

Implantando um banco de dados MySQL com AWS RDS

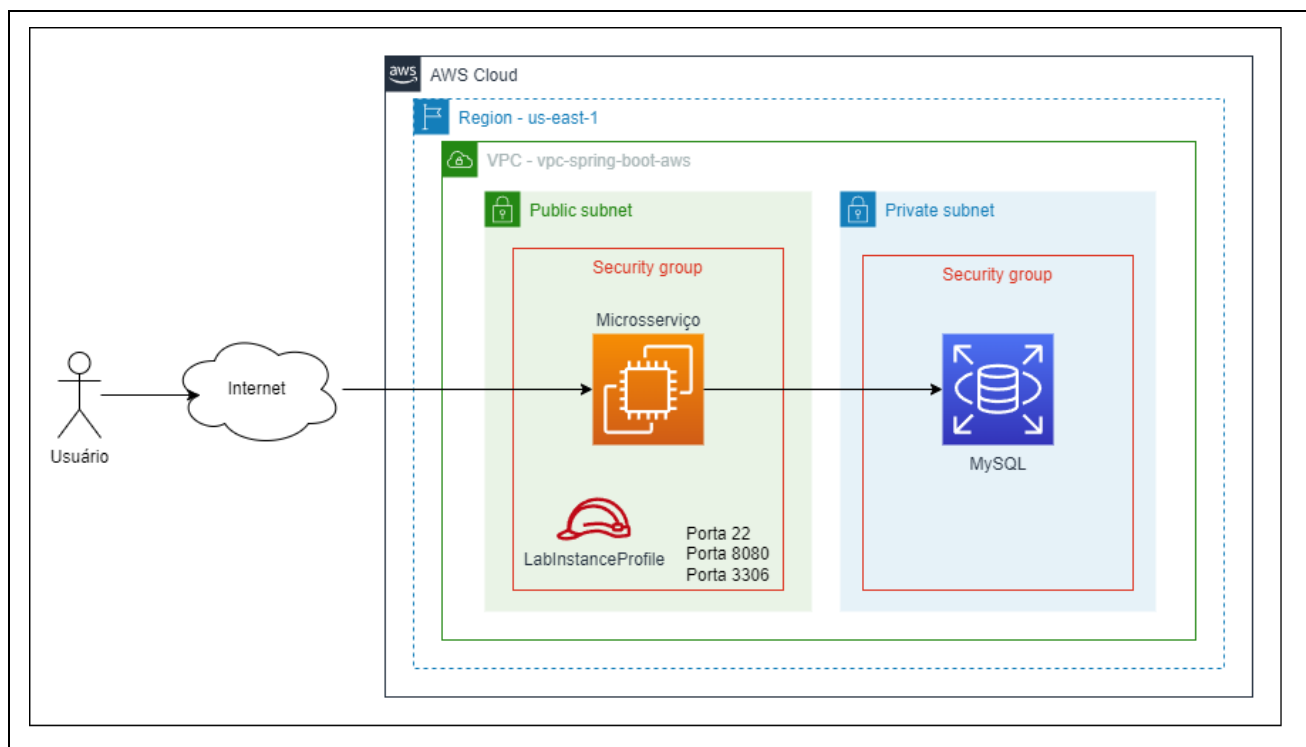


Prof. Thomás da Costa - <https://thomasdacosta.com.br>

Objetivo:

Implantar um banco de dados RDS para subir um microserviço desenvolvido em Spring Boot. Este guia é composto por mais partes onde em tutoriais futuros, subiremos o restante da aplicação para expor o serviço para que os usuários possam utilizar.

Desenho da Solução:

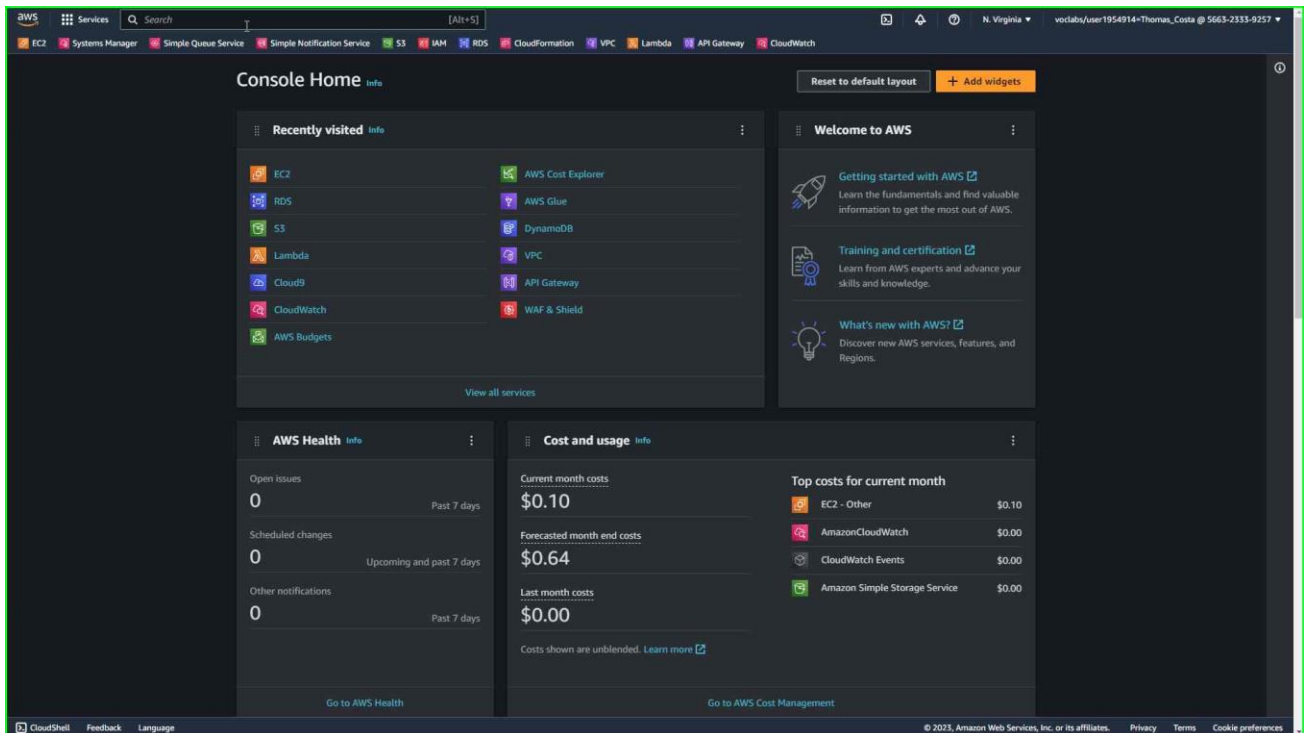


Premissas:

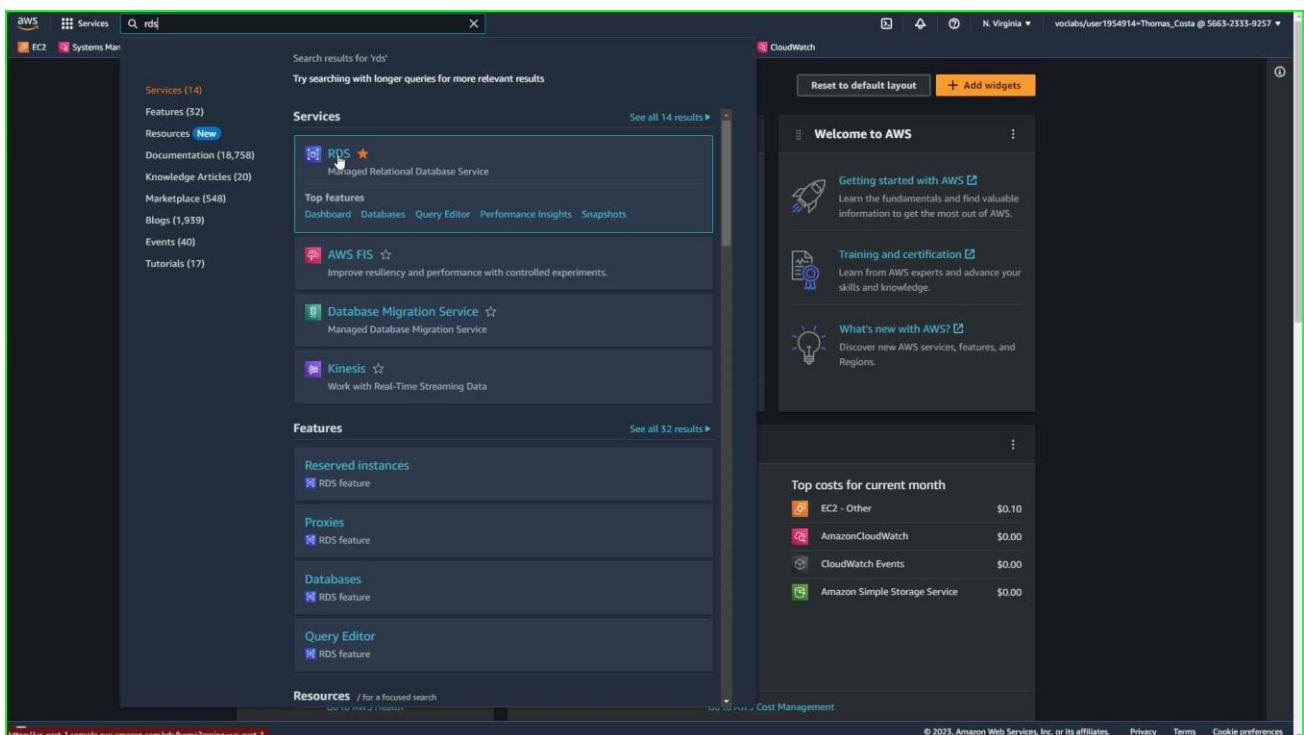
- Faça o tutorial do link abaixo antes de iniciar:
 - <https://thomasdacosta.com.br/2023/09/03/step-by-step-acessando-remotamente-um-aws-ec2-atraves-de-ssh/>
- Caso deseje somente subir o banco de dados, siga somente as instruções de criação de banco de dados;
- Utilize o microserviço em Spring Boot do seguinte link:
 - https://github.com/thomasdacosta/aula-aws/blob/main/aula_07_banco_de_dados_relacional/spring-boot-localstack.jar

Implantando um banco de dados MySQL com AWS RDS

Entre na opção "Search":

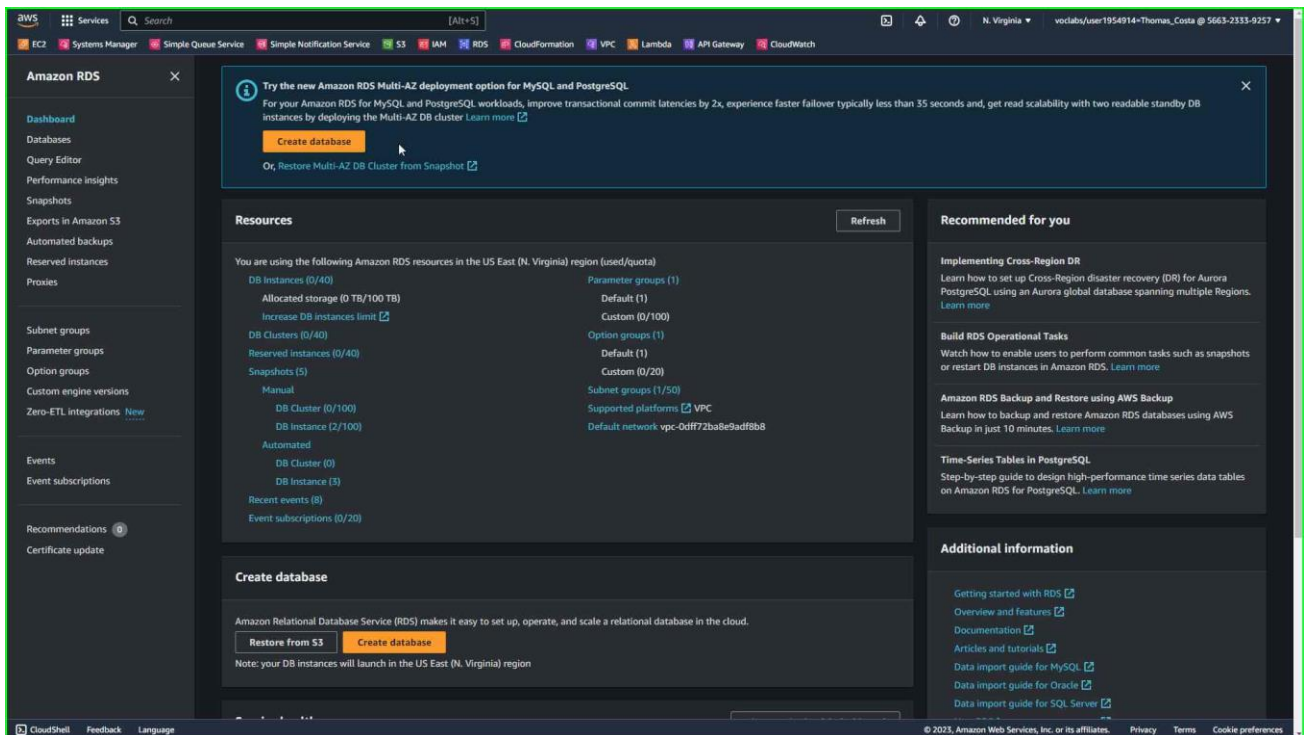


Na tela principal, pesquisar a opção "RDS":

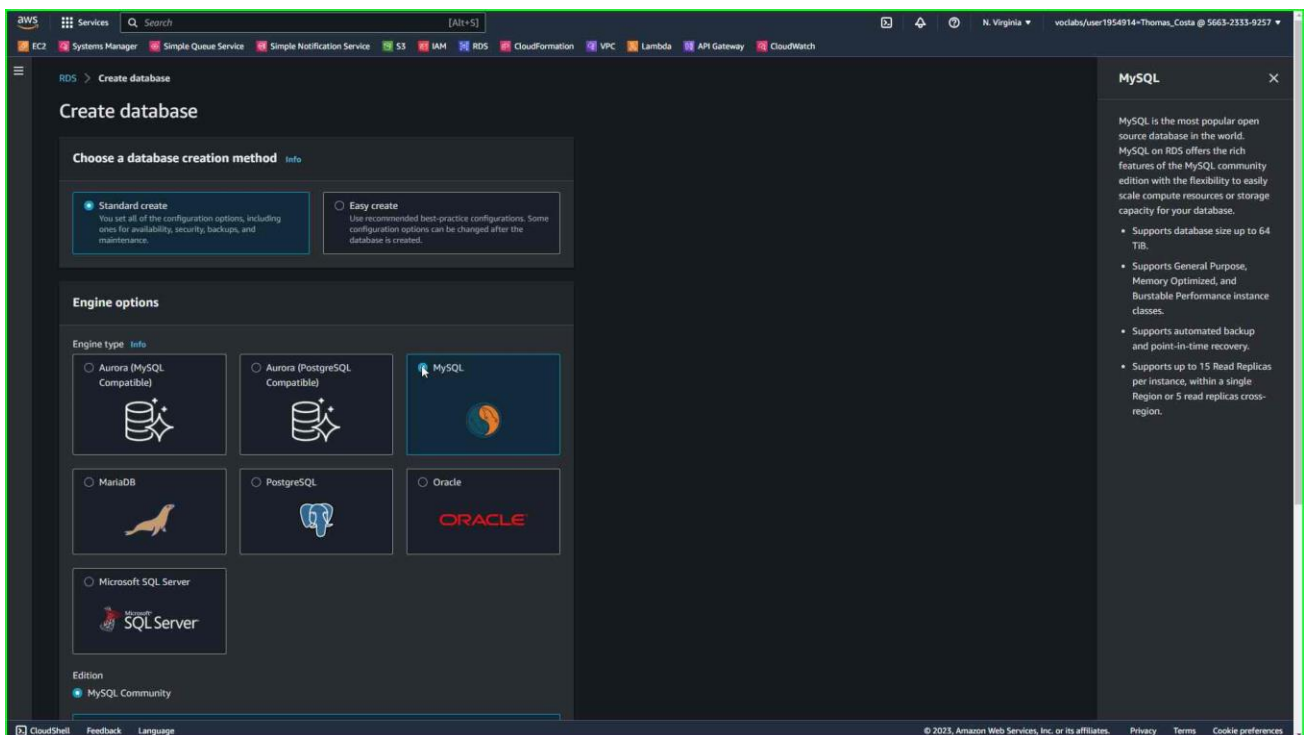


Implantando um banco de dados MySQL com AWS RDS

Clique no botão “Create database”:



