

2º ASIR

Práctica 6

Consola CLI

Administración de sistemas operativos

aws



Sergio Garcia Márquez
I.E.S SAN SEBASTIÁN

Índice

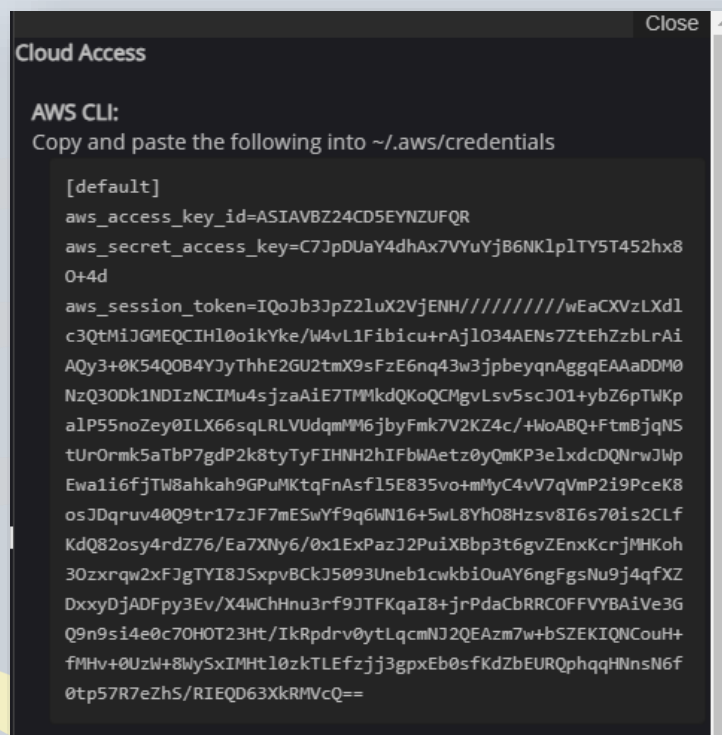
1. Acceso a la consola desde entorno gráfico de AWS	2
2. Instalación de AWS CLI en el SO y configurar AWS CLI en nuestro ordenador....	3
3. Listar todas las VPC	4
4. Crear grupo de seguridad desde la consola de AWS	4
5. Crear 1 instancia EC2 en el grupo de seguridad	5
6. Crear instancias con scripts de Linux.....	7
7. Modifica el anterior script para que lanzar 2 instancias y las actualice al lanzar	9



1. Acceso a la consola desde entorno gráfico de AWS

Esta práctica vamos a trabajar el modo consola de AWS. Usaremos la Shell de aws para diferentes operaciones. Comenzamos por conectarnos a la cli de aws desde el logo de la Shell.

En el inicio del laboratorio, tendremos como referencia estos datos para usarlos a la hora de configurar las credenciales.

A screenshot of a terminal window titled "Cloud Access" with a "Close" button in the top right corner. The terminal displays instructions for setting up the AWS CLI. It says "AWS CLI: Copy and paste the following into ~/.aws/credentials". Below this, it shows a sample configuration for the "default" profile, including the access key ID, secret access key, and session token, all represented by long alphanumeric strings.

```
Cloud Access
AWS CLI:
Copy and paste the following into ~/.aws/credentials

[default]
aws_access_key_id=ASIAVBZ24CD5EYNZUFQR
aws_secret_access_key=C7JpDUaY4dhAx7VYuYjB6NK1p1TY5T452hx8
0+4d
aws_session_token=IQoJb3JpZ2luX2VjENH////////wEaCXVzLXd1
c3QtMiJGMEQCIH10oikYke/W4vL1Fibicu+rAj1034AENs7ZtEhZzbLrAi
AQy3+0K54QOB4YJyThhE2GU2tmX9sFzE6nq43w3jpbeyqnAggqEAAaDDM0
NzQ3ODk1NDIzNCIMu4sJzaAiE7TMMkdQKoQCMgvLsv5scJ01+ybZ6pTWKp
a1P55noZey0ILX66sqLRLVUdqmM46jbyFmk7V2KZ4c/+WoABQ+FtmBjqNS
tUrOrmk5aTbP7gdP2k8tyTyFIHh2hIFbWAetz0yQmKP3elxdcdQNrWJwp
Ewa1i6fjTW8ahkah9GPuMKtqFnAsf15E835vo+mMyC4vV7qVmp2i9PceK8
osJDqruv40Q9tr17zJF7mESwYf9q6wN16+5wL8Yh08Hsv8I6s70is2CLf
KdQ82osy4rdZ76/Ea7XNy6/0x1ExPazJ2PuiXBbp3t6gvZENxKcrjMHKoh
30zxrqw2xFJgTYI8JSxpvBCkJS093Uneb1cwkb10uAY6ngFgsNu9j4qfXZ
DxxyDjADfpy3Ev/X4wChHnu3rf9JTFKqaI8+jrPdaCbRRCOFFVYBAiVe3G
Q9n9si4e0c70HOT23Ht/IkRpdv0ytLqcmNJ2QEAzm7w+bSZEKIQCouH+
fMHv+0UzW+8WySxIMHt10zkTLEfzjj3gpxEb0sfKdZbEURQphqHnNsN6f
0tp57R7eZhS/RIEQD63XkRMVcQ==
```

