



# Certified Tech Developer

The Ultimate Degree

## Infraestructura II

**Actividad obligatoria e individual**

**Dificultad: media**

# PARTE 1: Monitoreamos un RDS

En esta práctica vamos a levantar la infraestructura necesaria para poder usar una base de datos en AWS, correremos una prueba en ella y la monitoreamos con CloudWatch.

## ¿Qué necesitamos?

Recursos necesarios para esta práctica:

- Terraform instalado ([ayuda](#)).
- Mysql-client instalado (mariadb) ([ayuda](#)).
- Bash ([windows](#)).
- AWS-CLI ([ayuda](#)).
- Acceso a la cuenta de AWS.
- ZIP con los archivos para la práctica 1 ([link](#)).

Para iniciar, configuramos las credenciales de AWS para poder usarlas con Terraform.

Nos logueamos en el siguiente [link](#), vamos a "AWS Account" y hacemos clic en el botón "AWS Educate Starter Account". Luego en el botón "Account Details" y otra vez en "Show". Copiaremos el contenido del texto, donde nos indica "~/aws/credentials".

Para probar que ya tenemos las credenciales bien configuradas podemos correr un "ls" de S3 con el siguiente comando: "aws s3 ls". Si funciona, fue configurado correctamente.

## ¡Manos a la obra!

### Levantando la infraestructura

Vamos a utilizar los módulos de Terraform para implementar una VPC (Virtual Private Cloud) y un RDS (Relational Database Service).

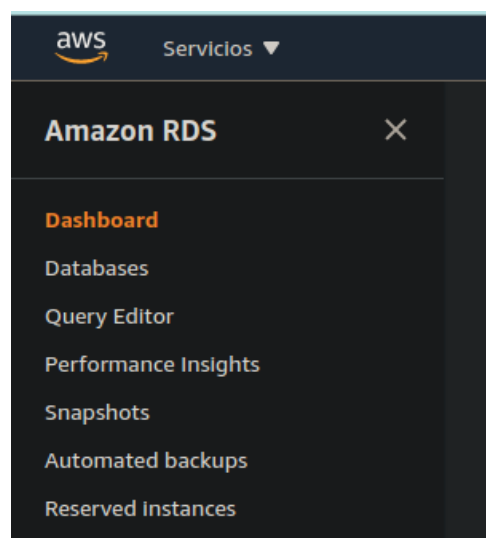
Una vez descargado y descomprimido el ZIP, nos situamos en la carpeta que se generó y luego en Terraform. Allí aplicamos el código con las siguientes instrucciones:

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

Nos pedirá una confirmación, a la que le escribimos "yes". Al terminar veremos el siguiente mensaje:

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

Esto dejará un RDS iniciando en la cuenta de AWS. Lo vemos al ingresar en la consola de AWS y buscar RDS. Luego, en el menú izquierdo, vamos a Databases:



RDS > Databases

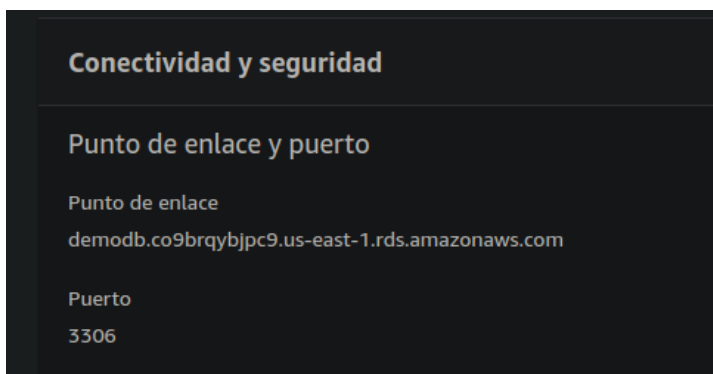
**Databases** Group resources Refresh Modify Actions Restore from S3 Create database

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

Cuando se muestre disponible (Available) ya podremos conectarnos. Necesitamos el DNS. Para conseguirlo, hacemos clic en el nombre de nuestro RDS "demodb". En la sección de "Conectividad y seguridad" copiamos el DNS que nos aparecerá debajo de "Punto de enlace":



Corremos el siguiente comando para ingresar. Cambiamos "DNS" por el valor que obtuvimos anteriormente: `$ mysql -h dns -uuser -pdemouser!`

