



## Infraestrutura II

**Atividade obrigatória e individual**

**Dificuldade: média**

# PARTE 1: Monitoramos um RDS

Nesta prática vamos construir a infraestrutura necessária para poder usar um banco de dados no AWS, faremos um teste nele e o monitoramos com o CloudWatch.

## Do que precisamos?

Recursos necessários para esta prática:

- Terraform instalado ([ajuda](#)).
- Mysql-client instalado (mariadb) ([ajuda](#)).
- Bash ([windows](#)).
- AWS-CLI ([ajuda](#)).
- Acesso a conta da AWS.
- ZIP com os arquivos para a prática 1 ([link](#)).

Para começar, configuramos as credenciais da AWS para poder usá-las com o Terraform.

Nós logamos no seguinte [link](#), vamos a “AWS Account” e clicamos no botão “AWS Educate Starter Account”. Depois, vamos no botão “Account Details” e outra vez em “Show”. Vamos copiar o conteúdo do texto, onde nos diz “~/.aws/credentials”.

Para testar se já temos as credenciais bem configuradas, podemos executar um “/s” do S3 com o seguinte comando: “aws s3 ls”. Caso funcione, configuramos corretamente.

## Seguindo!

### Levantando nossa infraestrutura

Vamos utilizar os módulos do Terraform para implementar uma VPC (Virtual Private Cloud) e um RDS (Relational Database Service).

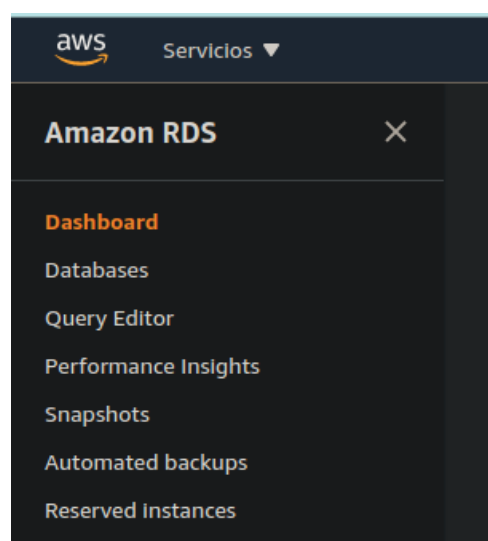
Uma vez extraído e descomprimido o ZIP, vamos para a pasta que foi gerada e depois para o Terraform. Lá executamos o código com as seguintes instruções:

```
$ terraform init
$ terraform apply
```

Ele nos pedirá uma confirmação, ao qual escrevemos "yes". No final veremos a seguinte mensagem:

```
module.db.module.db_instance.aws_db_instance.this[0]: Creation complete
Apply complete! Resources: 15 added, 0 changed, 0 destroyed.
```

Isso deixará um RDS logando na conta AWS. Vemos isso quando entramos no console da AWS e procuramos RDS. Em seguida, no menu à esquerda, vamos para Databases:



RDS > Databases

**Databases** Group resources Refresh Modify Actions Restore from S3 Create database

Filter databases

DB Identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maint
demodb	Instance	MySQL Community	us-east-1b	db.t2.small	Creating	-		none

DB Identifier	Role	Engine	Region & AZ	Size	Status
demodb	Instance	MySQL Community	us-east-1b	db.t2.small	Available

Quando mostra disponível (available), podemos conectar. Precisamos do DNS.

Para tal, clicamos no nome do nosso RDS "demodb". Na seção "Conectividade e segurança", copiamos o DNS que aparecerá em "Endpoint".

Executamos o seguinte comando para entrar. Mudamos "DNS" para o valor obtido anteriormente: "\$ mysql -h dns -uuser -pdemouser!"

Para ver os bancos de dados existentes, executamos o comando "show databases;" e para sair "quit;"

```
root@f9d41f32318f:~/DH/part1/terraform# mysql -h demodb.co9brqybipc9.us-east-1.rds.amazonaws.com -uuser -pdemouser!
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.7.19 MySQL Community Server (GPL)

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

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

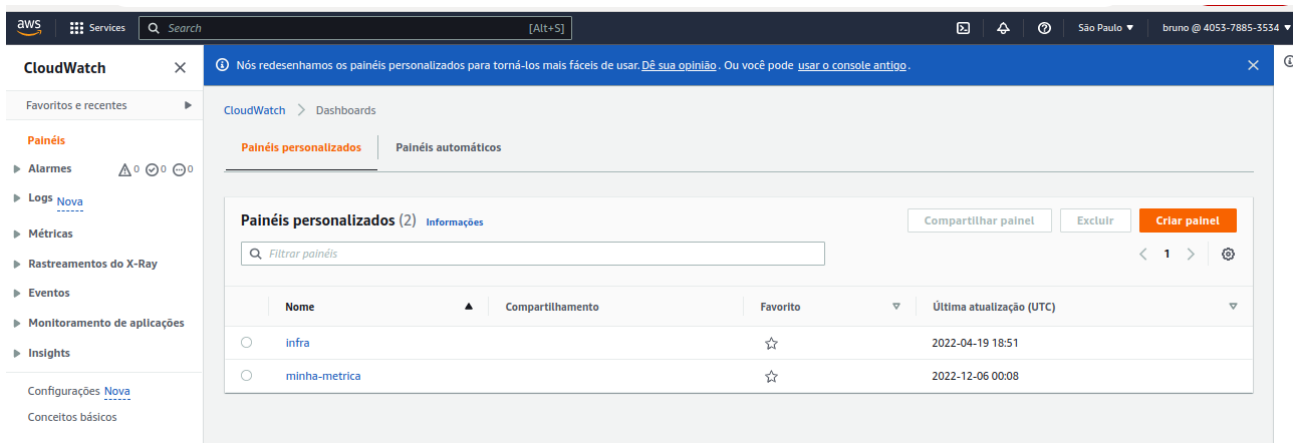
MySQL [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| demodb |
| innodb |
| mysql |
| performance_schema |
| sys |
+-----+
6 rows in set (0.165 sec)

MySQL [(none)]> QUIT;
Bye
root@f9d41f32318f:~/DH/part1/terraform#
```

