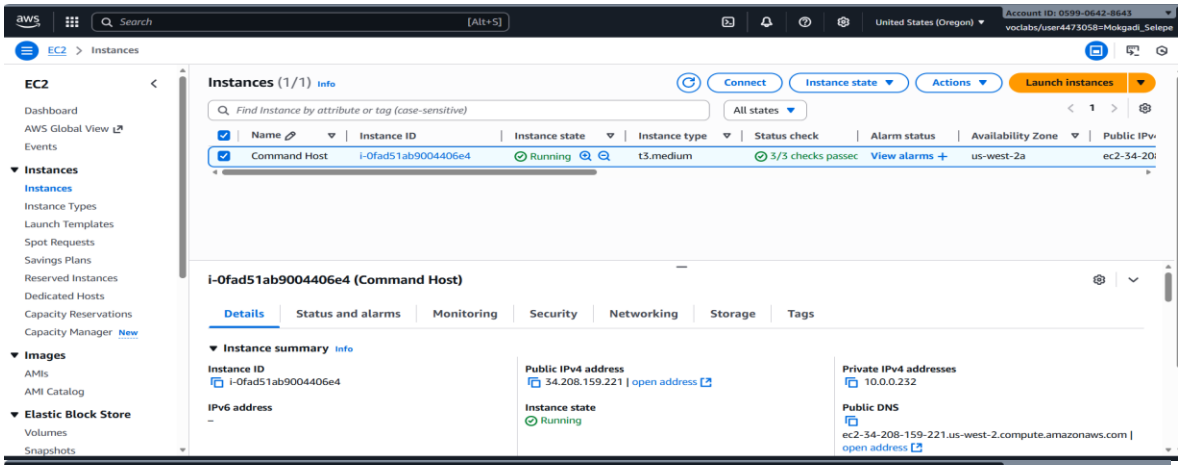
- I chose to create an Aurora database with MySQL compatibility.
- I set up the database with a username (admin) and password (admin123), and chose a database instance type (db.t3.medium).
- I configured the database to run in a specific Virtual Private Cloud (VPC) called LabVPC and subnet group called dbsubnetgroup.
- I created the database, and it's now launching (it'll take a few minutes to be ready).

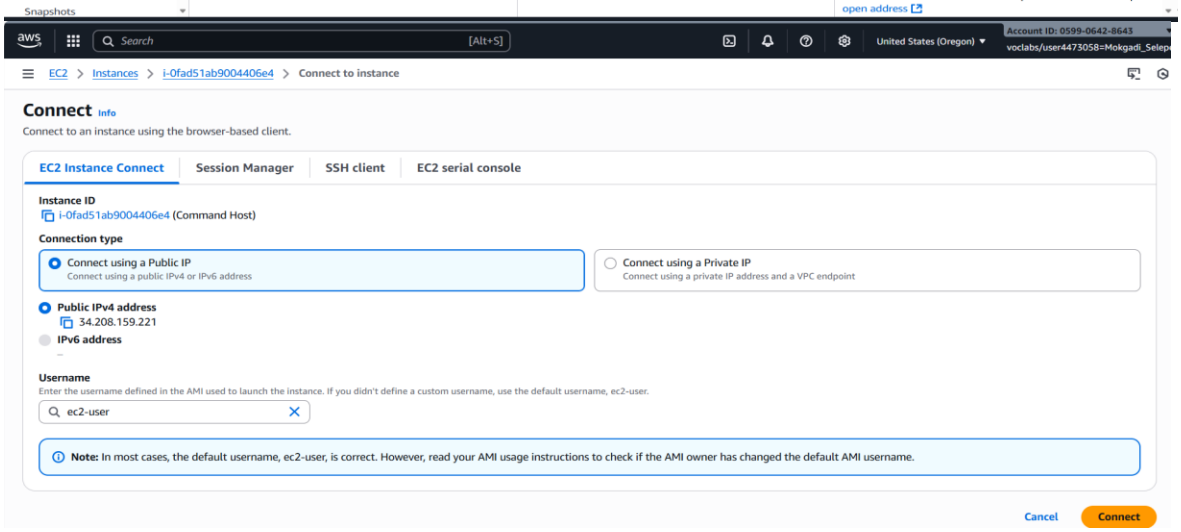
Think of it like setting up a new database server that's fast, reliable, and secure! I've given it a name, "aurora", and an initial database name, "world". I've also configured some security settings, like choosing an existing VPC security group called DBSecurityGroup.