Para ver las bases de datos existentes, escribimos y ejecutamos `show databases;` y para salir `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#
```

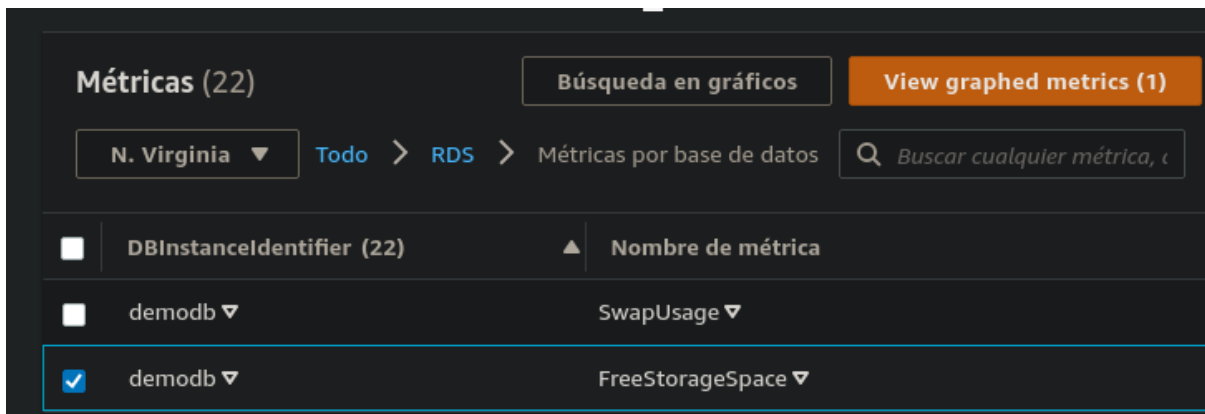
## Creando un dashboard simple

Si bien el RDS incorpora algunos gráficos de métricas de CloudWatch, **¡vamos a crear uno propio!**

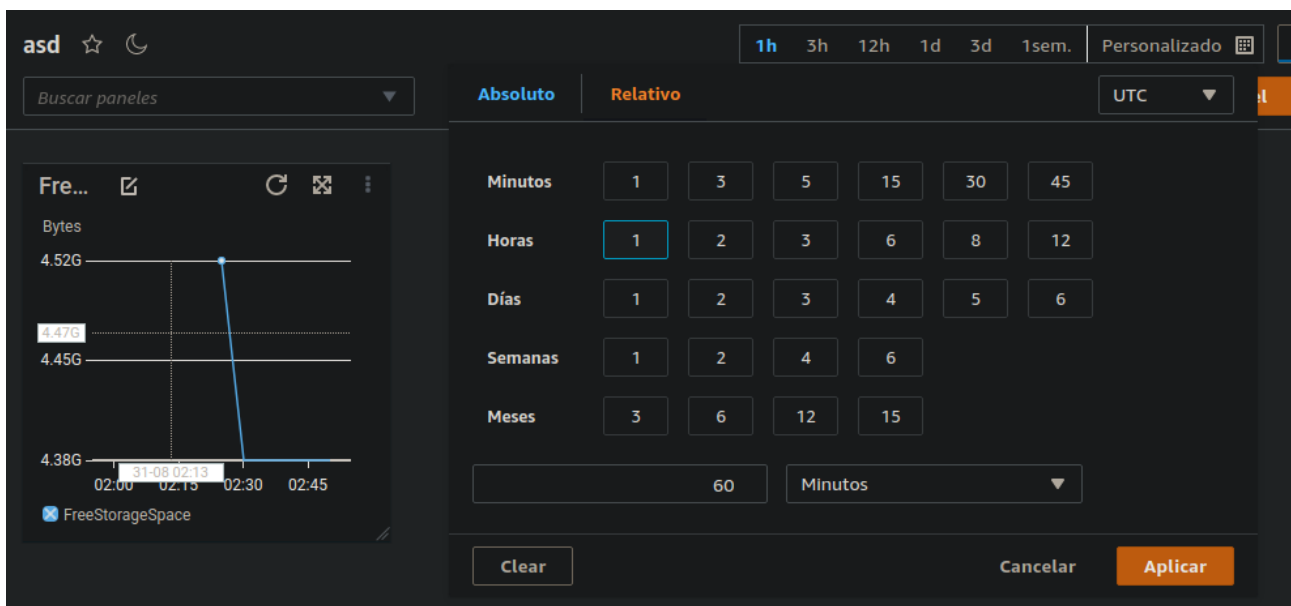
En la consola de AWS nos dirigimos al servicio de CloudWatch. En Dashboards o Paneles hacemos clic en “Crear un panel”. Damos un nombre y continuamos. En “Agregar un widget”, seleccionamos “Línea” y luego “Métricas”:



En la parte inferior del diálogo estarán las métricas. Buscamos RDS y luego “Métricas por base de datos”. En la lista de métricas que vemos, seleccionamos “FreeStorageSpace” con el nombre de nuestra RDS. Tal como vemos en la siguiente imagen:



Para terminar la creación del widget, clic en el botón “Crear un widget”. Haciendo clic en “Personalizado” -barra superior derecha- podemos agrandar nuestro gráfico arrastrándolo desde una esquina y cambiar el rango de tiempo a 30 minutos.



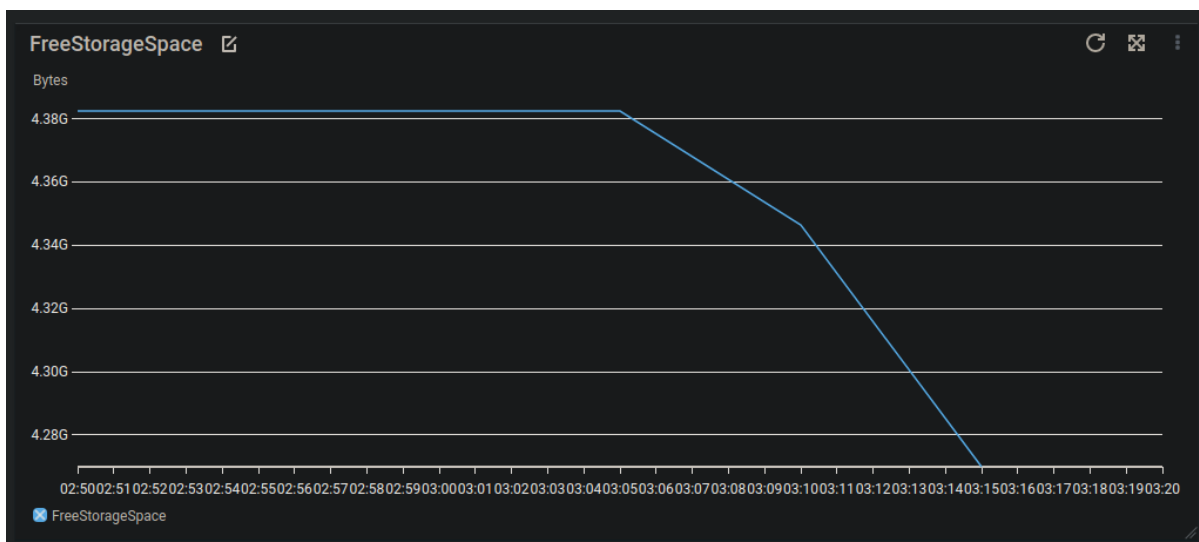
## Corriendo una prueba de carga

Vamos a pasar un script que ocupe espacio en la base de datos para poder monitorear cómo corre nuestro dashboard.

Situados en la carpeta que se generó al descomprimir el ZIP, editamos el archivo “.env” agregando la dirección DNS de nuestro RDS:

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

Corremos el script de carga ejecutando: "bash runTest.sh" y vamos a nuestro dashboard a ver el comportamiento. Luego de un tiempo podremos observar en nuestro gráfico el siguiente resultado:



Notaremos que el espacio disponible fue disminuyendo en el transcurso que se corrió el script.

## Limpiemos el ambiente

Una vez finalizada la práctica vamos a borrar nuestro dashboard haciendo clic en “Acciones” y luego en “Eliminar”. Eliminaremos el RDS corriendo el comando `terraform destroy` e ingresando “yes”. AWS nos cobra por uso y siempre que no usemos los recursos debemos borrarlos.

## Conclusión

En esta práctica realizamos una prueba de carga a una base de datos AWS RDS y monitoreamos el estado del disco con AWS CloudWatch. Si bien no llegamos a dejar el disco del RDS sin espacio, es fundamental monitorear siempre los discos ya que si nos quedamos sin espacio el sistema host y nuestra base de datos o servicio dejarán de funcionar.