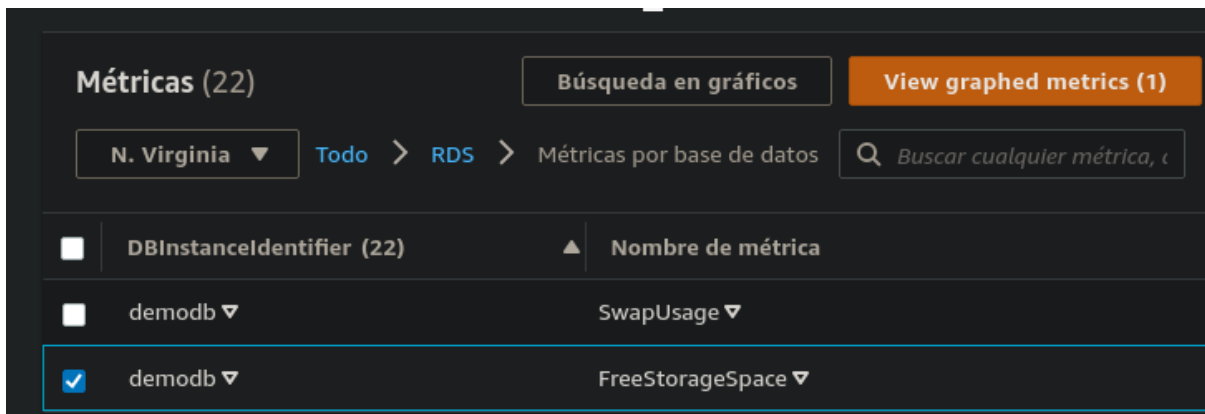
## Criando um dashboard simples

Enquanto o RDS incorpora alguns gráficos métricos do CloudWatch, **vamos criar nosso próprio dashboard!**

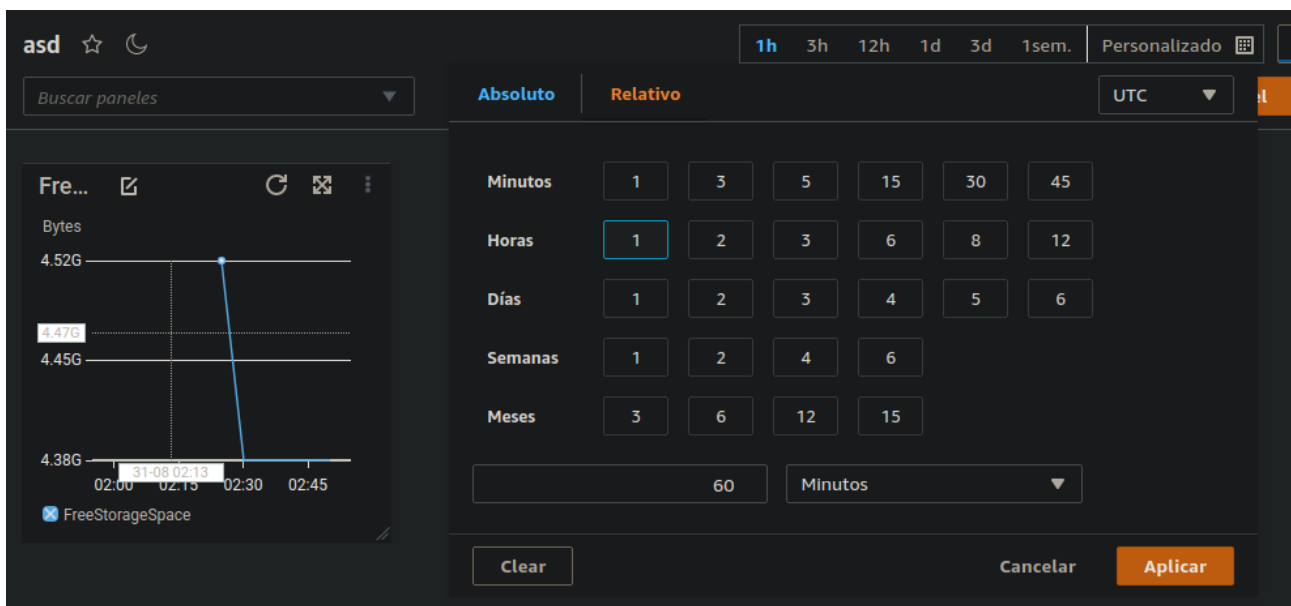
No console da AWS, vamos para o serviço CloudWatch. Em Dashboards ou Painéis, clicamos em "Criar um painel". Damos um nome e continuamos. Em "Adicionar um widget", selecionamos "Linha" e depois "Métricas":