Selecione o banco de dados “MySQL”:



Implantando um banco de dados MySQL com AWS RDS

Selecione a opção "Free tier":

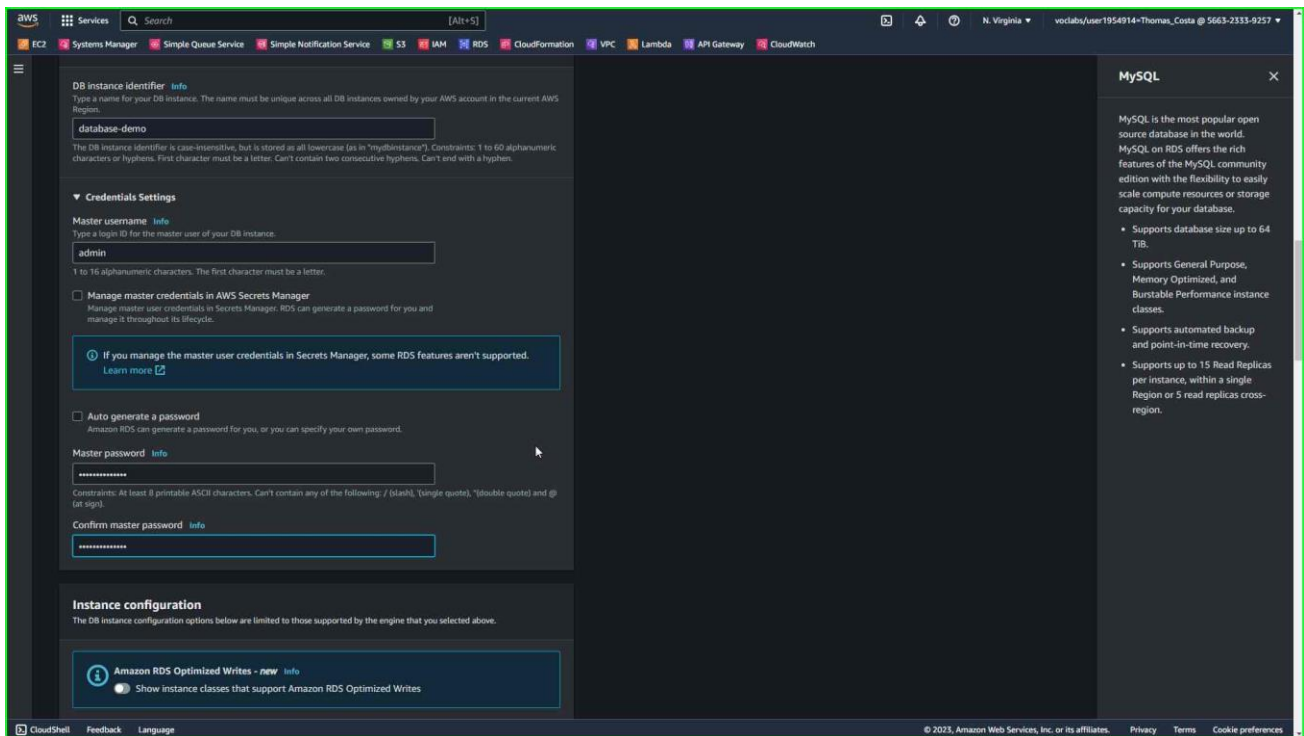
The screenshot shows the AWS Management Console interface for creating a new Amazon RDS instance. The 'Engine Version' is set to 'MySQL 8.0.33'. Under the 'Templates' section, three options are visible: 'Production', 'Dev/Test', and 'Free tier'. The 'Free tier' option is selected, indicated by a blue highlight and a mouse cursor. The 'Free tier' description states: 'Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.' The 'Availability and durability' section shows 'Multi-AZ DB Instance' as the selected deployment option. The right-hand sidebar displays the 'MySQL' product page with a list of features.

Digite o nome do banco de dados:

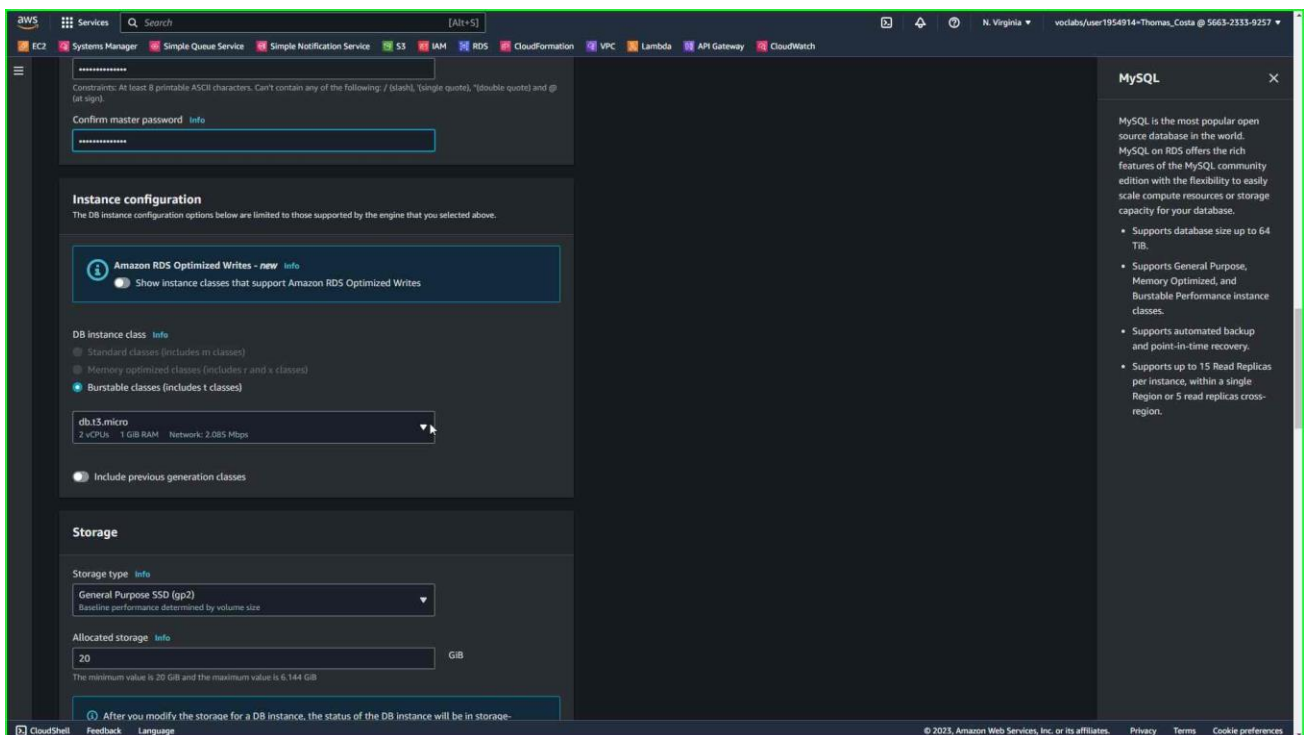
The screenshot shows the 'Settings' section of the AWS Management Console for creating a new Amazon RDS instance. The 'DB instance identifier' field is filled with the text 'database-demo'. Below this field, there are sections for 'Credentials Settings', including 'Master username' (filled with 'admin') and 'Master password'. The 'Auto generate a password' checkbox is unchecked. The right-hand sidebar displays the 'MySQL' product page with a list of features.

Implantando um banco de dados MySQL com AWS RDS

Digite a senha para acessar o banco de dados:

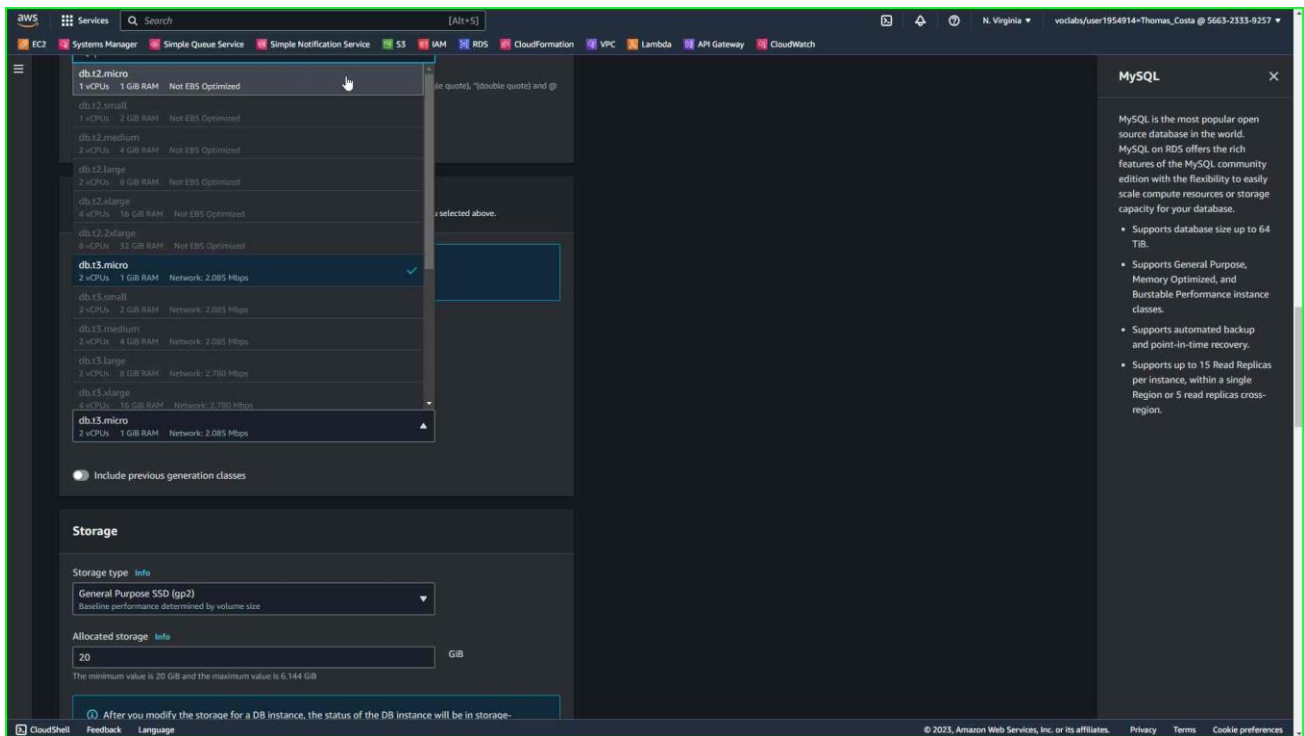


Modifique o tipo de máquina do banco de dados:

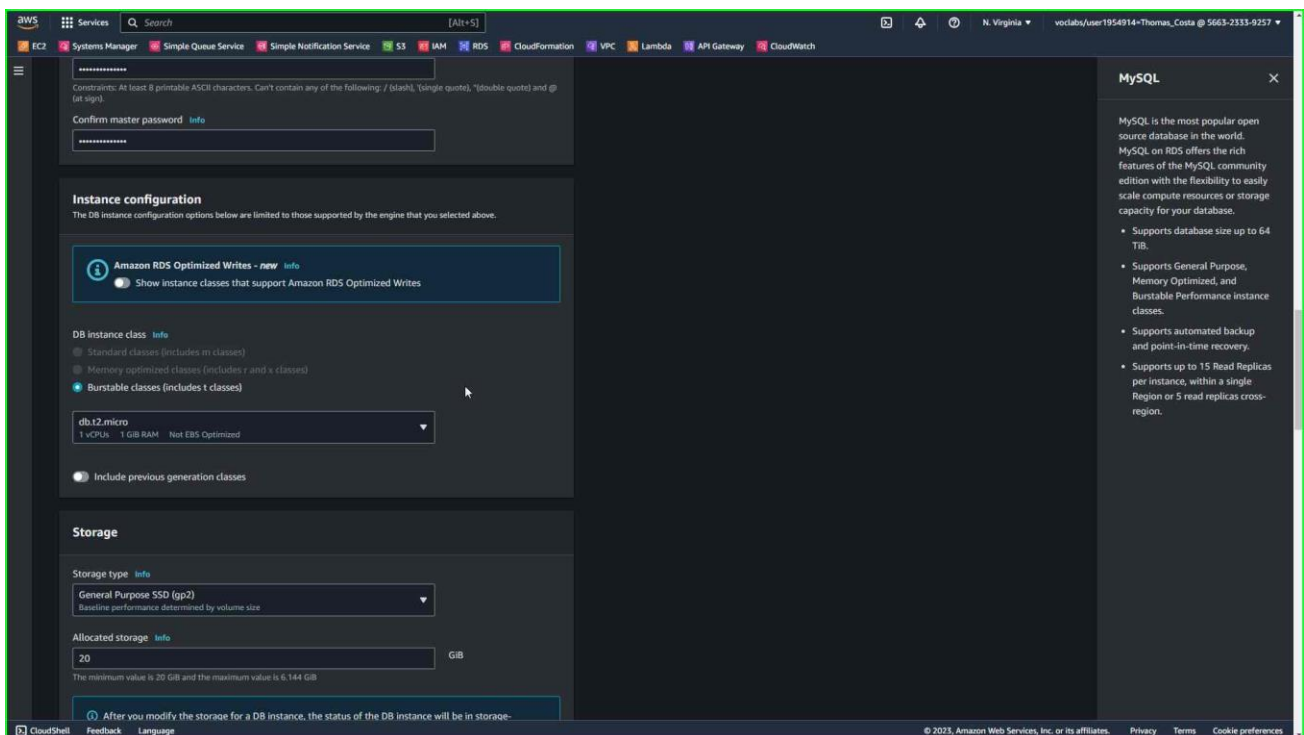


Implantando um banco de dados MySQL com AWS RDS

Selecione a opção “db.t2.micro”:

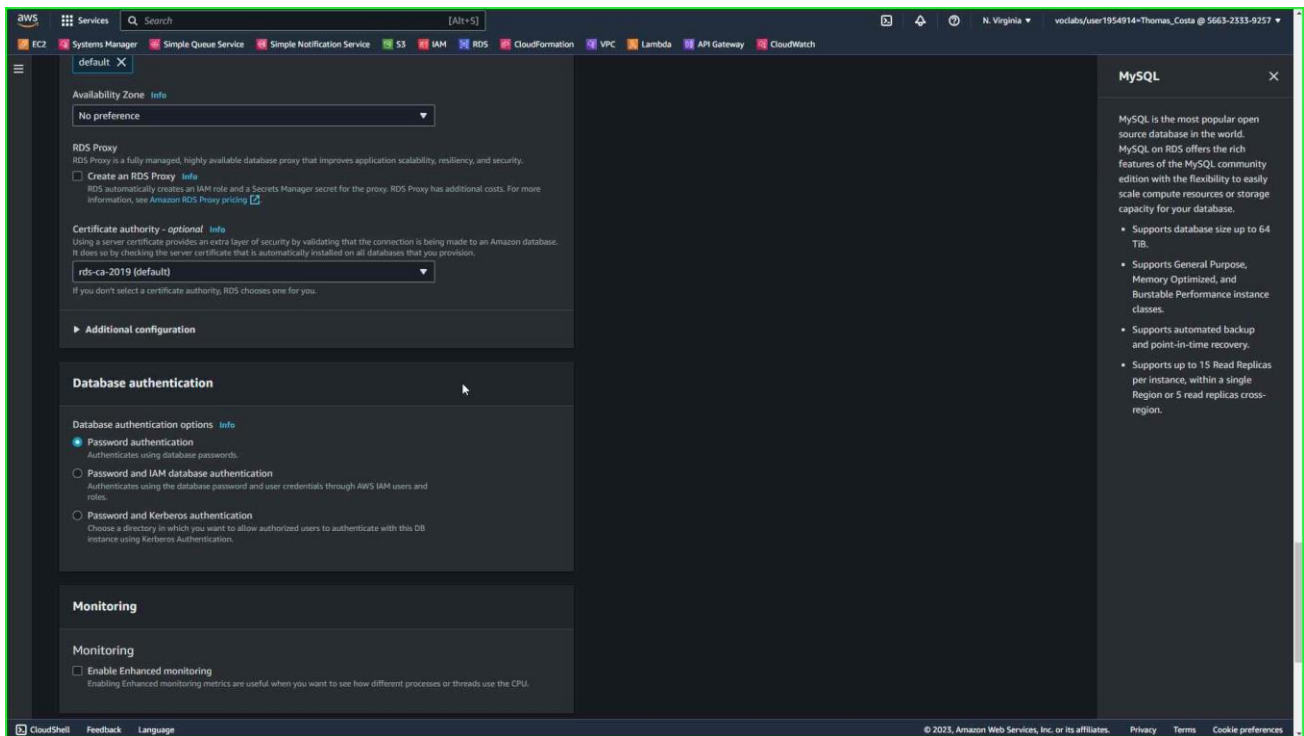


Banco de dados selecionado:



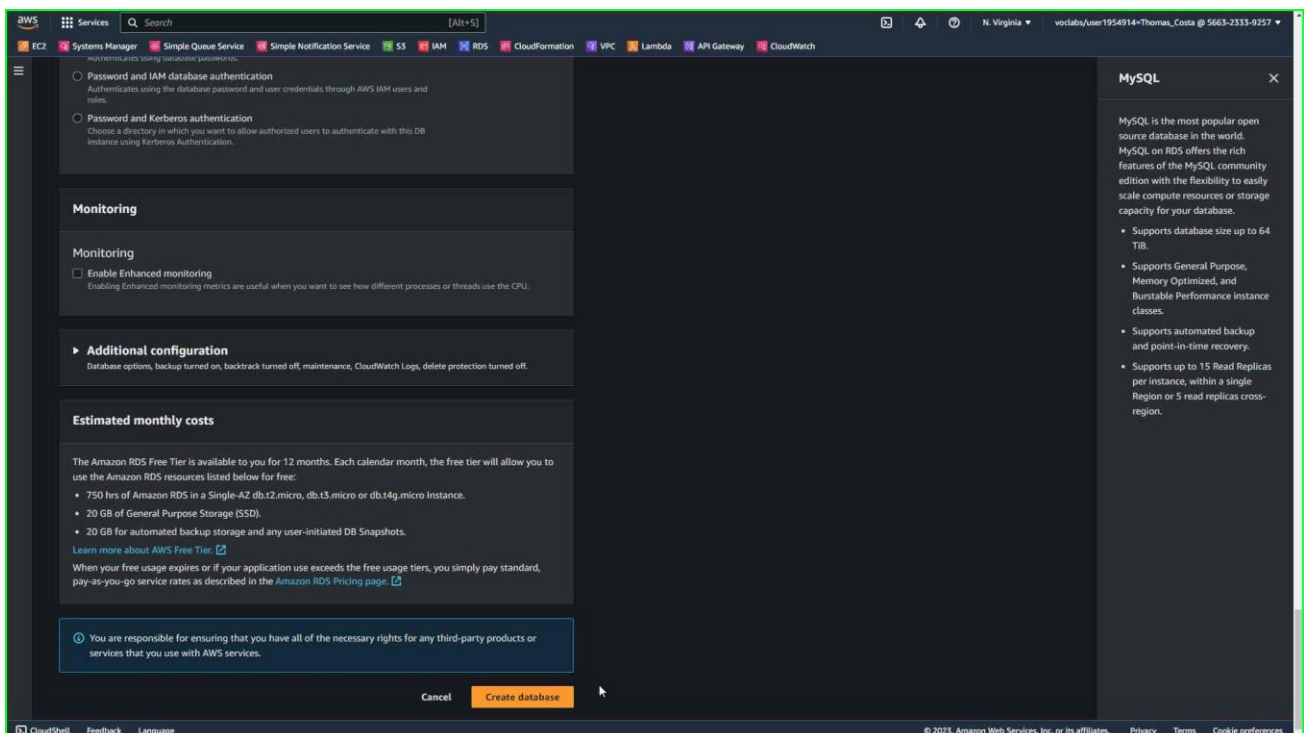
Implantando um banco de dados MySQL com AWS RDS

Mantenha as opções conforme abaixo:



The screenshot shows the AWS RDS console configuration page for a MySQL instance. The 'Availability Zone' is set to 'No preference'. The 'RDS Proxy' section has 'Create an RDS Proxy' unchecked. The 'Certificate authority - optional' section has 'rds-ca-2019 (default)' selected. The 'Database authentication' section has 'Password authentication' selected. The 'Monitoring' section has 'Enable Enhanced monitoring' unchecked. A 'MySQL' sidebar on the right lists features like database size up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, and up to 15 Read Replicas.

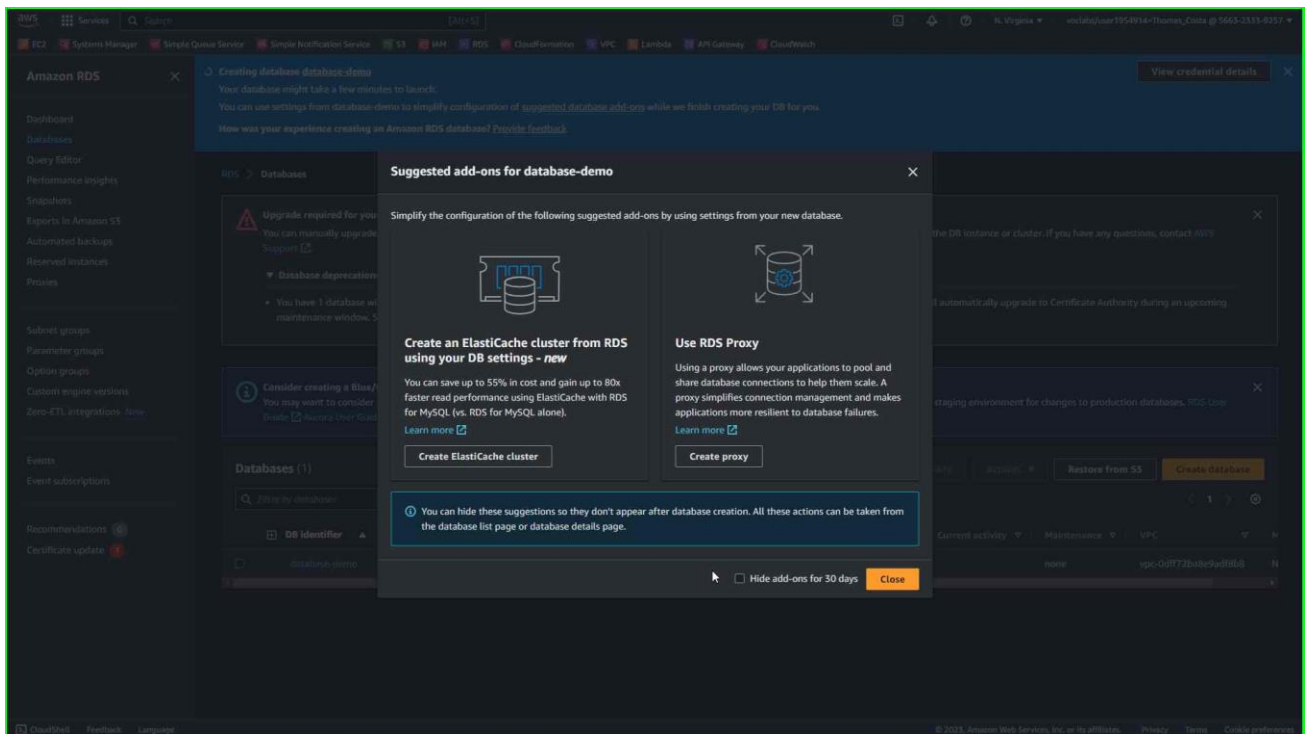
Clique em “Create database”:



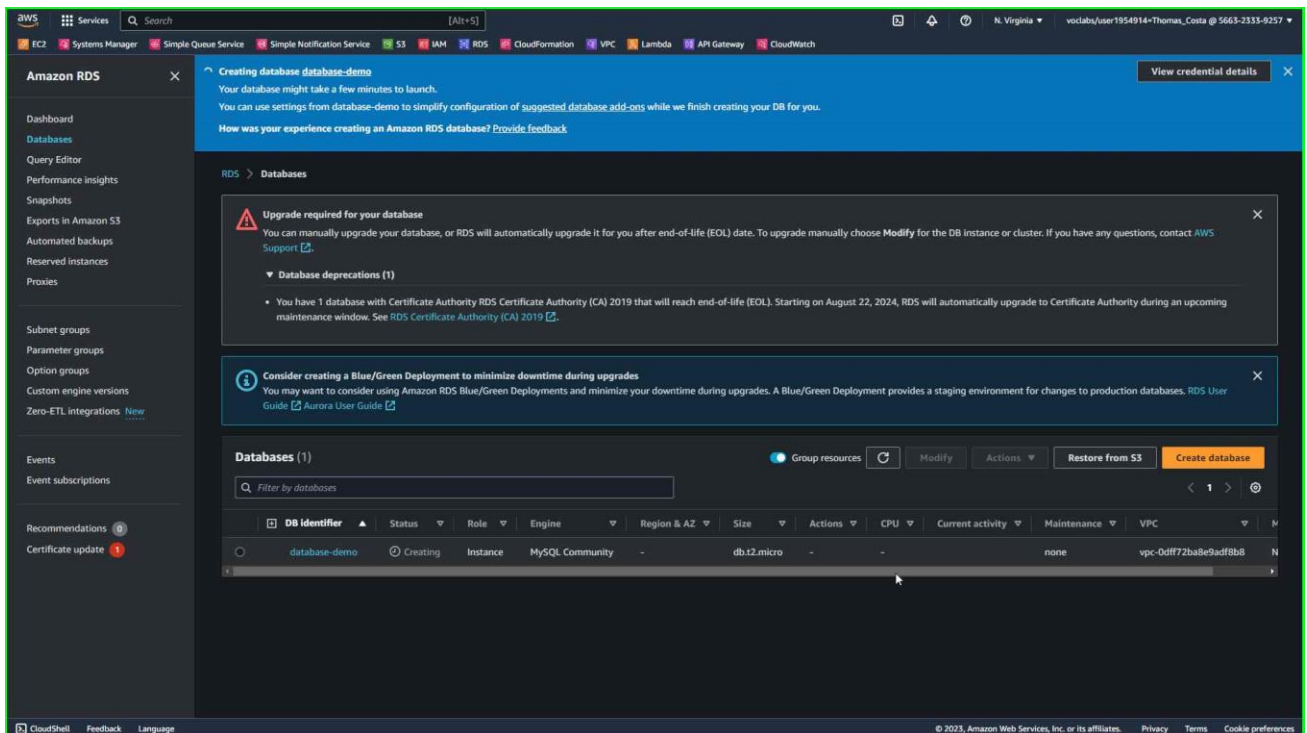
The screenshot shows the 'Estimated monthly costs' section of the AWS RDS console. It details the Amazon RDS Free Tier, which is available for 12 months. The free tier includes 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance, 20 GB of General Purpose Storage (SSD), and 20 GB for automated backup storage and any user-initiated DB Snapshots. A 'Create database' button is visible at the bottom right.

Implantando um banco de dados MySQL com AWS RDS

Clique em “Close”:

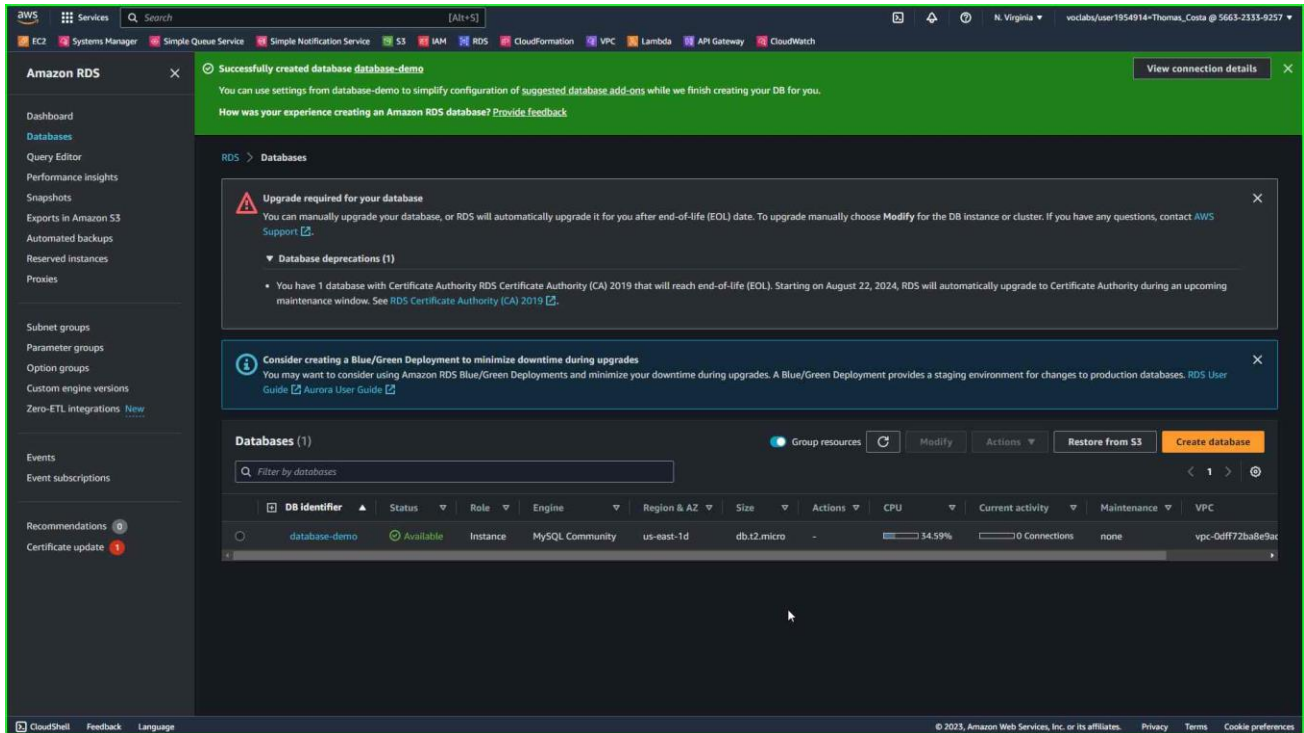


Aguarde a criação do banco de dados:

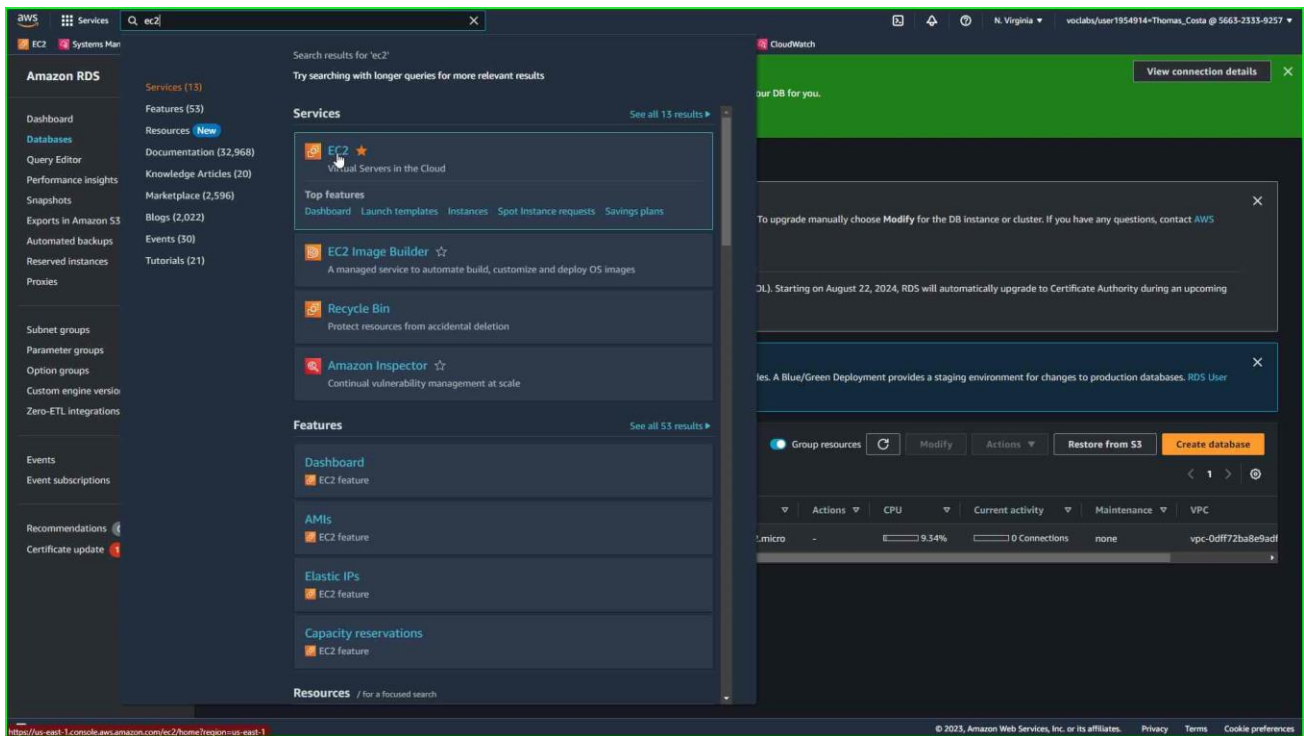


Implantando um banco de dados MySQL com AWS RDS

Espere o banco entrar no estado “Available”:



Selecione a opção “EC2”:



Implantando um banco de dados MySQL com AWS RDS

Selecione “Instances (running)”:

The screenshot shows the AWS Management Console interface. In the 'Resources' section, a table lists various Amazon EC2 resources. The 'Instances (running)' link is highlighted with a mouse cursor. The table contains the following data:

Resource	Count	Resource	Count
Instances (running)	1	Auto Scaling Groups	0
Elastic IPs	0	Instances	1
Load balancers	0	Placement groups	0
Snapshots	0	Volumes	1
		Key pairs	2
		Security groups	3

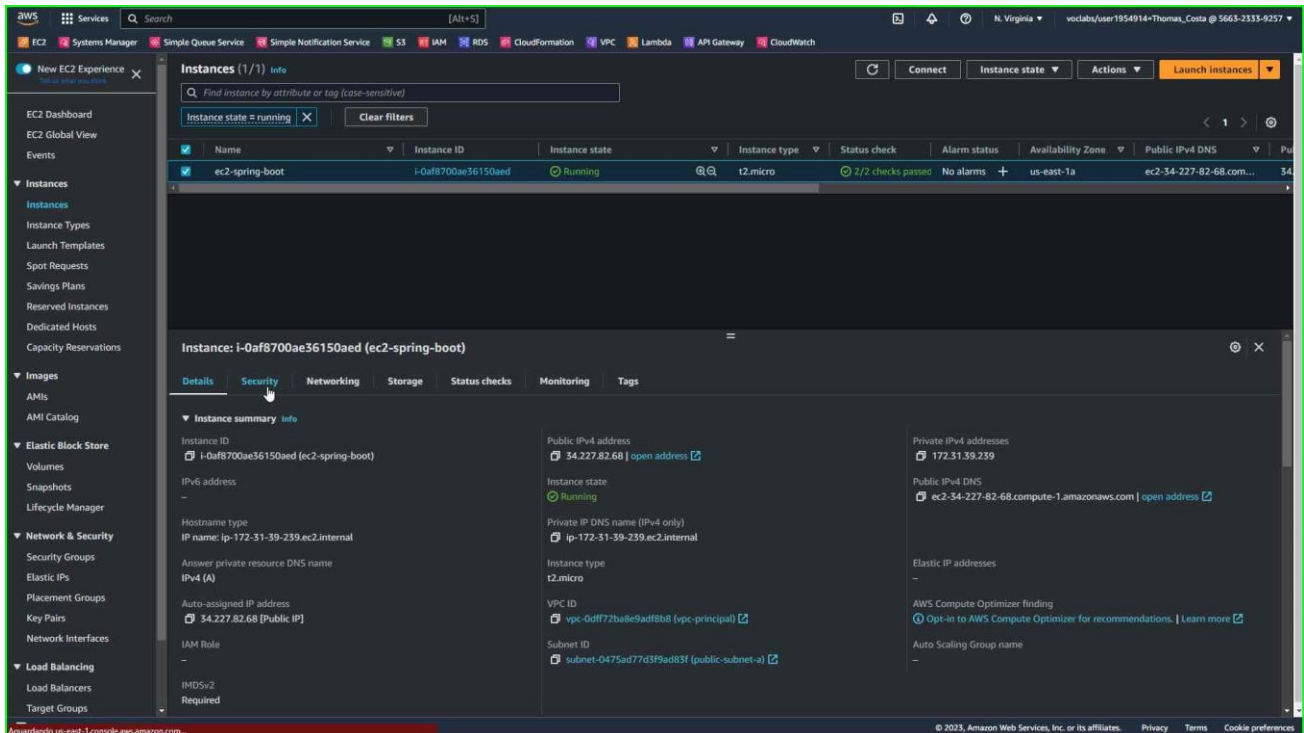
Selecione uma instância que tenha o microserviço em Spring Boot:

The screenshot shows the AWS Management Console interface for the 'Instances (1)' page. A table lists the instances, and the 'ec2-spring-boot' instance is selected. The table contains the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
ec2-spring-boot	i-0af8700ae36150aed	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	ec2-34-227-82-68.com...

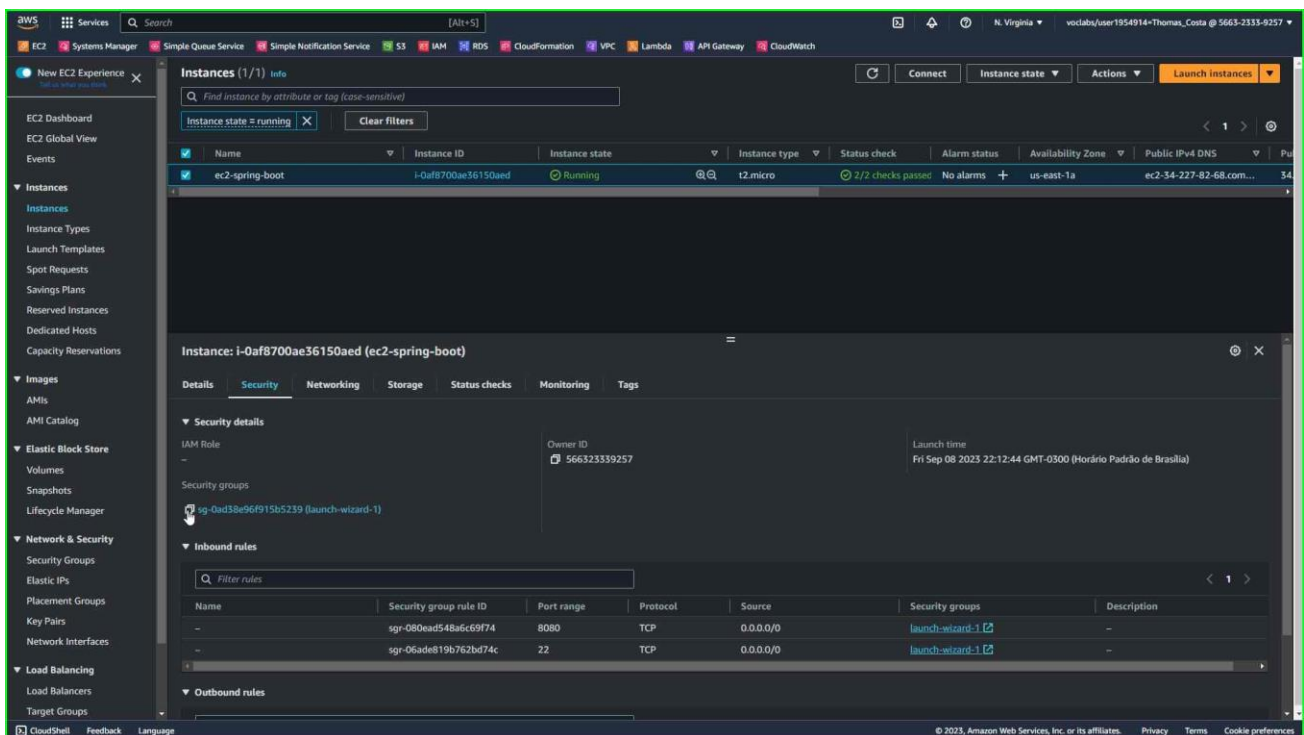
Implantando um banco de dados MySQL com AWS RDS

Clique na aba “Security”:



The screenshot shows the AWS Management Console interface. The left sidebar contains navigation options like 'Instances', 'Images', 'Elastic Block Store', 'Network & Security', and 'Load Balancing'. The main content area displays the details for an EC2 instance named 'ec2-spring-boot' with Instance ID 'i-0af8700ae36150aed'. The 'Security' tab is selected, showing the 'Instance summary' section. Key details include the Instance ID, Public IPv4 address (34.227.82.68), Instance state (Running), and various network and IAM configurations.

Copie o nome do “Security groups”:

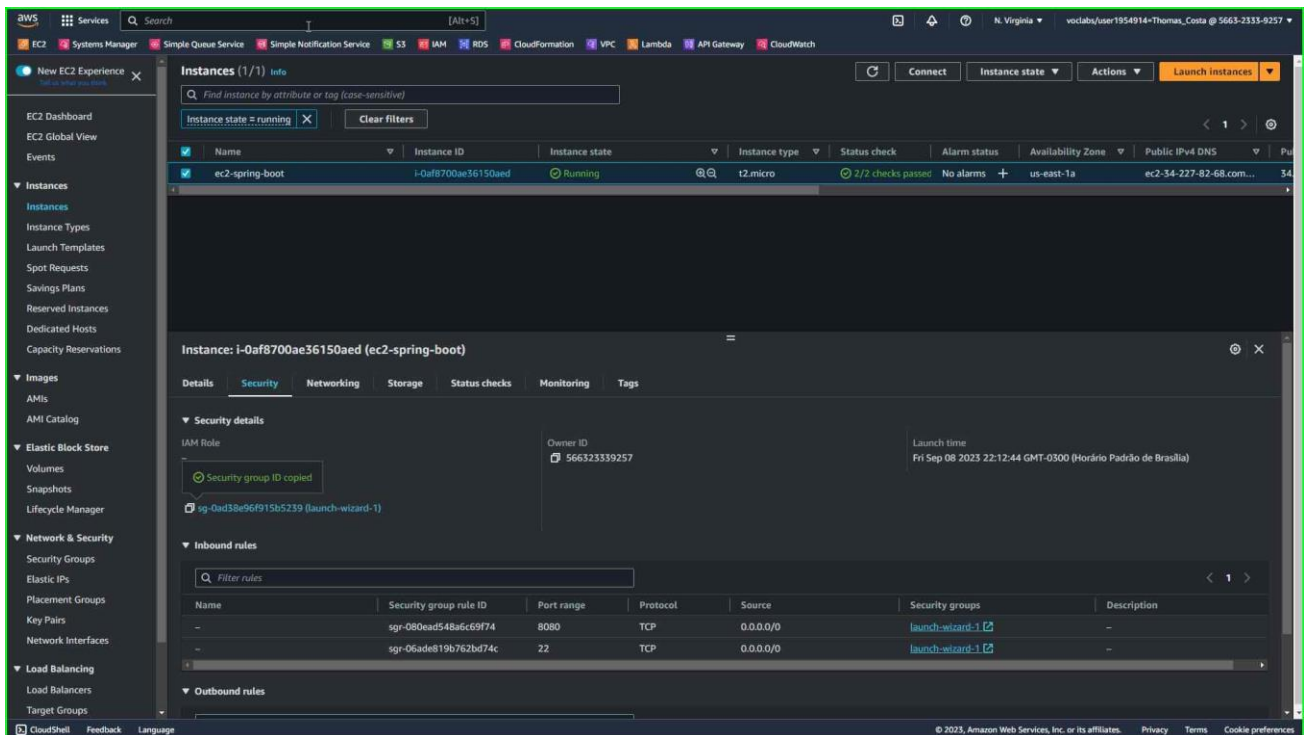


The screenshot shows the AWS Management Console interface, focusing on the 'Security groups' section for the same EC2 instance. The 'Security groups' list shows a single group: 'sg-0ad38e96f915b5239 (launch-wizard-1)'. Below this, the 'Inbound rules' section is expanded, showing a table of rules:

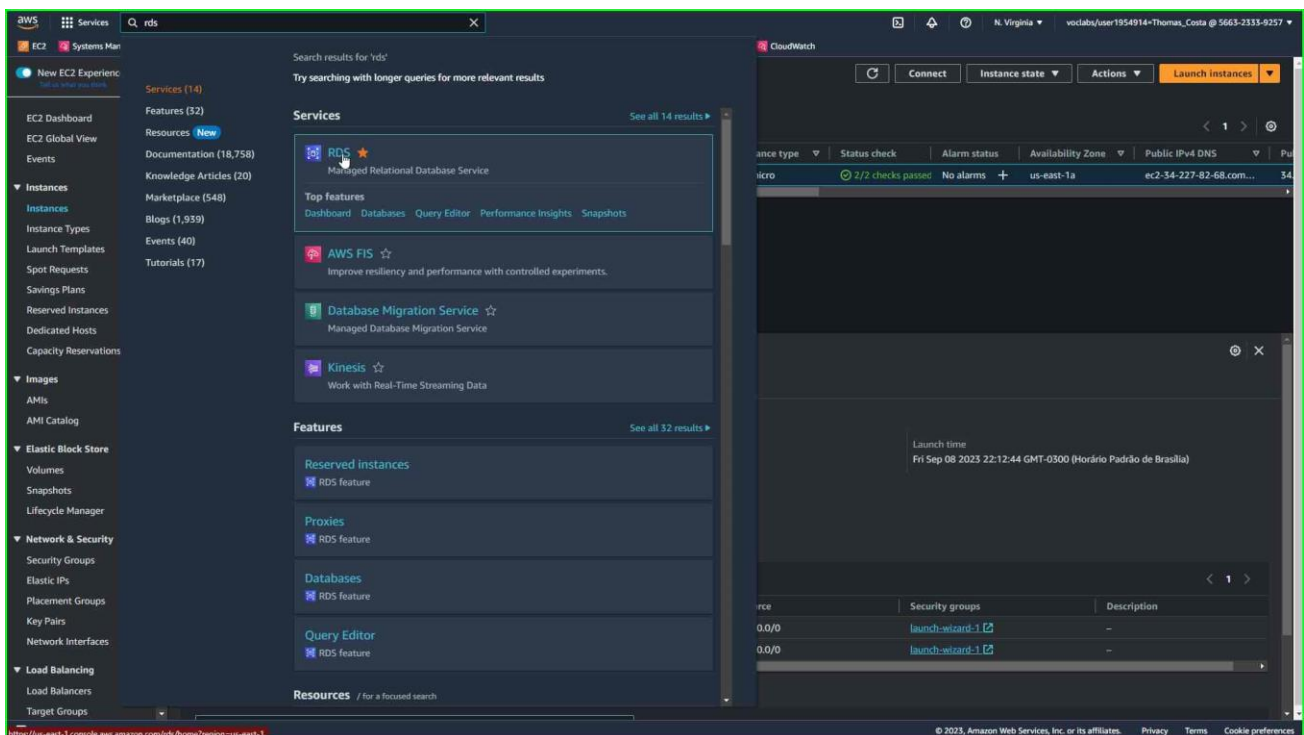
Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-080ead548af6c69f74	8080	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-06ade819b762bd74c	22	TCP	0.0.0.0/0	launch-wizard-1	-

