

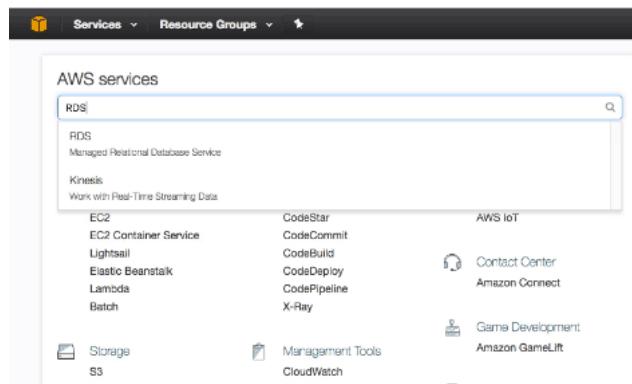
05

Configurando o Amazon RDS

Transcrição

O serviço da Amazon que ajuda atener as questões discutidas no vídeo anterior se chama RDS (Relational Data Service).

O RDS é um serviço a parte, independente do EC2 e por isso devemos buscá-lo separadamente:



RDS faz parte da categoria SaaS (Software as a Service) e a tarefa dele é fornecer uma banco de dados relacionais na nuvem pronto para usar, sem se preocupar com detalhes da instalação ou configuração da hardware

Primeiros passos com RDS

Para começar com RDS basta clicar no botão *Get Started Now* ou no menu *Instances* clicar no botão "Launch DB Instance":



Step 1 - Qual banco?

A partir dai já começa a configuração do banco e no formulário devemos escolher qual banco gostaríamos de configurar. No nosso caso ficamos com o free tier usando o MySQL, mas também poderíamos usar Oracle, Postgres, SQLServer, MariaDB ou Aurora (que é um banco próprio da Amazon).

Step 1: Select Engine

...

 Free tier eligible only 

Select Engine

To get started, choose a DB Engine below and click Select.



MySQL

MySQL Community Edition

 Free tier eligible Select

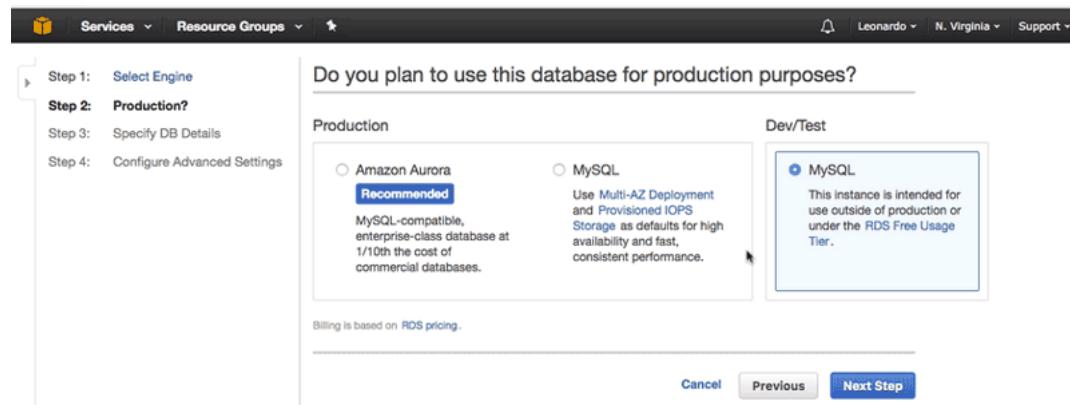
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 6 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.

Step 2 - Qual ambiente?

O próximo passo é definir se usamos o MySQL em um ambiente de produção ou para um ambiente de desenvolvimento/teste. Ambientes de produção podem usar deployments em vários datacenters na mesma região (*availability zone*). Nesses casos a Amazon automaticamente replica os dados para aumentar a disponibilidade do banco de dados. Além disso, a Amazon dá garantias de uma quantidade de operações na leitura e escrita (*Provisioned IOPS*).

Como estamos usando os serviço gratuito devemos escolher o ambiente Dev/Test:



Do you plan to use this database for production purposes?

Production

Amazon Aurora **Recommended**
MySQL-compatible, enterprise-class database at 1/10th the cost of commercial databases.

MySQL
Use Multi-AZ Deployment and Provisioned IOPS Storage as defaults for high availability and fast, consistent performance.

Dev/Test

MySQL
This instance is intended for use outside of production or under the RDS Free Usage Tier.

Billing is based on RDS pricing.

Cancel Previous Next Step

Step 3 - Qual configuração?

Nesse passo devemos configurar detalhes do banco de dados como o tamanho e tipo do volume (HD), tipo de instancia, licença e versão. Aqui também já tem uma opção para filtrar as opções do *free tier* que vamos escolher:

Step 2: Production?

Step 3: Specify DB Details

Step 4: Configure Advanced Settings

Free Tier

The Amazon RDS Free Tier provides a single db.t2.micro instance as well as up to 20 GB of storage, allowing new AWS customers to gain hands-on experience with Amazon RDS. Learn more about the RDS Free Tier and the instance restrictions [here](#).

Only show options that are eligible for RDS Free Tier

Instance Specifications

DB Engine: mysql

License Model: general-public-license

DB Engine Version: MySQL 5.6.27

DB Instance Class: - Select One -

Multi-AZ Deployment: - Select One -

Storage Type: - Select One -

Allocated Storage*: 100 GB

Seguem os detalhes:

Only show options that are eligible for RDS Free Tier

Instance Specifications

DB Engine: mysql

License Model: general-public-license

DB Engine Version: MySQL 5.6.27

Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.

DB Instance Class: db.t2.micro — 1 vCPU, 1 GiB RAM

Multi-AZ Deployment: No

Storage Type: General Purpose (SSD)

Allocated Storage*: 10 GB

E a configuração do usuário e senha, no nosso caso usaremos root e a senha catalogo :

Settings

DB Instance Identifier*: mysql-catalogo

Master Username*: root

Master Password*:

Confirm Password*:

Step 4 - Configurações avançadas

Por fim devemos configurar o acesso via rede. Vamos aproveitar as configurações padrões e usar o VPC (*Virtual Private Cloud*) já configurado. Importante é que deixamos o MySQL com acesso público:

Network & Security

VPC*: Default VPC (vpc-e9c6fc8f)

Subnet Group: default

Publicly Accessible: Yes

Availability Zone: No Preference

VPC Security Group(s): Create new Security Group, default (VPC), launch-wizard-1 (VPC), launch-wizard-2 (VPC)

Ainda na mesma página devemos preparar o nosso banco (esquema) no MySQL. Como nome do banco a aplicação já define projeto_jpa , então vamos usar esse também. A porta deve ser 3306 .

Database Options

Database Name

Note: If no database name is specified then no initial MySQL database will be created on the DB instance.

Database Port

DB Parameter Group

Option Group

Copy Tags To Snapshots

Enable Encryption

Por fim, falta configurar a frequência do backup e manutenção. Vamos manter 7 dias, com uma janela das 22:00s até 23:00:

Backup

Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to detail [here](#).

Backup Retention Period days

Backup Window

Start Time

Duration hours

Também vamos autorizar atualizações automáticas de versões *minors*:

Maintenance

Auto Minor Version Upgrade

Maintenance Window

Start Day

Start Time

Duration hours

Finalizadas as configurações vamos clicar no botão "Launch DB Instance" para inicializar a criação do banco de dados. No próximo vídeo vamos acessar o banco! Te vejo lá :)