Escribiremos el comando `aws configure`. Este comando crea el archivo de configuración. Para configurar, el sistema te pedirá una serie de valores que se almacenarán en archivos de configuración dentro de tu máquina local. Estos valores incluyen:

- **AWS Access Key ID** - Este es el ID de clave de acceso asociado a tu cuenta de AWS, que se utiliza para autenticar solicitudes desde la CLI.
- **AWS Secret Access Key** - Es la clave secreta que se utiliza junto con el Access Key ID para firmar tus solicitudes de AWS.
- **Default region name** - Define la región por defecto en la que se realizarán las solicitudes si no especificas una región en los comandos. En el laboratorio usaremos `us-east-1`.

- **Default output format** - formato de salida de los comandos de AWS CLI.
Pondremos json.

```
[cloudshell-user@ip-10-132-35-75 ~]$ aws configure
AWS Access Key ID [*****X3Z5]: ASIABV224CD5EYNZUFQR
AWS Secret Access Key [*****kv6I]: C7jP0UaY4dhAx7VYuYjB6NK1p1TY5T452hx80+4d
Default region name [us-east-1]: us-east-1
Default output format [json]: json
[cloudshell-user@ip-10-132-35-75 ~]$
```

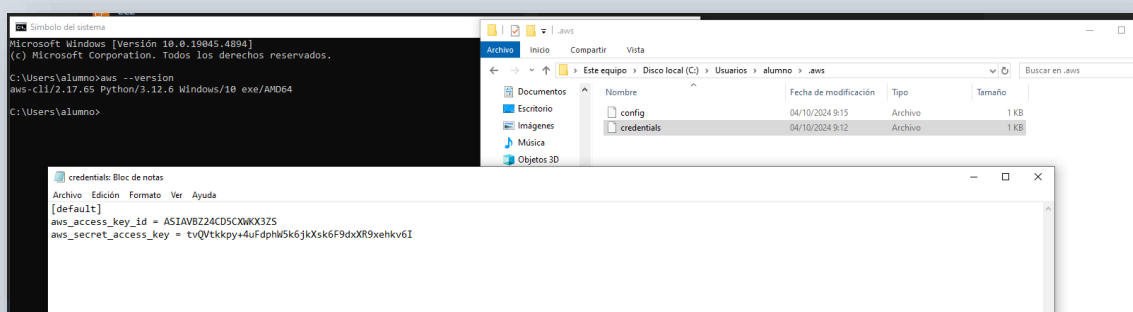
Luego se habrá creado el archivo `.aws/credentials`. Haremos un `sudo nano` e insertaremos en el archivo el `aws_session_token` de la primera imagen.

[illegible]

2. Instalación de AWS CLI en el SO y configurar AWS CLI en nuestro ordenador

Descargamos el cliente AWS cli y ahí podremos acceder, quedando instalado en cmd y powershell.

Si no funcionase, volveremos a poner las credenciales, es lo mismo que hemos hecho en el primer apartado.



3. Listar todas las VPC

Buscaremos las VPC que tenemos con el comando:

✓ `aws ec2 describe-vpcs --query "Vpcs[*].{ID:VpcId,CIDR:CidrBlock}"`

```
[cloudshell-user@ip-10-134-45-7 ~]$ aws ec2 describe-vpcs --query "Vpcs[*].{ID:VpcId,CIDR:CidrBlock}"
[
  {
    "ID": "vpc-0d6bbc3b6947f78f6",
    "CIDR": "172.31.0.0/16"
  }
]
```