Implantando um banco de dados MySQL com AWS RDS

Nome copiado:

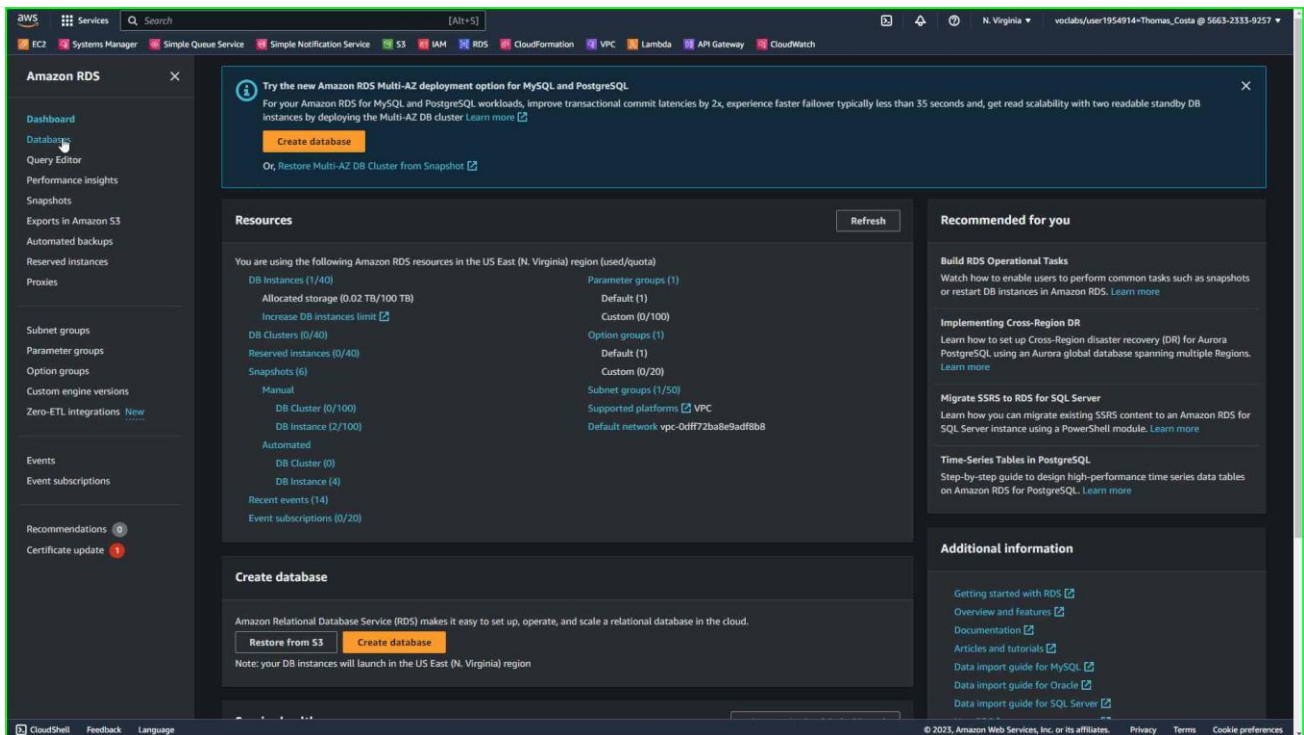


Volte para o banco de dados criado em RDS:

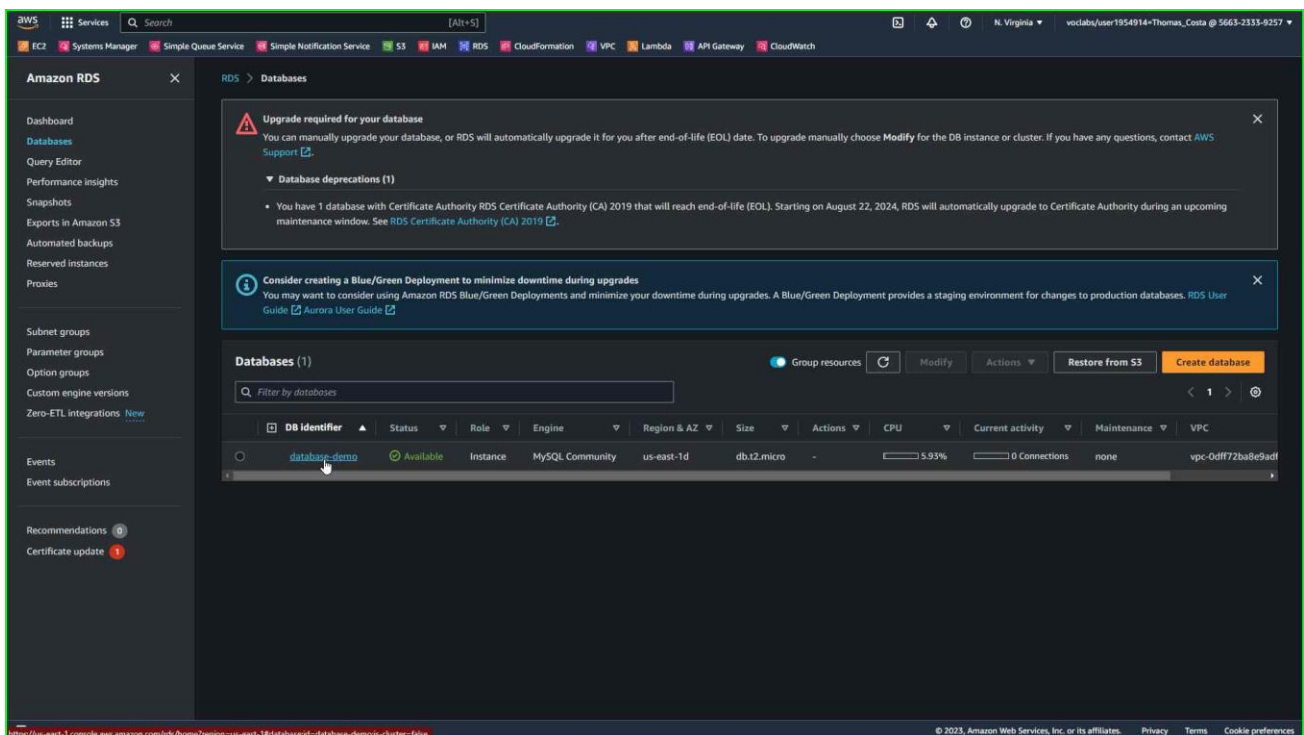


Implantando um banco de dados MySQL com AWS RDS

Selecione a opção “Databases”:

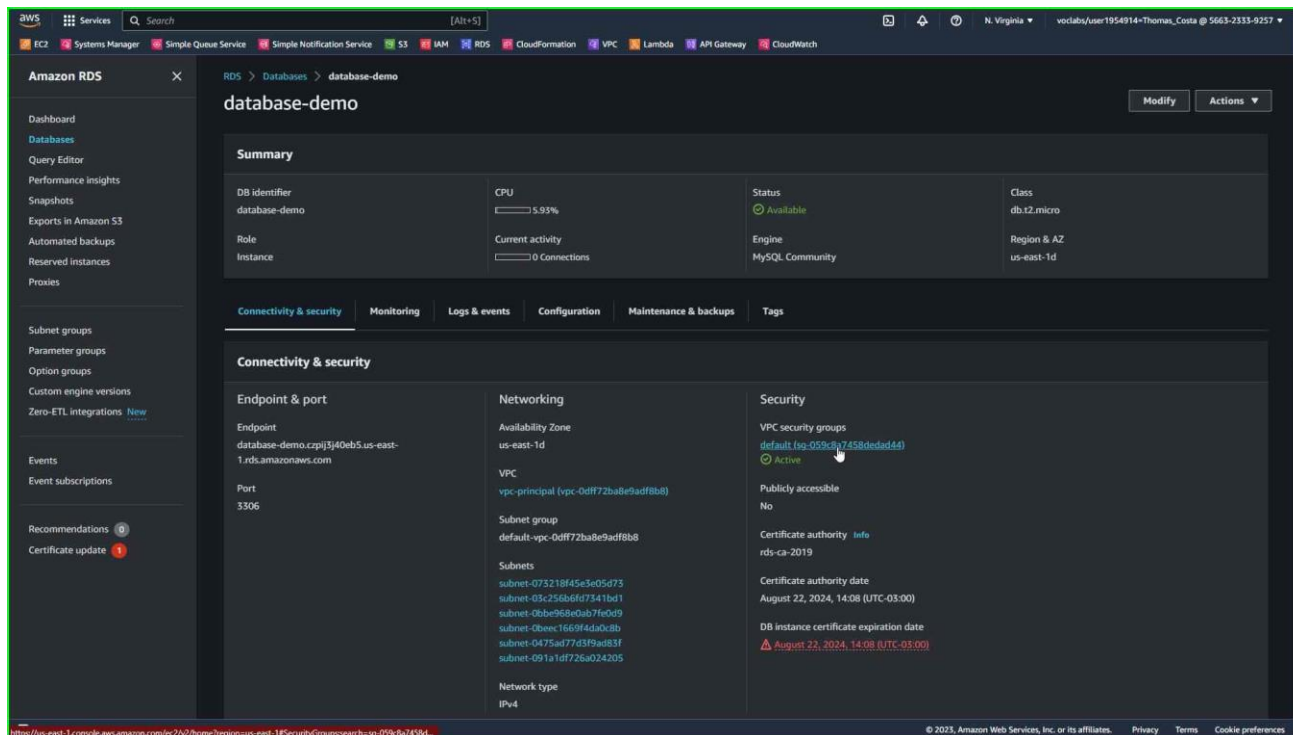


Clique no banco de dados criado:

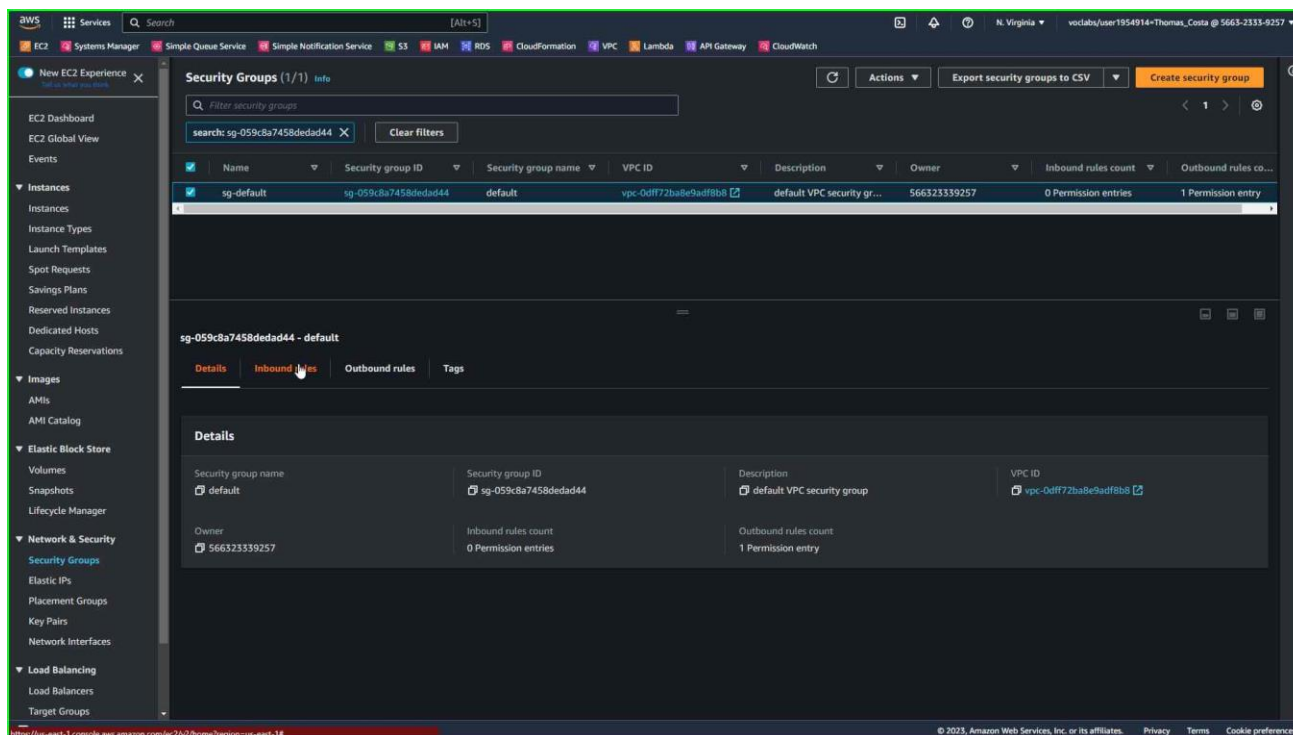


Implantando um banco de dados MySQL com AWS RDS

Clique em “VPC security groups”:

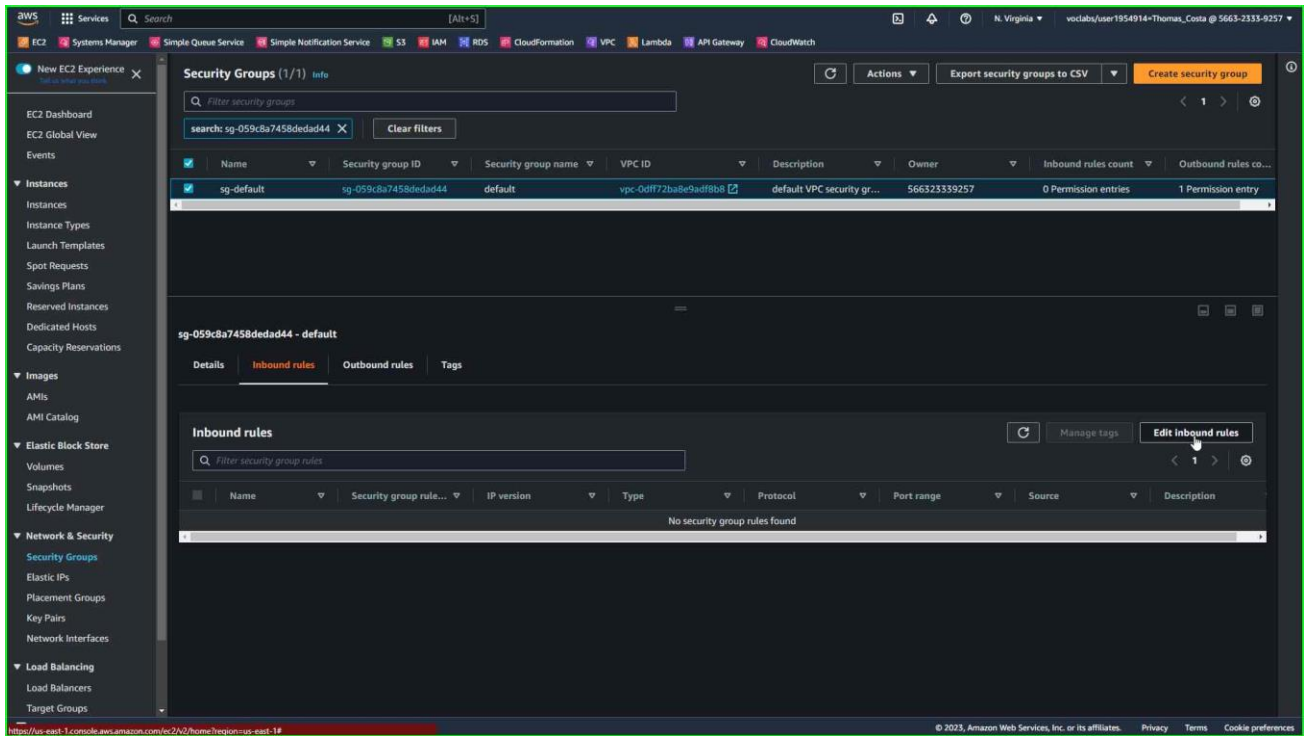


Clique em “Inbound rules”:

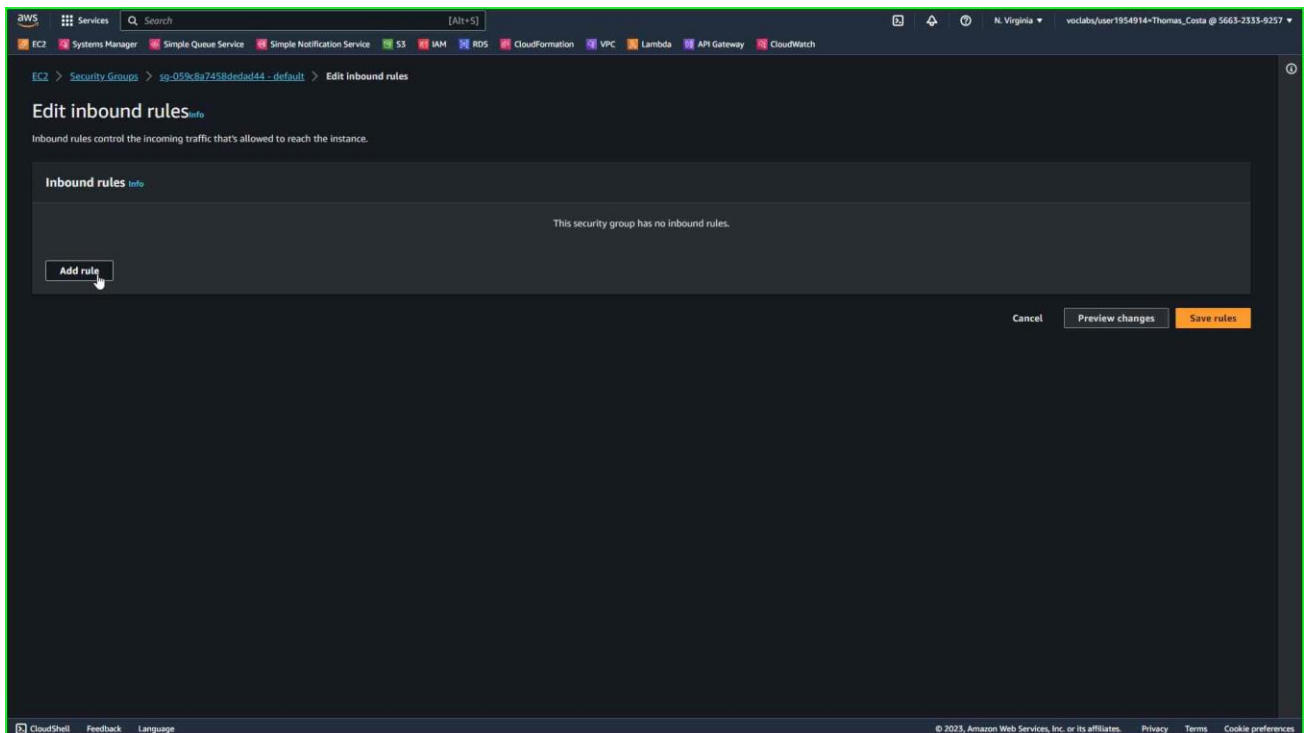


Implantando um banco de dados MySQL com AWS RDS

Clique em “Edit inbound rules”:

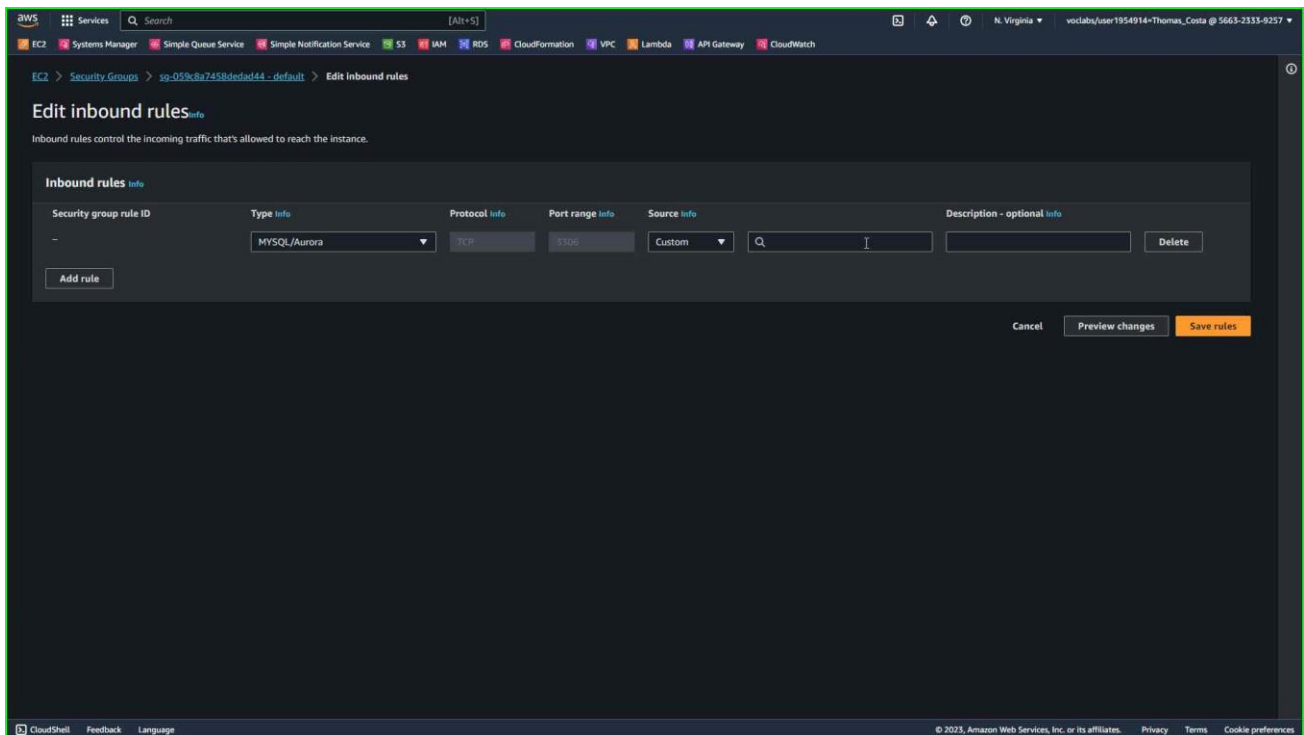


Clique em “Add rule”:

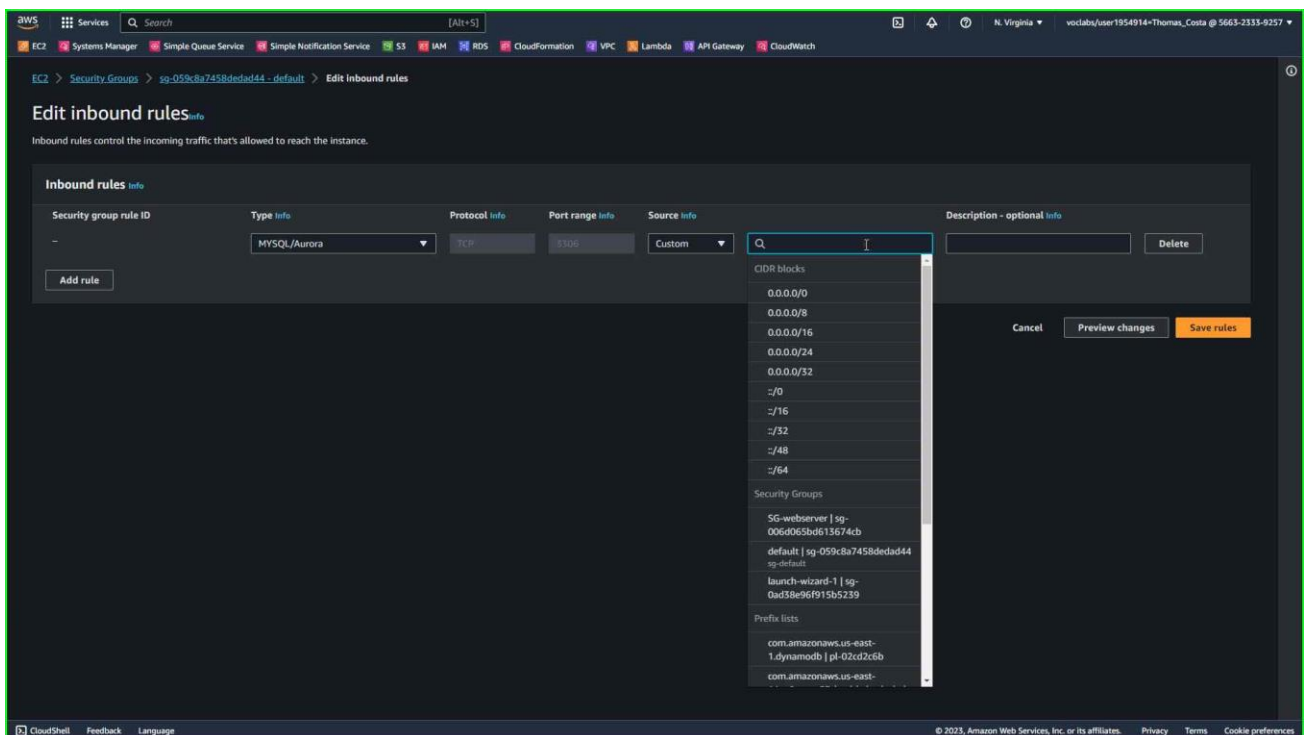


Implantando um banco de dados MySQL com AWS RDS

Selecione as opções abaixo:

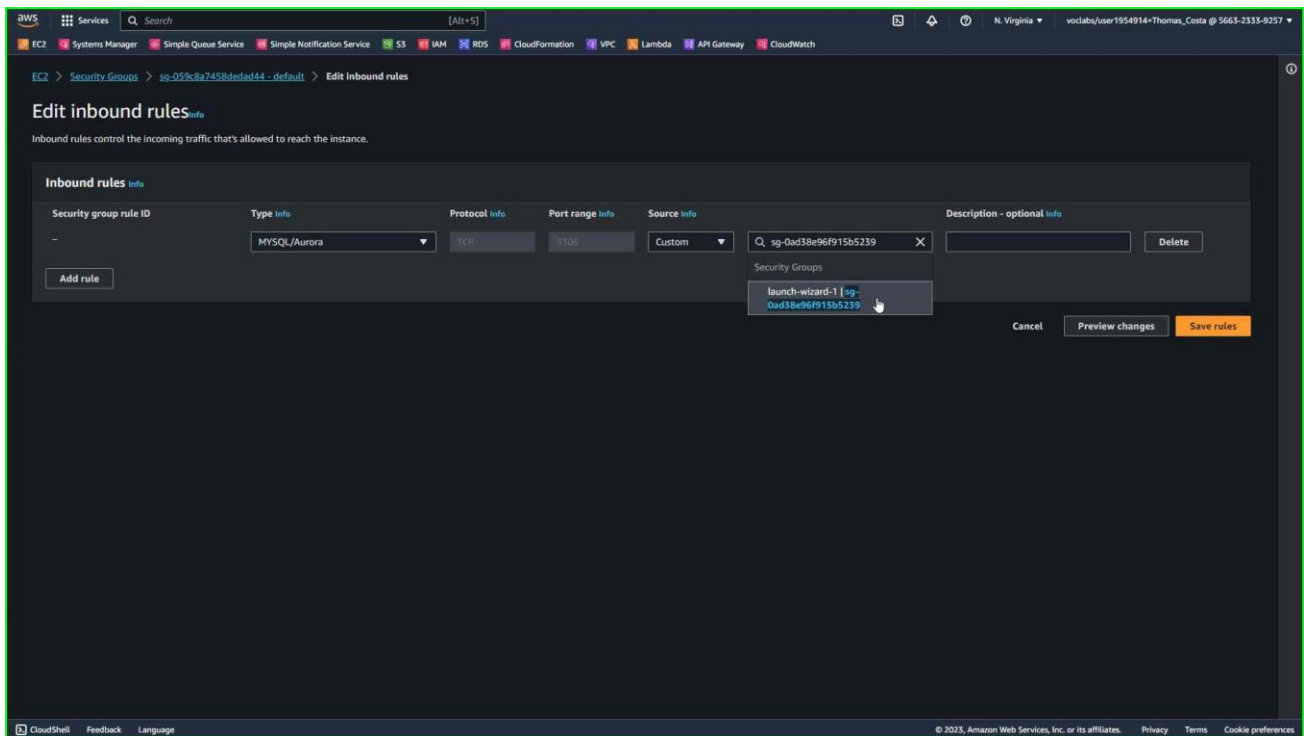


No campo abaixo selecione o "Security group" do EC2:

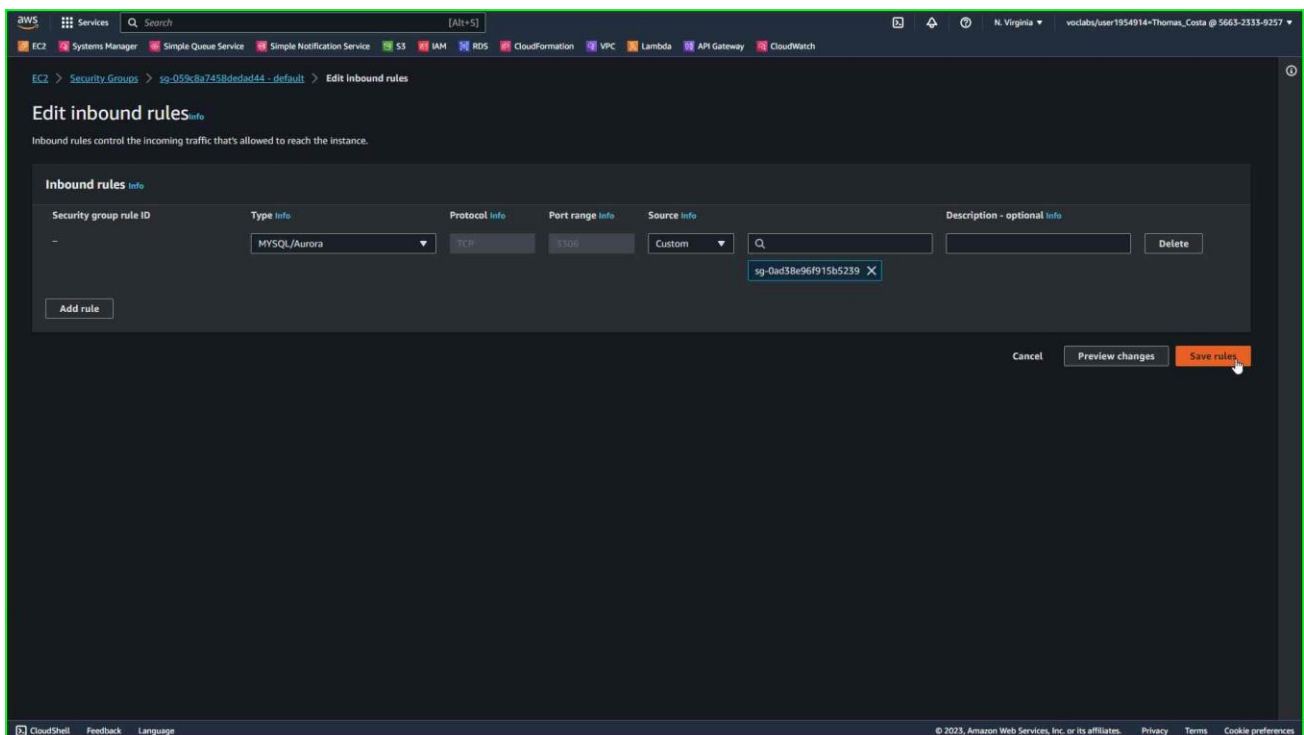


Implantando um banco de dados MySQL com AWS RDS

Copie o nome selecionado do “Security group” do EC2:

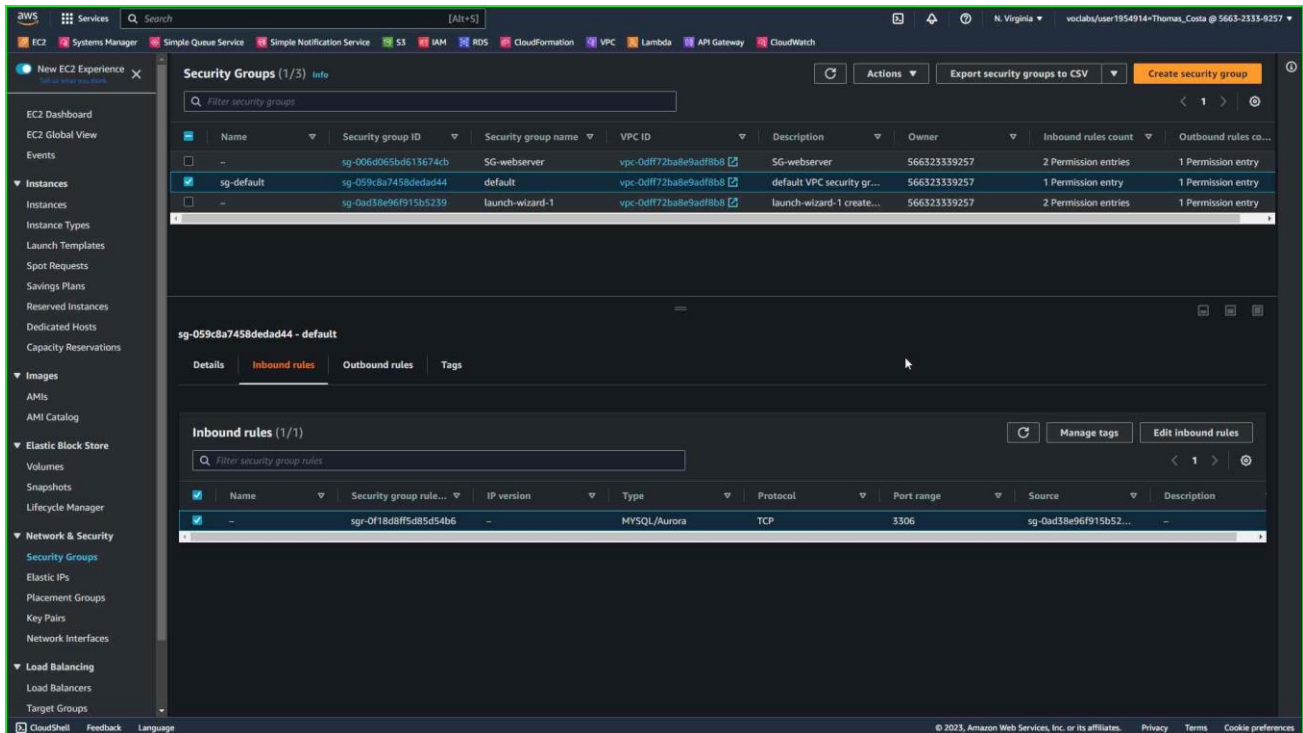


Clique em “Save rules”:

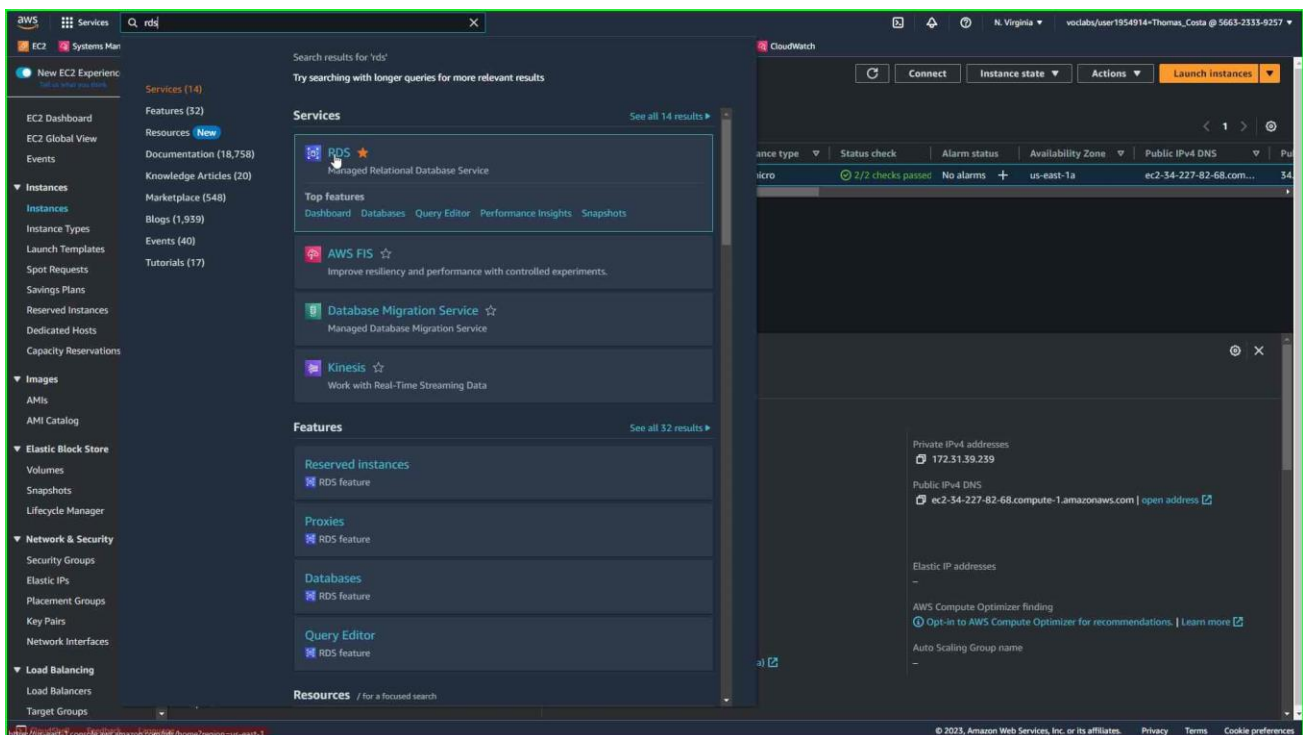


Implantando um banco de dados MySQL com AWS RDS

Security group alterado com sucesso:

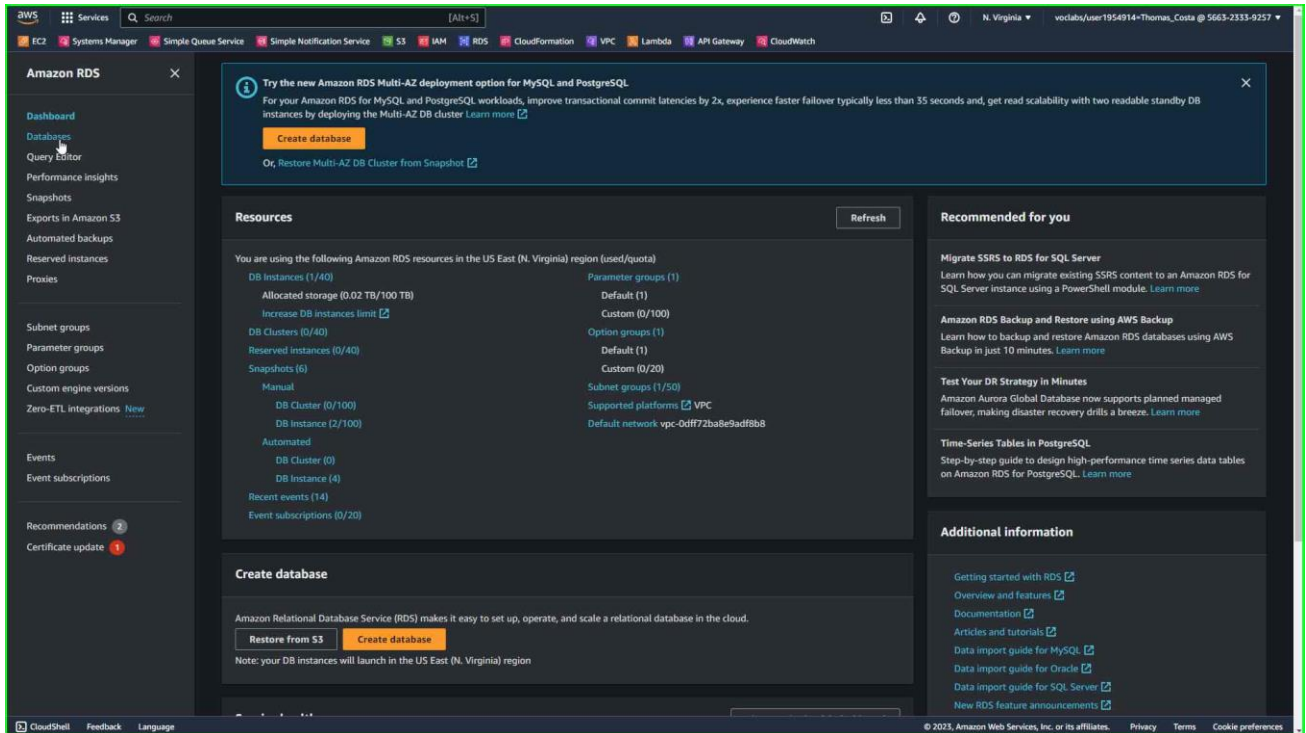


Volte para o banco de dados criado em RDS:



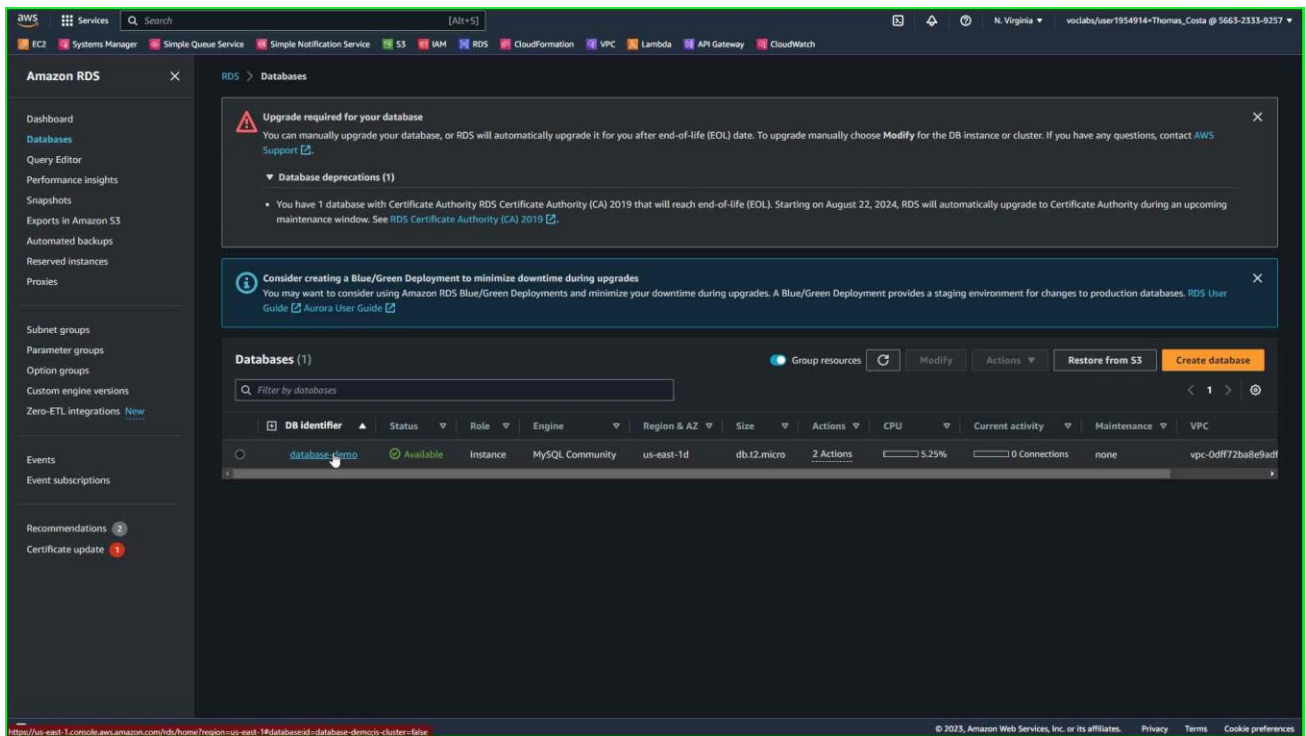
Implantando um banco de dados MySQL com AWS RDS

Selecione a opção “Databases”:



The screenshot shows the Amazon RDS console interface. On the left, the navigation menu includes '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', 'Recommendations', and 'Certificate update'. The main content area displays a 'Resources' section with a 'Refresh' button, a 'Create database' section with 'Restore from S3' and 'Create database' buttons, and a 'Recommended for you' section with various links. A 'Create database' button is highlighted in orange.

Clique no banco de dados criado:



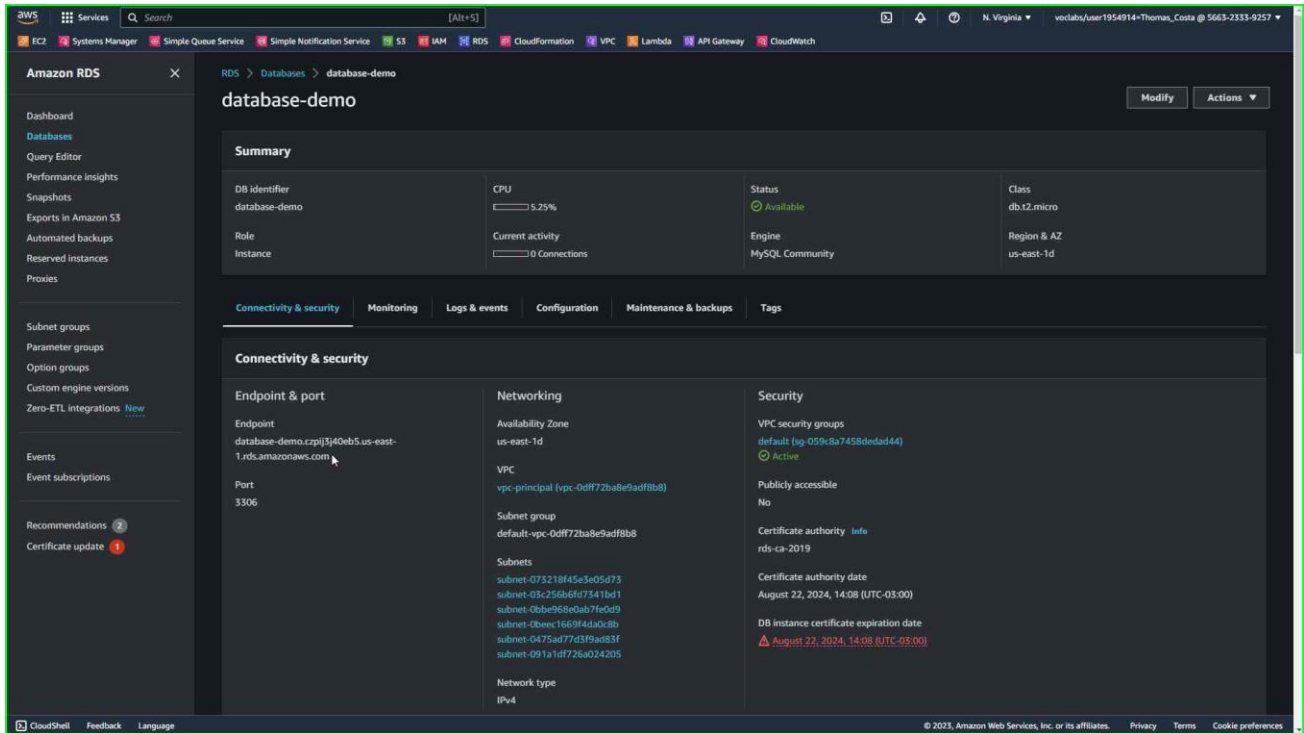
The screenshot shows the Amazon RDS console interface with the 'Databases' section selected. A notification banner at the top indicates 'Upgrade required for your database'. Below it, a 'Consider creating a Blue/Green Deployment' notification is visible. The 'Databases (1)' section shows a table with one database instance:

DB Identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance	VPC
database-1	Available	Instance	MySQL Community	us-east-1d	db.t2.micro	2 Actions	5.25%	0 Connections	none	vpc-0dff72ba8e9adfd

The 'database-1' instance is highlighted in the table. The 'Create database' button is highlighted in orange.

