



Certified Tech Developer

The Ultimate Degree

Infraestructura II

Actividad obligatoria e individual

Dificultad: media

PARTE 2: agregamos autoescalado

Esta práctica es la continuación de la parte 1 “Monitoreamos un RDS”. Lo que haremos es agregar a nuestra infraestructura autoescalado de almacenamiento. Al llenarse el disco de nuestro RDS crecerá automáticamente para no quedarnos sin espacio.

¿Qué vamos a necesitar?

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 2 ([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), un RDS (Relational Database Service) y un EC2(Elastic Cloud Computer).

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:

```
$ chmod 400 sshkey
$ 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 after 10m20s [id=demodb]
Apply complete! Resources: 18 added, 0 changed, 0 destroyed.
```

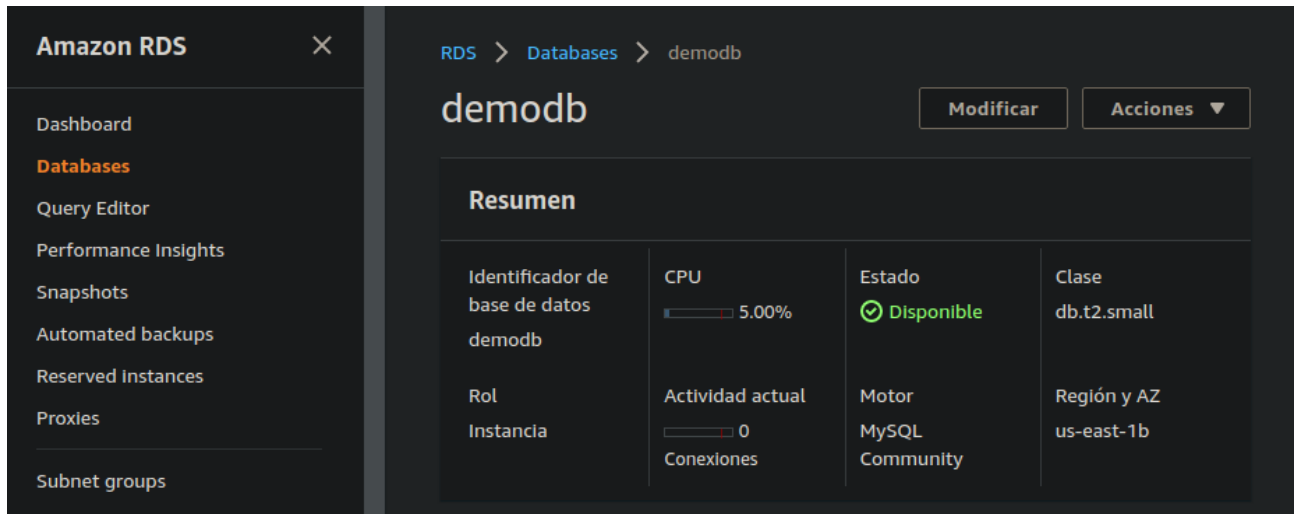
Outputs:

```
jumphost_ip = "10.0.101.30"
rds_dns = "demodb.co9brqybjpg9.us-east-1.rds.amazonaws.com:3306"
```

Guardemos los datos que aparecen en el output. ¡Los vamos a necesitar!

Configuramos escalado automático de almacenamiento

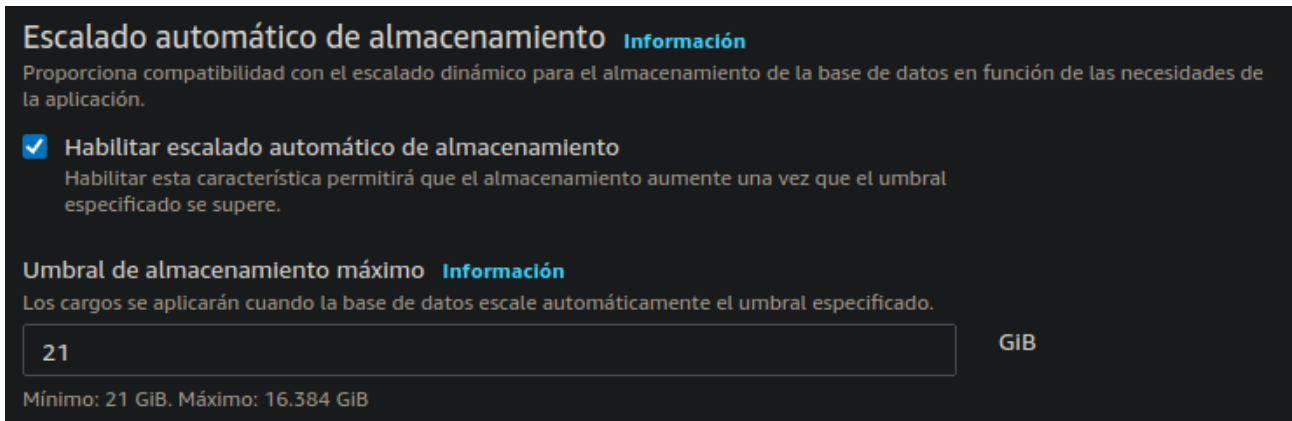
Nos dirigimos al servicio de RDS y damos clic en “Modificar”.