Na parte inferior da caixa de diálogo estão as métricas. Procuramos RDS e depois "Métricas por banco de dados". Na lista de métricas que vemos, selecionamos "FreeStorageSpace" com o nome do nosso RDS. Como podemos ver na imagem a seguir:



Para finalizar a criação do widget, clique no botão "Criar um widget". Ao clicar em "Personalizar" - barra superior direita - podemos ampliar nosso gráfico arrastando-o de um canto e alterando o intervalo de tempo para 30 minutos.



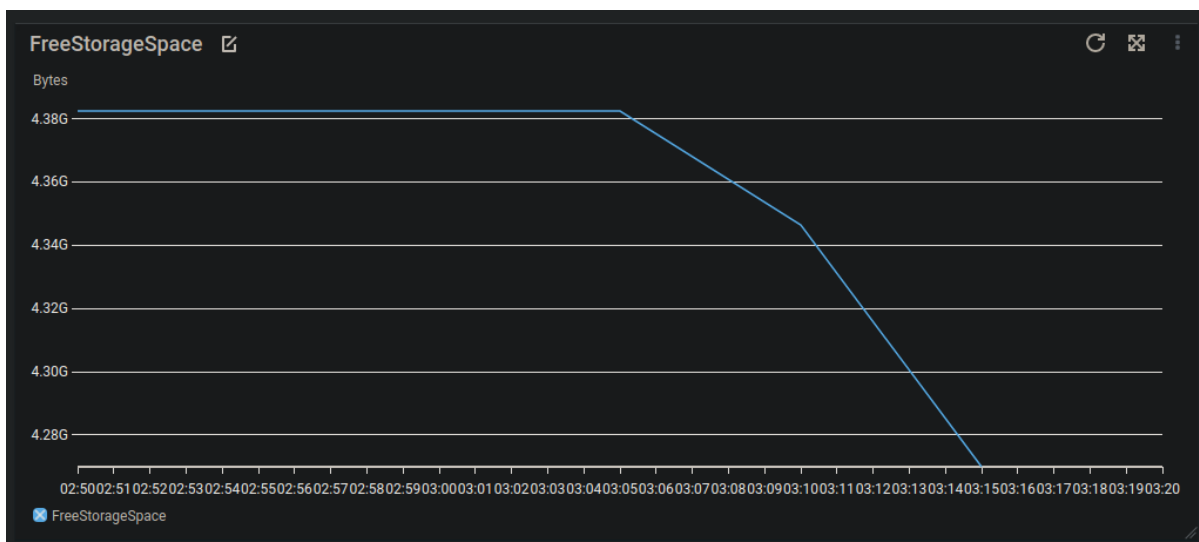
## Executando um teste de carga

Vamos executar um script que ocupa espaço na base de dados para poder monitorar como nosso dashboard é executado.

Localizado na pasta que foi gerada ao descompactar o ZIP, editamos o arquivo ".env" adicionando o endereço DNS do nosso RDS:

```
part2 > .env
TEST_NAME="AutoScalingTest"
DNS_DB=""
USER="user"
PASSWORD="demouser!"
NUM_DBS=1
NUM_TABLES=5
NUM_USERS=20
```

Executamos o script de carregamento executando: "bash runTest.sh" e vamos ao nosso painel para ver o comportamento. Depois de um tempo, podemos ver o seguinte resultado em nosso gráfico:



Notamos que o espaço disponível estava diminuindo no decorrer da execução do script.

## Limpendo o ambiente

Assim que a prática estiver concluída, vamos deletar nosso painel clicando em "Ações" e depois em "Deletar". Removeremos o RDS executando o comando "terraform destroy" e depois "yes". **A AWS nos cobra pelo uso** e sempre que não usarmos os recursos, devemos **excluí-los**.

## Conclusão

Nesta prática, testamos a carga de um banco de dados AWS RDS e monitoramos a saúde do disco com o AWS CloudWatch. Embora não tenhamos deixado o disco RDS sem espaço, é essencial monitorar sempre os discos, pois se ficarmos sem espaço **o sistema host e nosso banco de dados ou serviço irá parar de funcionar**. Isso compromete a disponibilidade do cliente, podendo causar até problemas contratuais e multas dependendo do tempo de indisponibilidade.