Implantando um banco de dados MySQL com AWS RDS

Copie o endereço do banco de dados em “Endpoint & port”. Este endereço será utilizado no aplicativo desenvolvido em Spring Boot para acessar o banco de dados:



The screenshot displays the Amazon RDS console for a MySQL instance named 'database-demo'. The interface is divided into several sections:

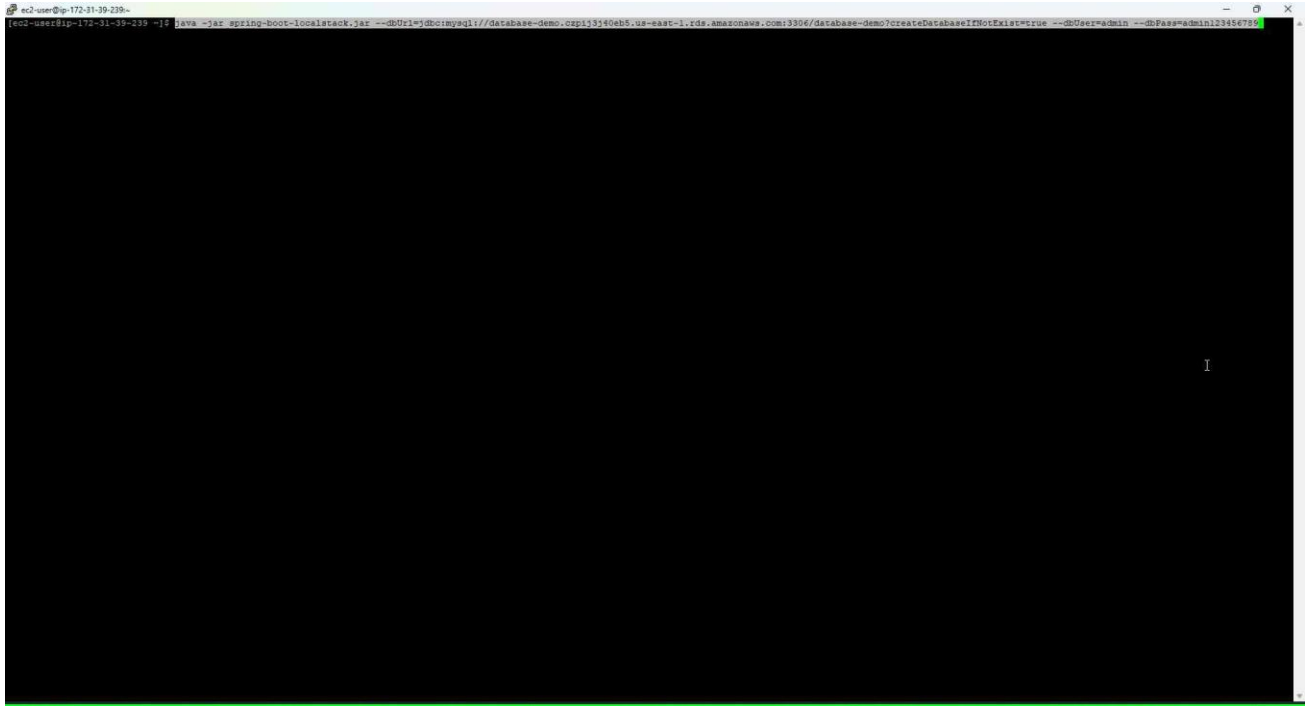
- Summary:** Shows the DB identifier as 'database-demo', CPU usage at 5.25%, Status as 'Available', and Class as 'db.t2.micro'. The Role is 'Instance' with 0 connections, and the Engine is 'MySQL Community'. The Region & AZ is 'us-east-1d'.
- Connectivity & security:** This section is expanded to show:
 - Endpoint & port:** The endpoint is 'database-demo.czpjj3j40eb5.us-east-1.rds.amazonaws.com' and the port is '3306'.
 - Networking:** The instance is in the 'us-east-1d' Availability Zone, connected to the 'vpc-principal' VPC. It is associated with the 'default-vpc-0dff72ba8e9adff8b8' Subnet group, which includes subnets: 'subnet-073218f45e5e05d73', 'subnet-03c256b69f7341bd1', 'subnet-0bbe96889ab7fe0e9', 'subnet-0b6ec1669f4da0c8b', 'subnet-0475ad77d3f9ad83f', and 'subnet-091a1df726a024205'. The network type is 'IPv4'.
 - Security:** The VPC security groups are 'default (sg-059c8a7458dedad44)', which is 'Active'. It is not 'Publicly accessible'. The Certificate authority is 'rds-ca-2019'. The Certificate authority date is 'August 22, 2024, 14:08 (UTC-03:00)'. The DB instance certificate expiration date is 'August 22, 2024, 14:08 (UTC-03:00)'.

Implantando um banco de dados MySQL com AWS RDS

Conecte na máquina EC2 e execute o Spring Boot com o seguinte comando:

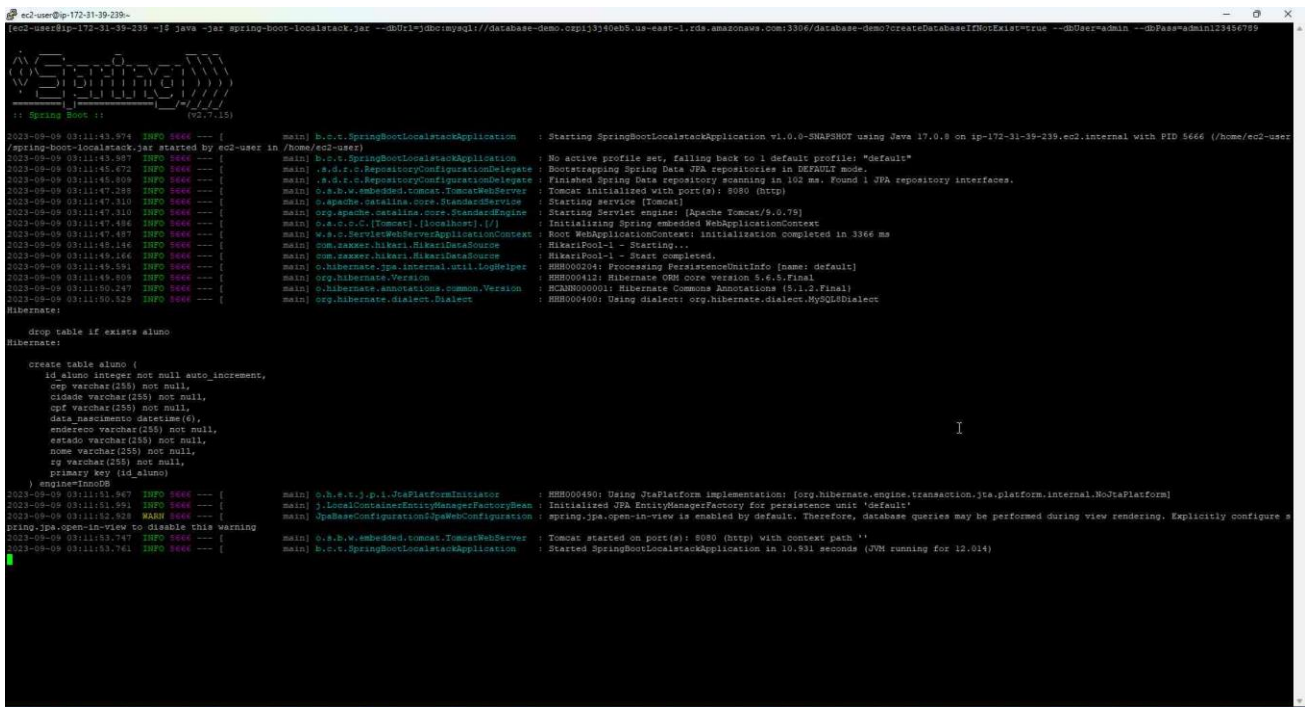
```
“java -jar spring-boot-localstack.jar --dbUrl=jdbc:mysql://database-demo.czpij3j40eb5.us-east-1.rds.amazonaws.com:3306/database-demo?createDatabaseIfNotExist=true --dbUser=admin --dbPass=admin123456789”
```

Coloque a URL, nome do banco, usuário e senha configurados no banco de dados:



```
ec2-user@ip-172-31-39-239: ~$ java -jar spring-boot-localstack.jar --dbUrl=jdbc:mysql://database-demo.czpij3j40eb5.us-east-1.rds.amazonaws.com:3306/database-demo?createDatabaseIfNotExist=true --dbUser=admin --dbPass=admin123456789
Spring-Boot-Localstack-Jar started by ec2-user in /home/ec2-user
2023-08-09 03:11:43.974 INFO 5868 --- [main] b.o.s.c.SpringBootLocalstackApplication : Starting SpringBootLocalstackApplication v1.0.0-SNAPSHOT using Java 17.0.8 on ip-172-31-39-239.ec2.internal with PID 5666 (/home/ec2-user
2023-08-09 03:11:43.997 INFO 5868 --- [main] b.o.s.c.SpringBootLocalstackApplication : No active profile set, falling back to 1 default profile: "default"
2023-08-09 03:11:44.472 INFO 5868 --- [main] .s.d.f.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-08-09 03:11:44.908 INFO 5868 --- [main] .s.d.f.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 162 ms. Found 1 JPA repository interfaces.
2023-08-09 03:11:47.058 INFO 5868 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-08-09 03:11:47.310 INFO 5868 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-08-09 03:11:47.310 INFO 5868 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.78]
2023-08-09 03:11:47.484 INFO 5868 --- [main] w.s.c.s.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-08-09 03:11:47.487 INFO 5868 --- [main] w.s.c.s.c.WebServerApplicationContext : Root WebApplicationContext: initialization completed in 3366 ms
2023-08-09 03:11:48.186 INFO 5868 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-08-09 03:11:48.186 INFO 5868 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-08-09 03:11:49.591 INFO 5868 --- [main] org.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2023-08-09 03:11:49.808 INFO 5868 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.5.Final
2023-08-09 03:11:50.247 INFO 5868 --- [main] org.hibernate.annotations.common.Version : HCANN000004: Hibernate Commons Annotations 5.1.2.Final
2023-08-09 03:11:50.529 INFO 5868 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Hibernate:
drop table if exists aluno
Hibernate:
create table aluno (
  id_aluno integer not null auto_increment,
  cpf varchar(255) not null,
  cidade varchar(255) not null,
  cpf varchar(255) not null,
  data_nascimento date(4),
  endereco varchar(255) not null,
  estado varchar(255) not null,
  nome varchar(255) not null,
  rg varchar(255) not null,
  primary key (id_aluno)
) engine=InnoDB
2023-08-09 03:11:51.967 INFO 5868 --- [main] o.h.e.s.j.p.i.JpaPlatformInitiator : HHH000490: Using JpaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2023-08-09 03:11:51.991 INFO 5868 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2023-08-09 03:11:52.528 WARN 5868 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure a
spring.jpa.open-in-view to disable this warning
2023-08-09 03:11:53.747 INFO 5868 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-08-09 03:11:53.761 INFO 5868 --- [main] b.o.s.c.SpringBootLocalstackApplication : Started SpringBootLocalstackApplication in 10.931 seconds (JVM running for 12.014)
```

Microsserviço em execução com sucesso:



```
ec2-user@ip-172-31-39-239: ~$ java -jar spring-boot-localstack.jar --dbUrl=jdbc:mysql://database-demo.czpij3j40eb5.us-east-1.rds.amazonaws.com:3306/database-demo?createDatabaseIfNotExist=true --dbUser=admin --dbPass=admin123456789
Spring Boot v1.0.0-SNAPSHOT
2023-08-09 03:11:43.974 INFO 5868 --- [main] b.o.s.c.SpringBootLocalstackApplication : Starting SpringBootLocalstackApplication v1.0.0-SNAPSHOT using Java 17.0.8 on ip-172-31-39-239.ec2.internal with PID 5666 (/home/ec2-user
2023-08-09 03:11:43.997 INFO 5868 --- [main] b.o.s.c.SpringBootLocalstackApplication : No active profile set, falling back to 1 default profile: "default"
2023-08-09 03:11:44.472 INFO 5868 --- [main] .s.d.f.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2023-08-09 03:11:44.908 INFO 5868 --- [main] .s.d.f.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 162 ms. Found 1 JPA repository interfaces.
2023-08-09 03:11:47.058 INFO 5868 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2023-08-09 03:11:47.310 INFO 5868 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-08-09 03:11:47.310 INFO 5868 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.78]
2023-08-09 03:11:47.484 INFO 5868 --- [main] w.s.c.s.c.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-08-09 03:11:47.487 INFO 5868 --- [main] w.s.c.s.c.WebServerApplicationContext : Root WebApplicationContext: initialization completed in 3366 ms
2023-08-09 03:11:48.186 INFO 5868 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2023-08-09 03:11:48.186 INFO 5868 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2023-08-09 03:11:49.591 INFO 5868 --- [main] org.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2023-08-09 03:11:49.808 INFO 5868 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.5.Final
2023-08-09 03:11:50.247 INFO 5868 --- [main] org.hibernate.annotations.common.Version : HCANN000004: Hibernate Commons Annotations 5.1.2.Final
2023-08-09 03:11:50.529 INFO 5868 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Hibernate:
drop table if exists aluno
Hibernate:
create table aluno (
  id_aluno integer not null auto_increment,
  cpf varchar(255) not null,
  cidade varchar(255) not null,
  cpf varchar(255) not null,
  data_nascimento date(4),
  endereco varchar(255) not null,
  estado varchar(255) not null,
  nome varchar(255) not null,
  rg varchar(255) not null,
  primary key (id_aluno)
) engine=InnoDB
2023-08-09 03:11:51.967 INFO 5868 --- [main] o.h.e.s.j.p.i.JpaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2023-08-09 03:11:51.991 INFO 5868 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2023-08-09 03:11:52.528 WARN 5868 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure a
spring.jpa.open-in-view to disable this warning
2023-08-09 03:11:53.747 INFO 5868 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2023-08-09 03:11:53.761 INFO 5868 --- [main] b.o.s.c.SpringBootLocalstackApplication : Started SpringBootLocalstackApplication in 10.931 seconds (JVM running for 12.014)
```