4. Crear grupo de seguridad desde la consola de AWS

Se creó un grupo de seguridad para las máquinas del Backend con el nombre practica6, copiando la VPC del apartado anterior.

```
[cloudshell-user@ip-10-134-45-7 ~]$ aws ec2 describe-vpcs --query "Vpcs[*].{ID:VpcId,CIDR:CidrBlock}"
[
  {
    "ID": "vpc-0d6bbc3b6947f78f6",
    "CIDR": "172.31.0.0/16"
  }
]
[cloudshell-user@ip-10-134-45-7 ~]$
[cloudshell-user@ip-10-134-45-7 ~]$ aws ec2 create-security-group --group-name practica6 --description "Reglas backend" --vpc-id vpc-0d6bbc3b6947f78f6
{
  "GroupId": "sg-02f362db096fe62c8"
}
[cloudshell-user@ip-10-134-45-7 ~]$
```

```
aws ec2 create-security-group --group-name practica6 --description "Reglas backend" --
vpc-id vpc-0d6bbc3b6947f78f6
```

Ahora tenemos que abrir los puertos, copiando el group-id podemos hacerlo de la siguiente forma:

```
[cloudshell-user@ip-10-134-45-7 ~]$ aws ec2 authorize-security-group-ingress --group-id sg-02f362db096fe62c8 --protocol tcp --port 22 --cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-058331725793f4596",
      "GroupId": "sg-02f362db096fe62c8",
      "GroupOwnerId": "347478954234",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
[cloudshell-user@ip-10-134-45-7 ~]$ aws ec2 authorize-security-group-ingress --group-id sg-02f362db096fe62c8 --protocol tcp --port 3306 --cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0464f83774eea8296",
      "GroupId": "sg-02f362db096fe62c8",
      "GroupOwnerId": "347478954234",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 3306,
      "ToPort": 3306,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
```

```
aws ec2 authorize-security-group-ingress --group-id sg-02f362db096fe62c8 --protocol
tcp --port 22 --cidr 0.0.0.0/0
```

```
aws ec2 authorize-security-group-ingress --group-id sg-02f362db096fe62c8 --protocol
tcp --port 3306 --cidr 0.0.0.0/0
```

5. Crear 1 instancia EC2 en el grupo de seguridad

El comando para ver instancias: `aws ec2 describe-instances`

Para solo las instancias running, usa el comando con un filtro: `aws ec2 describeinstances --filters "Name=instance-state-name,Values=running"`

Para crear una instancia con todos los datos anteriores, usaremos el siguiente comando:

```
aws ec2 run-instances --image-id ami-053b0d53c279acc90 --count 1 --instance-type
t2.micro --key-name vockey --security-group-ids sg-02f362db096fe62c8 --subnet-id
subnet-058609a657cca92b4 --tag-specifications
'ResourceType=instance,Tags=[{ Key=Name, Value=Sergio-GM-P6Server}]'
```

```
[cloudshell-user@ip-10-134-45-7 ~]$ aws ec2 run-instances --image-id ami-053b0d53c279acc90 --count 1 --instance-type t2.micro --key-name vockey --security-group-ids sg-02f362d096fe62c8 --subnet-id subnet-058609a657cca92b4 --tag-specifications "ResourceType=Instance,Tags=[{Key=Name,Value=Sergio-GM-P6Server}]"
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-053b0d53c279acc90",
      "InstanceId": "i-0fd93400e5cbae2bc",
      "InstanceType": "t2.micro",
      "KeyName": "vockey",
      "LaunchTime": "2024-10-08T14:51:02+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1a",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-10-173.ec2.internal",
      "PrivateIpAddress": "172.31.10.173",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-058609a657cca92b4",
      "VpcId": "vpc-046b0c3b09477816",
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "31323dde-c70d-413e-aa78-036c44c146df",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hvfnetwork": "en"
    }
  ]
}
```

Y listo, ya tendríamos la instancia creada.

<input checked="" type="checkbox"/>	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de estado	Estado de la al...	Zona de dispon...	DNS de IPv4 pública	Dirección IP...
<input checked="" type="checkbox"/>	Sergio-GM-P6Server	i-0fd93400e5cbae2bc	En ejecución	t2.micro	Inicializando	Ver alarmas +	us-east-1a	ec2-3-223-127-118.co...	3.223.127.118

i-0fd93400e5cbae2bc (Sergio-GM-P6Server)

Detalles

Estado y alarmas

Monitoreo

Seguridad

Redes

Almacenamiento

Etiquetas

▼ Resumen de instancia Información

ID de la instancia

i-0fd93400e5cbae2bc (Sergio-GM-P6Server)

