



Universidad Politecnica de Chiapas

Ingeniería en software 5-B

Cliente servidor
Crond job practica 2

Alumno:
Carlos Mario Ruiz Pinacho

Matricula:
221220

Como ya tenemos una instancia de ec2 donde realizamos nuestra primera practica lo que unico que necesitaremos para esta practica es instalar lo siguiente para realizar la practica:

-Python3: nos permitira crear scripts para realizar las tareas que se menciona en la practica

Comando: sudo apt install python3

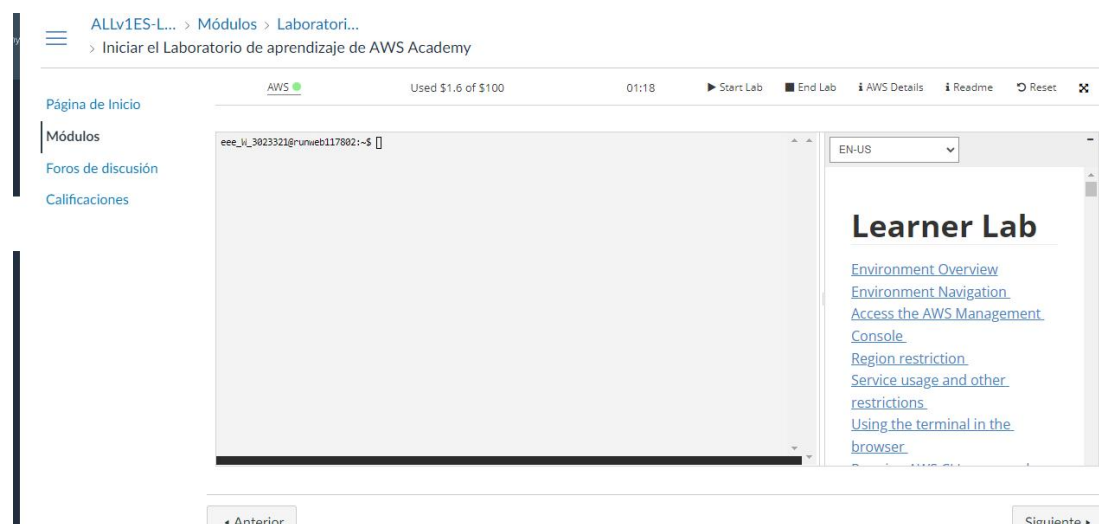
Al tener instalado python procederemos a instalar una libreria adicional para poder interactuar con aws y asi interactuar con los bucket .

Librerias requeridas:

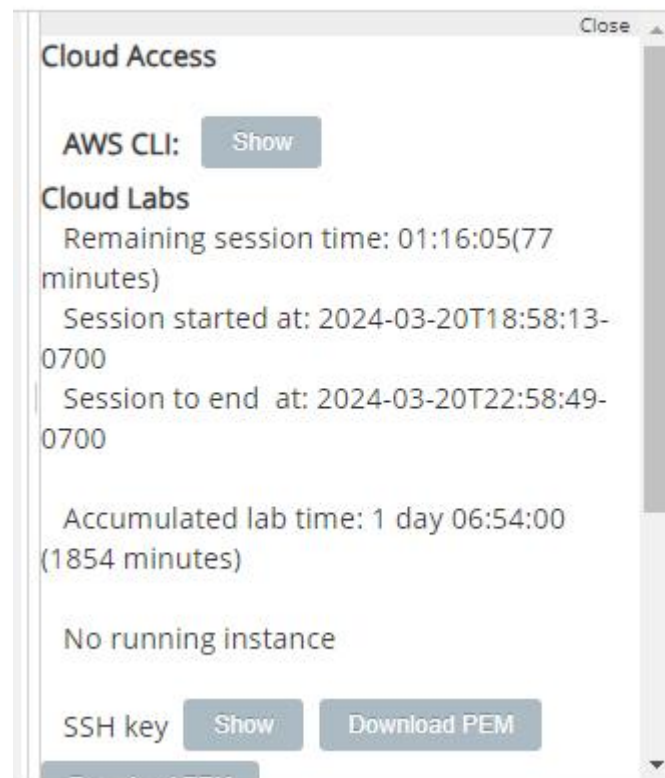
-boto3

Configuración de las credenciales:

Una parte importante para realizar la practica es tener las credenciales para trabajar con los buckets de s3 si estamos trabajando con laboratorio para estudiantes de aws podemos consultar aca en **aws details**.