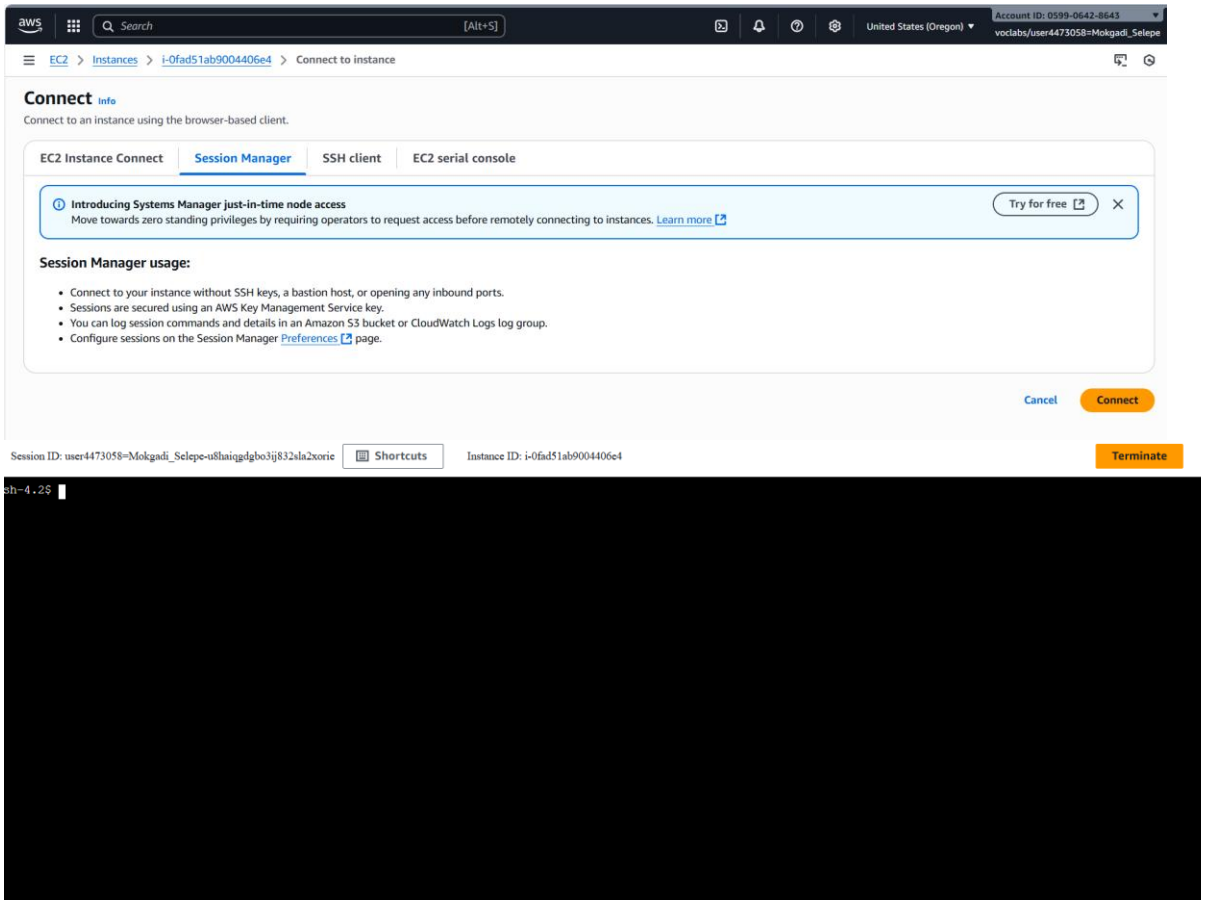
I'm expecting to see a notification message saying "Successfully created database aurora" once it's ready.