The screenshot shows the Amazon RDS console interface. On the left is a navigation menu with options like Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Automated backups, Reserved Instances, Proxies, and Subnet groups. The main area displays the details for a database instance named 'demodb'. At the top right of the main area are buttons for 'Modificar' and 'Acciones'. Below this is a 'Resumen' (Summary) section with a table of instance details:

Identificador de base de datos demodb	CPU 5.00%	Estado ✓ Disponible	Clase db.t2.small
Rol Instancia	Actividad actual 0 Conexiones	Motor MySQL Community	Región y AZ us-east-1b

En el formulario, buscamos la sección “Escalado automático de almacenamiento”. Lo habilitamos seleccionando la casilla y seteamos el “Umbral de almacenamiento máximo” en 21:



The screenshot shows the 'Escalado automático de almacenamiento' (Automatic Storage Scaling) configuration page. It includes a description: 'Proporciona compatibilidad con el escalado dinámico para el almacenamiento de la base de datos en función de las necesidades de la aplicación.' Below this is a checkbox labeled 'Habilitar escalado automático de almacenamiento' which is checked. A note states: 'Habilitar esta característica permitirá que el almacenamiento aumente una vez que el umbral especificado se supere.' Further down, there is a section for 'Umbral de almacenamiento máximo' (Maximum storage threshold) with a value of '21' entered in a text box. The unit 'GiB' is shown to the right. At the bottom, it indicates the range: 'Mínimo: 21 GiB. Máximo: 16.384 GiB'.

Con esto definimos que, cuando se necesite más espacio de almacenamiento en la instancia de la base de datos, el tamaño escalará o se incrementará satisfaciendo dicha necesidad. Clic en “Continuar”. Antes de guardar estos cambios dejemos marcada la opción “Aplicar Inmediatamente” en la sección “Programación de modificaciones”, para que se apliquen inmediatamente.

Tengamos en cuenta que las bases de datos son recursos que en producción no podemos apagar en cualquier momento y muchas de las modificaciones requieren un

reinicio. Por eso, las modificaciones suelen programarse para ser realizadas en una ventana de mantenimiento.

Programación de modificaciones

Cuándo aplicar las modificaciones

- ☐ **Aplicar durante el próximo periodo de mantenimiento programado**
Periodo de mantenimiento actual: September 02, 2021 02:59 - 03:29 UTC-3
- ☒ **Aplicar inmediatamente**
Los cambios en esta solicitud y cualquier modificación pendiente se aplicarán de forma asíncrona lo antes posible, independientemente de la configuración del periodo de mantenimiento para esta instancia de base de datos.

Para terminar la modificación damos en el botón "Modificar la instancia de base de datos".

¡Veamos el auto escalado en acción!

Para ver en acción el escalado necesitamos dos cosas:

1. Monitorear el espacio libre en el almacenamiento de la instancia de nuestro RDS.
Para ello podemos ir a la sección de métricas de CloudWatch dentro del mismo RDS o a algún panel o dashboard que hayamos creado en el servicio de CloudWatch.
2. Generar una carga de datos al RDS.

Generemos la carga

Utilizaremos la red interna de AWS para generar la carga. Nos conectaremos a la máquina virtual que se levantó con Terraform y desde ahí enviaremos datos al RDS.

Para generar la carga corremos un script que se encuentra en el ZIP de esta ejercitación. ¡Tenemos que subirlo a nuestra instancia ec2! Para hacerlo, corremos el comando situado en la carpeta de Terraform:

```
$ scp -i ./sshkey ../part2.zip ec2-user@jumphost_ip:/home/ec2-user/
```

Tengamos en cuenta que las rutas deben estar correctas: **part2.zip** es el archivo con los recursos de esta práctica y **sshkey** es la clave para acceder a la instancia que estará en la carpeta generada al descomprimir el ZIP. ¡No olvidemos cambiar **jumphost_ip** por el IP que nos devolvió Terraform al correr el apply.