Una vez dado click se mostrara toda esta opciones:



Pero lo que nos interesa es aws cli para obtener todas las credenciales para ingresarlo en nuestro script de python

AWS CLI:

Copy and paste the following into
./aws/credentials

```
[default]
aws_access_key_id=ASIA6GBMC4PVGCGLG0Q
L
aws_secret_access_key=D/A5z9h2PUgkw/C
MX7mKWzZVqvkdff+couMKMYM+
aws_session_token=FwoGZXIvYXdzEBsaDKl
7b28NtrxxYNCA6yLKAZpQIJuDweUBUzqkA5Yn
vdYtuf8l0gO8uj7MsdM8b0v+Q2xPggBg34/JH
GvSRwlinU9gwqd30hpjLSp8MN6rpe370zOpZF
olCSGYyrJESNzShGJlL/3XyD8+wSELXi+1z8U
0QwldO6qPna8b7w9xKJlu+uQHLUuSEjIZ22kO
Mc651bLnzwM6Q7mTwfzAwQpPsCwZq/h1pk8hn
```

Una vez instalado lo requerido para la practica ejecutaremos el comando **crontab -e** en la terminal de ubuntu en nuestra instancia de aws para editar el archivo de las tareas que se realizaran

```
ubuntu@ip-172-31-36-190:~$ crontab -e
```

Una vez dentro del archivo de los cronb job procedemos agregar los cron que queremos que se ejecuten:

```
#test de prueba
# Respaldo a las 11:59 am y pm
59 11 * * * python3 script.py
59 23 * * * python3 script.py
#para la subida y eliminacion
0 1 * * 1,3,5 python3 deleteup.py
```

Las dos primeras lineas servirán para crear respaldo a nuestra base de datos alojada en rds a travez de un script de python.

La tercera linea es para subir los archivos a s3 asi mismo eliminar de nuestro sistema operativo.

Scripts de python

```
ubuntu@ip-172-31-36-190:~$ ls
deleteup.py  script.py
ubuntu@ip-172-31-36-190:~$
```

Nuestro scripts de python estaran alojada en home/ubuntu podemos colocar donde queramos esos archivos pero debemos recordar modificar el cron correspondiente para modificar la ubicacion donde estara ubicado la ruta de nuestros scripts.

Script.py

Este es script sera el encargado de generar los respaldo del archivo sql de nuestra base de datos alojados en rds

Codigo del script

```
import subprocess
from datetime import datetime

# Configuración de la conexión a la base de datos
host = "bdaws.cx2k8kkm6yok.us-east-1.rds.amazonaws.com"
usuario = "admin"
contraseña = "12345678"
nombre_base_de_datos = "awsCarlos"

# Generar un nombre de archivo de respaldo basado en la fecha y hora actual
fecha_hora_actual = datetime.now().strftime('%Y%m%d_%H%M%S')
nombre_archivo_respaldo = f"backup_{nombre_base_de_datos}_{fecha_hora_actual}.sql"

# Comando para crear un respaldo completo de la base de datos
comando = f"mysqldump -h {host} -u {usuario} -p{contraseña} {nombre_base_de_datos} > {nombre_archivo_respaldo}"

# Ejecutar el comando en la línea de comandos
try:
    subprocess.run(comando, shell=True, check=True)
    print("Respaldo completado.")
except subprocess.CalledProcessError as e:
    print("Error al crear el respaldo:", e)
```