2: Connect to an Amazon EC2 Linux instance




Search
[Alt+S]
United States (Oregon)
EC2 > Instances
EC2
Dashboard
AWS Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4
Command Host
i-0fad51ab9004406e4
Running
t3.medium
3/3 checks passed
View alarms
us-west-2a
i-0fad51ab9004406e4 (Command Host)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-0fad51ab9004406e4
Public IPv4 address
34.208.159.221 | open address
Private IPv4 addresses
10.0.0.232
IPv6 address
Instance state
Running
Public DNS
ec2-34-208-159-221.us-west-2.compute.amazonaws.com | open address


Search
[Alt+S]
United States (Oregon)
EC2 > Instances > i-0fad51ab9004406e4 > Connect to instance
Connect Info
Connect to an instance using the browser-based client.
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-0fad51ab9004406e4 (Command Host)
Connection type
Connect using a Public IP
Connect using a public IPv4 or IPv6 address
Connect using a Private IP
Connect using a private IP address and a VPC endpoint
Public IPv4 address
34.208.159.221
IPv6 address
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.
ec2-user
Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
Connect

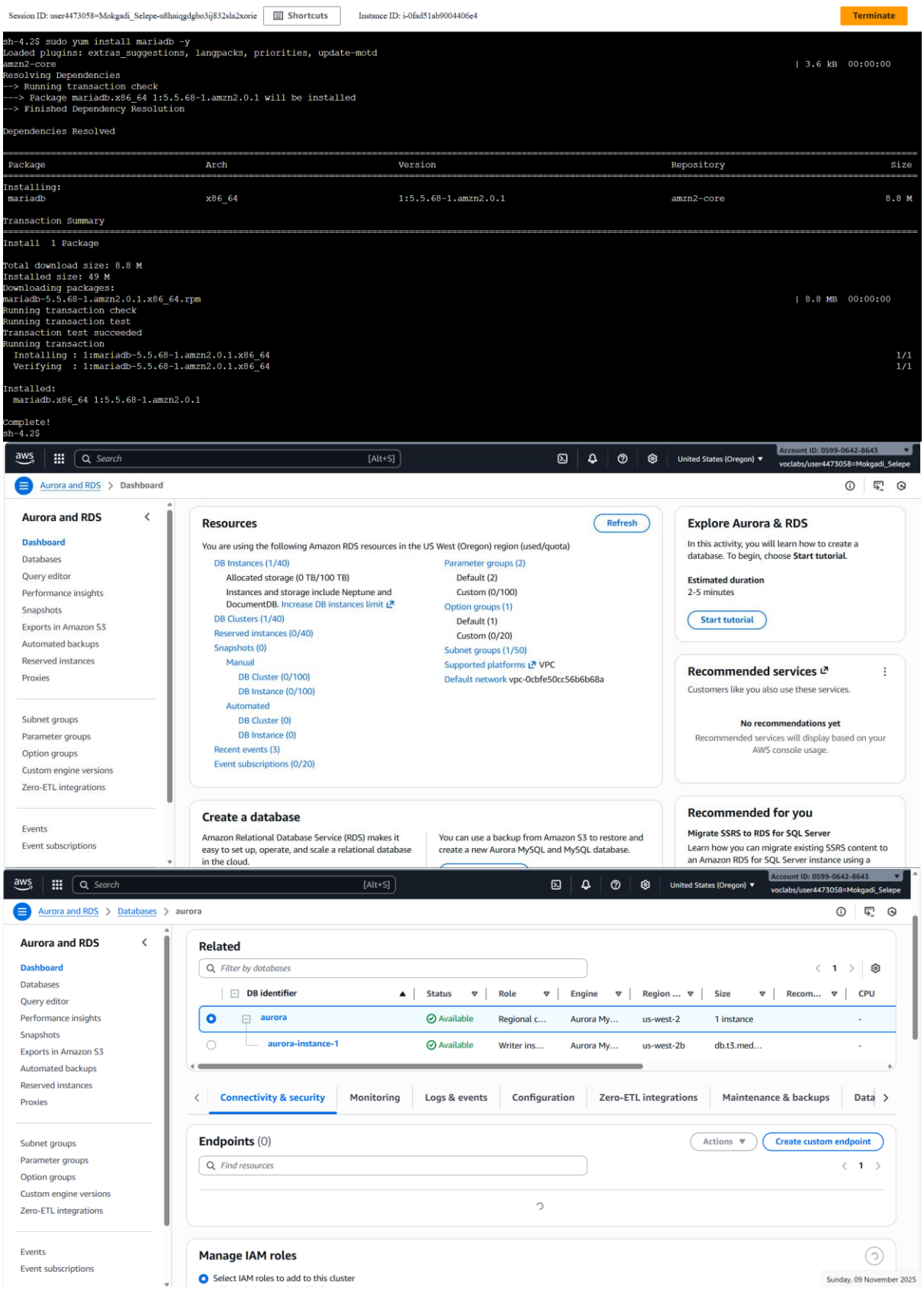


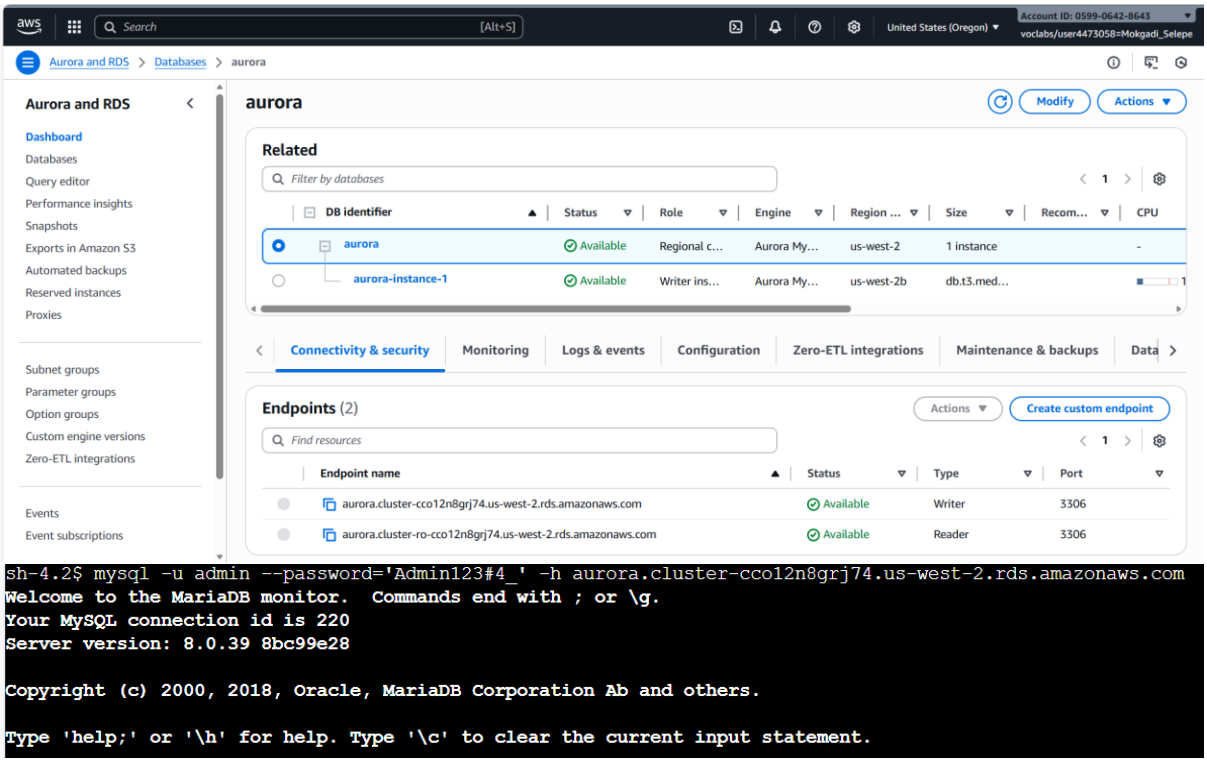
I've connected to an Amazon EC2 Linux instance! Here's what I did:

- I searched for and opened the EC2 service in the AWS Management Console.
- I selected the instance labelled "Command Host" and chose to connect to it.
- I used Session Manager to connect to the instance, which opened a terminal window.

Think of it like logging into a remote computer! I'm now connected to the Command Host instance and can start running commands.