Una vez copiado el ZIP ingresamos al jumphost de la siguiente forma:

```
$ ssh -i ./sshkey ec2-user@jumphost_ip
```

Una vez dentro descomprimos el ZIP que subimos:

```
$ unzip parte2.zip
```

Entramos en la carpeta y editamos el archivo .env:

```
$ cd /parte2  
$ nano part2
```

Vamos a modificar el archivo ".env", agregaremos el DNS sin el puerto y modificamos los parámetros de la siguiente manera:

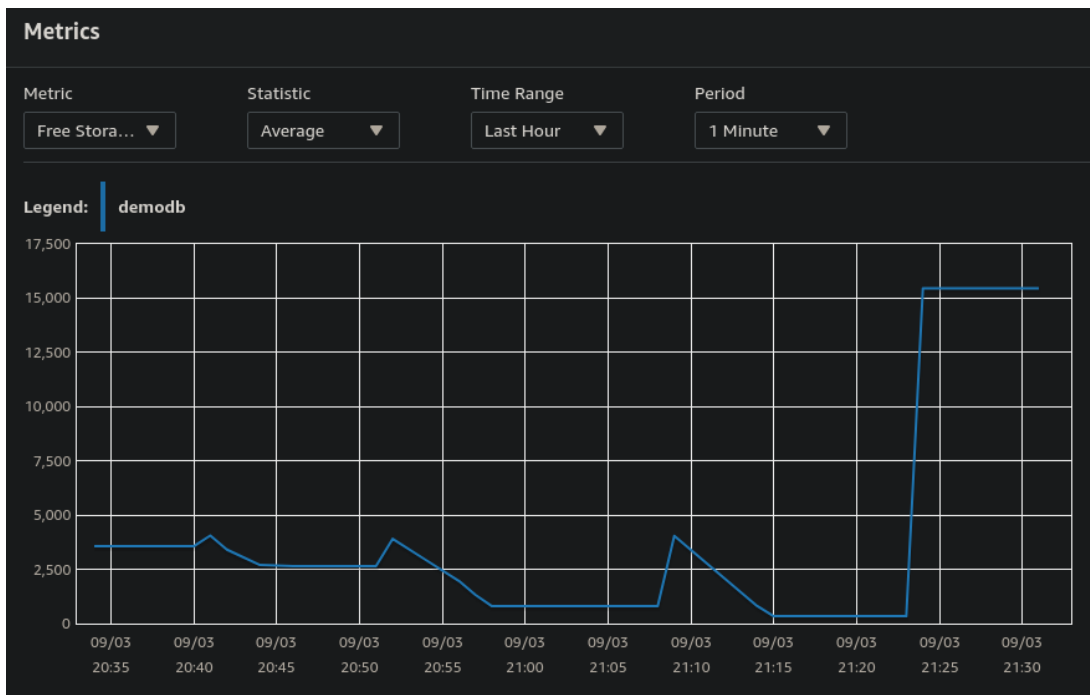
```
TEST_NAME="loadDiskTest"  
DNS_DB="demodb.co9brqybjpgc9.us-east-1.rds.amazonaws.com"  
USER="user"  
PASSWORD="demouser!"  
NUM_DBS=5  
NUM_TABLES=25  
NUM_USERS=20
```

Luego corremos el script de carga. Este proceso demora un tiempo.

```
$ ./runTest.sh
```

Observaremos en nuestro monitoreo que el espacio se completa hasta dejar solo unos 300 a 500 megas libres. Esto es menos del 10%, valor al que debemos llegar para que se active el autoescalado de storage que configuramos (este porcentaje ya es así y no lo

podemos modificar). Luego de un tiempo de espera notaremos que de forma automática la base de datos pasará a modificarse y veremos lo siguiente en nuestro monitoreo:



¡El escalado automático de almacenamiento surtió efecto!

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 volvimos a monitorear un RDS, pero esta vez configuramos un autoescalado que pudimos ver funcionando en las métricas de CloudWatch.

Si bien no es muy común poner un autoescalado en el storage de las bases de datos -ya que se calcula según el uso que se le da a la misma-, en algunos casos puede ser de ayuda. Pensemos un caso posible. Tenemos una plataforma con poco tráfico y de pronto

hay más interacciones. Los datos que guardemos pueden incrementar notablemente (usuarios, post, etc.). ¡Mucho más de lo que hubiésemos imaginado! En este caso, esta característica de RDS nos es de gran utilidad.