Deleteup.py

Este script sera el encargado de eliminar y subir los respaldos generados a s3

Codigo

```
import subprocess
from datetime import datetime

# Configuración de la conexión a la base de datos
host = "bdaws.cx2k8kkm6yok.us-east-1.rds.amazonaws.com"
usuario = "admin"
contraseña = "12345678"
nombre_base_de_datos = "awsCarlos"

# Generar un nombre de archivo de respaldo basado en la fecha y hora actual
fecha_hora_actual = datetime.now().strftime('%Y%m%d_%H%M%S')
nombre_archivo_respaldo = f"backup_{nombre_base_de_datos}_{fecha_hora_actual}.sql"

# Comando para crear un respaldo completo de la base de datos
comando = f"mysqldump -h {host} -u {usuario} -p{contraseña} {nombre_base_de_datos} > {nombre_archivo_respaldo}"

# Ejecutar el comando en la línea de comandos
try:
    subprocess.run(comando, shell=True, check=True)
    print("Respaldo completado.")
except subprocess.CalledProcessError as e:
    print("Error al crear el respaldo:", e)

ubuntu@ip-172-31-36-190:~$ ls
deleteup.py  script.py
ubuntu@ip-172-31-36-190:~$ cat deleteup.py
import os
import boto3

# Establecer las credenciales de AWS
s3 = boto3.client(
    's3',
    aws_access_key_id='ASIA6GBMC4PVGCGLG0QL',
    aws_secret_access_key='D/A5z9h2PUgkw/CMX7mKWzZVqvkdff+couMKMYM+',

    aws_session_token='FwoGZXIvYXZlEBsaDK17b28NtrxxYNCA6yLKAZpQIJuDweUBUzqkA5YnvdYtuf810g08uj7MsdM8b0v+Q2xPg
gBg34/JHGvSRwlinU9gwqd30hpjLSp8MN6rpe370z0pZFolCSGYrJESNzShGJLL/3XyD8+wSELXi+1z8U0Qwld06qPna8b7w9xKJlu+u
QHLUuSEjIZ22k0Mc651bLnzm6Q7mTfzAWQpPsCWZq/hlpk8hnfCEDA0y0rc1p4s7wfxvt18d8t5xQC3+LJU4C1HQnYaeFaG5cBA4b2c
Jm/NPp1KnMRYotqjurYyLdA2nINv17nx7pvaFGdoHU5U33M2040+seJ1VIpYBmcy7KXERfiT4SLdioVV6w=='
)

# Nombre del bucket en S3
bucket_name = 'awscarloscserver'

# Directorio donde se encuentran los archivos de respaldo (misma carpeta que el script)
directorio_respaldo = os.path.dirname(os.path.abspath(__file__))
# Obtener la lista de archivos de respaldo con extensión .sql
archivos_respaldo = [archivo for archivo in os.listdir(directorio_respaldo) if
archivo.startswith('backup') and archivo.endswith('.sql')]
print(archivos_respaldo)
# Subir cada archivo de respaldo a S3 y luego eliminarlo
for archivo in archivos_respaldo:
    ruta_completa = os.path.join(directorio_respaldo, archivo)
    s3.upload_file(ruta_completa, bucket_name, archivo)
    os.remove(ruta_completa)

print("Archivos de respaldo SQL subidos a S3 y eliminados localmente.")
```

Epilogo:

Como para ver los resultados sera algo tardado veremos la funcionalidad de los script realizados.

Test script.py:

```
ubuntu@ip-172-31-36-190:~$ ls
deleteup.py  script.py
ubuntu@ip-172-31-36-190:~$ python3 script.py
mysqldump: [Warning] Using a password on the command line interface can be insecure.
Warning: A partial dump from a server that has GTIDs will by default include the GTIDs of all transactions, even those that changed suppressed parts of the database. If you don't want to restore GTIDs, pass --set-gtid-purged=OFF. To make a complete dump, pass --all-databases --triggers --routines --events.
Warning: A dump from a server that has GTIDs enabled will by default include the GTIDs of all transactions, even those that were executed during its extraction and might not be represented in the dumped data. This might result in an inconsistent data dump.
In order to ensure a consistent backup of the database, pass --single-transaction or --lock-all-tables or --master-data.
Respaldo completado.
ubuntu@ip-172-31-36-190:~$ ls
backup_awsCarlos_20240321_044745.sql  deleteup.py  script.py
ubuntu@ip-172-31-36-190:~$
```

Como se puede observar se realizo correctamente el respaldo del archivo sql

Test deleteup.py:

Ahora pondremos a prueba el script que nos permitira subir a s3 y eliminar de nuestro entorno

```
ubuntu@ip-172-31-36-190:~$ python3 deleteup.py
['backup_awsCarlos_20240321_044745.sql']
Archivos de respaldo SQL subidos a S3 y eliminados localmente.
ubuntu@ip-172-31-36-190:~$ ls
deleteup.py  script.py
ubuntu@ip-172-31-36-190:~$
```

Como podemos ver el resultado fue exitoso elimino el archivo y lo subio a nuestro bucket de s3

Resultado en s3

Objetos (1) Información

Copiar URI de S3

Copiar URL

Descargar

Abrir

Eliminar

Acciones ▼

Crear carpeta

Cargar

Los objetos son las entidades fundamentales que se almacenan en Amazon S3. Puede utilizar el [Inventario de Amazon S3](#) para obtener una lista de todos los objetos de su bucket. Para que otras personas obtengan acceso a sus objetos, tendrá que concederles permisos de forma explícita. [Más información](#)

🔍 Buscar objetos por prefijo

< 1 > ⚙️

<input type="checkbox"/>	Nombre ▲	Tipo ▼	Última modificación ▼	Tamaño ▼	Clase de almacenamiento ▼
<input type="checkbox"/>	backup_awsCarlos_20240321_044745.sql	sql	20 Mar 2024 10:50:28 PM CST	3.7 KB	Estándar