3: Configure the Amazon EC2 Linux instance to connect to Aurora


Session ID: user4473058=Mokgadi_Selepe-u8haiqqdgbo3ij832sla2xorie
Shortcuts
Instance ID: i-0fad51ab9004406e4
Terminate
sh-4.2$ sudo yum install mariadb -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00
Resolving Dependencies
--> Running transaction check
---> Package mariadb.x86_64 1:5.5.68-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
Package Arch Version Repository Size
Installing:
mariadb x86_64 1:5.5.68-1.amzn2.0.1 amzn2-core 8.8 M
Transaction Summary
Install 1 Package
Total download size: 8.8 M
Installed size: 49 M
Downloading packages:
mariadb-5.5.68-1.amzn2.0.1.x86_64.rpm | 8.8 MB 00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1
Verifying : 1:mariadb-5.5.68-1.amzn2.0.1.x86_64 1/1
Installed:
mariadb.x86_64 1:5.5.68-1.amzn2.0.1
Complete!
sh-4.2$
Search [Alt+S]
United States (Oregon)
Account ID: 0599-0642-8643
voclabs/user4473058=Mokgadi_Selepe
Aurora and RDS > Dashboard
Aurora and RDS
Dashboard
Databases
Query editor
Performance insights
Snapshots
Exports in Amazon S3
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Custom engine versions
Zero-ETL integrations
Events
Event subscriptions
Resources
Refresh
You are using the following Amazon RDS resources in the US West (Oregon) region (used/quota)
DB Instances (1/40)
Allocated storage (0 TB/100 TB)
Instances and storage include Neptune and DocumentDB. Increase DB instances limit
DB Clusters (1/40)
Reserved instances (0/40)
Snapshots (0)
Manual
DB Cluster (0/100)
DB Instance (0/100)
Automated
DB Cluster (0)
DB Instance (0)
Recent events (3)
Event subscriptions (0/20)
Parameter groups (2)
Default (2)
Custom (0/100)
Option groups (1)
Default (1)
Custom (0/20)
Subnet groups (1/50)
Supported platforms VPC
Default network vpc-0cbfe50cc56b6b68a
Explore Aurora & RDS
In this activity, you will learn how to create a database. To begin, choose Start tutorial.
Estimated duration
2-5 minutes
Start tutorial
Recommended services
Customers like you also use these services.
No recommendations yet
Recommended services will display based on your AWS console usage.
Create a database
Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale a relational database in the cloud.
You can use a backup from Amazon S3 to restore and create a new Aurora MySQL and MySQL database.
Recommended for you
Migrate SSRS to RDS for SQL Server
Learn how you can migrate existing SSRS content to an Amazon RDS for SQL Server instance using a
Aurora and RDS > Databases > aurora
Related
Filter by databases
DB identifier Status Role Engine Region ... Size Recom... CPU
aurora Available Regional c... Aurora My... us-west-2 1 instance -
aurora-instance-1 Available Writer ins... Aurora My... us-west-2b db.t3.med... -
Connectivity & security
Monitoring
Logs & events
Configuration
Zero-ETL integrations
Maintenance & backups
Data
Endpoints (0)
Actions
Create custom endpoint
Find resources
Manage IAM roles
Select IAM roles to add to this cluster
Sunday, 09 November 2025




```
sh-4.2$ mysql -u admin --password='Admin123#4_' -h aurora.cluster-cco12n8grj74.us-west-2.rds.amazonaws.com
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 220
Server version: 8.0.39 8bc99e28

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

I've configured my Amazon EC2 Linux instance to connect to my Aurora database!

Here's what I did:

- I installed the MariaDB client using the yum package manager, which allows me to connect to my Aurora instance.
- I went back to the AWS Management Console, found my Aurora instance, and copied the endpoint name for the Writer instance.
- I used the endpoint to log into my database using the MySQL Command-Line Client, providing my username and password.

Think of it like setting up a connection between my Linux instance and my database, so I can start interacting with it! I've got the MariaDB monitor open, and I'm ready to start running queries.

4: Create a table and insert and query records

```
MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| world              |
+--------------------+
5 rows in set (0.00 sec)

MySQL [(none)]> USE world;
Database changed
MySQL [world]>
```

Session ID: user4473058=Mokgadi_Selepe-eeg7nroqatl2nrkzelt3tcnhpi | Shortcuts | Instance ID: i-01220c8369e039c6a | Terminate

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| world              |
+--------------------+
5 rows in set (0.00 sec)

MySQL [(none)]> USE world;
Database changed
MySQL [world]> CREATE TABLE `country` (
    -> `Code` CHAR(3) NOT NULL DEFAULT '',
    -> `Name` CHAR(52) NOT NULL DEFAULT '',
    -> `Continent` enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America') NOT NULL DEFAULT 'Asia',
    -> `Region` CHAR(26) NOT NULL DEFAULT '',
    -> `SurfaceArea` FLOAT(10,2) NOT NULL DEFAULT '0.00',
    -> `IndepYear` SMALLINT(6) DEFAULT NULL,
    -> `Population` INT(11) NOT NULL DEFAULT '0',
    -> `LifeExpectancy` FLOAT(3,1) DEFAULT NULL,
    -> `GNP` FLOAT(10,2) DEFAULT NULL,
    -> `GNPOld` FLOAT(10,2) DEFAULT NULL,
    -> `LocalName` CHAR(45) NOT NULL DEFAULT '',
    -> `GovernmentForm` CHAR(45) NOT NULL DEFAULT '',
    -> `Capital` INT(11) DEFAULT NULL,
    -> `Code2` CHAR(2) NOT NULL DEFAULT '',
    -> PRIMARY KEY (`Code`)
    -> );
Query OK, 0 rows affected, 7 warnings (0.09 sec)

MySQL [world]>

MySQL [world]> INSERT INTO `country` VALUES ('GAB','Gabon','Africa','Central Africa',267668.00,1960,1226000,50.1,5493.00,5279.00,'Le Gabon','Republic',902,'GA');
Query OK, 1 row affected (0.00 sec)

MySQL [world]>
MySQL [world]> INSERT INTO `country` VALUES ('IRL','Ireland','Europe','British Islands',70273.00,1921,3775100,76.8,75921.00,73132.00,'Ireland/Éire','Republic',1447,'IE');
Query OK, 1 row affected (0.00 sec)

MySQL [world]>
MySQL [world]> INSERT INTO `country` VALUES ('THA','Thailand','Asia','Southeast Asia',513115.00,1350,61399000,68.6,116416.00,153907.00,'Prathet Thai','Constitutional Monar
chy',3320,'TH');
Query OK, 1 row affected (0.00 sec)

MySQL [world]>
MySQL [world]> INSERT INTO `country` VALUES ('CRI','Costa Rica','North America','Central America',51100.00,1821,4023000,75.8,10226.00,9757.00,'Costa Rica','Republic',584,'
CR');
Query OK, 1 row affected (0.00 sec)

MySQL [world]>
MySQL [world]> INSERT INTO `country` VALUES ('AUS','Australia','Oceania','Australia and New Zealand',7741220.00,1901,18886000,79.8,351182.00,392911.00,'Australia','Constit
utional Monarchy, Federation',135,'AU');
Query OK, 1 row affected (0.00 sec)

MySQL [world]>
MySQL [world]> SELECT * FROM country WHERE GNP > 35000 and Population > 10000000;
+------+-----------+-----------+--------------------------+-------------+-----------+------------+----------------+-----------+-----------+--------------+----------------
--------------------+---------+-------+
| Code | Name      | Continent | Region                   | SurfaceArea | IndepYear | Population | LifeExpectancy | GNP       | GNPOld    | LocalName    | GovernmentForm
                    | Capital | Code2 |
+------+-----------+-----------+--------------------------+-------------+-----------+------------+----------------+-----------+-----------+--------------+----------------
--------------------+---------+-------+
| AUS  | Australia | Oceania   | Australia and New Zealand |  7741220.00 |      1901 |   18886000 |           79.8 | 351182.00 | 392911.00 | Australia    | Constitutional
Monarchy, Federation |     135 | AU    |
| THA  | Thailand  | Asia      | Southeast Asia           |   513115.00 |      1350 |   61399000 |           68.6 | 116416.00 | 153907.00 | Prathet Thai | Constitutional
Monarchy              |    3320 | TH    |
+------+-----------+-----------+--------------------------+-------------+-----------+------------+----------------+-----------+-----------+--------------+----------------
--------------------+---------+-------+
2 rows in set (0.00 sec)
```

I've worked with a database!

Here's what I did:

- I listed the available databases and found the "world" database I created earlier.
- I switched to the "world" database and created a new table called "country" with columns for country data.
- I inserted five new records into the "country" table with data about different countries.
- I ran a query to find countries with a GNP (Gross National Product) over 35,000 and a population over 10 million, and it returned two records: Australia and Thailand!

Think of it like creating a spreadsheet, adding data to it, and then searching for specific information in the spreadsheet.

Conclusion

I've wrapped up my project!

Here's what I've accomplished:

- I created an Amazon Aurora database instance, which is a powerful and reliable database.
- I connected to an Amazon EC2 instance, which is a virtual computer in the cloud.
- I set up the EC2 instance to connect to my Aurora database, so they can talk to each other.
- I used the EC2 instance to query my Aurora database, running commands and getting results.

Think of it like building a house:

- I built a strong foundation (Aurora database).
- I created a workshop (EC2 instance) to work on my project.
- I connected the workshop to the foundation (configured EC2 to connect to Aurora).
- I started working on my project, using the workshop to query the foundation (ran queries on Aurora).

It's been a great learning experience, and I'm excited to keep exploring and building with AWS!