Dirección IPv6

-

Dirección IPv4 pública

3.223.127.118 | dirección abierta

Estado de la instancia

En ejecución

Direcciones IPv4 privadas

172.31.10.173

DNS de IPv4 pública

ec2-3-223-127-118.compute-1.amazonaws.com | dirección abierta

6. Crear instancias con scripts de Linux

Ahora crearemos una instancia mediante un script de Linux. Para ello creamos el archivo `lanzar_instancia.sh` y le damos permisos con `chmod`.

```
chmod: changing permissions of 'lanzar_instancia.sh': Operation not per
[cloudshell-user@ip-10-134-45-7 ~]$ sudo chmod +x lanzar_instancia.sh
[cloudshell-user@ip-10-134-45-7 ~]$
```

Ahora meteremos el siguiente script dentro del archivo:

```
#!/bin/bash

# Parámetros para la instancia
IMAGE_ID="ami-053b0d53c279acc90"
INSTANCE_TYPE="t2.micro"
KEY_NAME="vockey"
SECURITY_GROUP_ID="sg-02f362db096fe62c8"
SUBNET_ID="subnet-058609a657cca92b4"
TAG_SPECIFICATIONS='ResourceType=instance,Tags=[{ Key=Name, Value=SergioGarcia-P6ServerII}]'

# Ejecutar el comando para lanzar la instancia
echo "Lanzando la instancia EC2..."

INSTANCE_ID=$(aws ec2 run-instances \
  --image-id $IMAGE_ID \
  --count 1 \
  --instance-type $INSTANCE_TYPE \
  --key-name $KEY_NAME \
  --security-group-ids $SECURITY_GROUP_ID \
  --subnet-id $SUBNET_ID \
  --tag-specifications "$TAG_SPECIFICATIONS" \
  --query 'Instances[0].InstanceId' \
  --output text)

if [ -z "$INSTANCE_ID" ]; then
  echo "Error al lanzar la instancia."; exit 1
else
  echo "Instancia lanzada con éxito. ID de la instancia: $INSTANCE_ID"
fi
```



```

PrivateEnvironmentOptions : {
  GNU nano 5.8
  #!/bin/bash
  # Parámetros para la instancia
  IMAGE_ID="ami-053b0d53c279acc90"
  INSTANCE_TYPE="t2.micro"
  KEY_NAME="vockey"
  SECURITY_GROUP_ID="sg-02f362db096fe62c8"
  SUBNET_ID="subnet-050609a657cca92b4"
  TAG_SPECIFICATIONS='ResourceType=instance,Tags=[{Key=Name,Value=SergioGarcia-P6ServerII}]'

  # Ejecutar el comando para lanzar la instancia

  echo "Lanzando la instancia EC2..."
  INSTANCE_ID=$(aws ec2 run-instances \
    --image-id $IMAGE_ID \
    --count 1 \
    --instance-type $INSTANCE_TYPE \
    --key-name $KEY_NAME \
    --security-group-ids $SECURITY_GROUP_ID \
    --subnet-id $SUBNET_ID \
    --tag-specifications "$TAG_SPECIFICATIONS" \
    --query 'Instances[0].InstanceId' \
    --output text)

  # Comprobar si la instancia fue lanzada exitosamente
  if [ -z "$INSTANCE_ID" ]; then echo "Error al lanzar la instancia."; exit 1
  else echo "Instancia lanzada con éxito. ID de la instancia:
$INSTANCE_ID"
fi
  lanzar_instancia.sh

```

Lo ejecutaremos y se nos creará la instancia.

```

[cloudshell-user@ip-10-134-45-7 ~]$ ./lanzar_instancia.sh
Lanzando la instancia EC2...
Instancia lanzada con éxito. ID de la instancia:

i-0f2f0d0366a6819fa
[cloudshell-user@ip-10-134-45-7 ~]$

```

<input type="checkbox"/>	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de estado	Estado de la al...	Zona de dispen...	DNS de IPv4 pública	Dirección IP...
<input type="checkbox"/>	Sergio-GM-P6Server	i-0fd93400e5cbae2bc	En ejecución	t2.micro	2/2 comprobaciones superadas.	Ver alarmas +	us-east-1a	ec2-3-223-127-118.co...	3.223.127.118
<input type="checkbox"/>	SergioGarcia-P6ServerII	i-0f2f0d0366a6819fa	En ejecución	t2.micro	2/2 comprobaciones superadas.	Ver alarmas +	us-east-1a	ec2-3-235-18-146.com...	3.235.18.146

7. Modifica el anterior script para que lanzar 2 instancias y las actualice al lanzar

Para este script se realizó para que pidiese el nombre de ambas instancias por consola y meterlos en el script, así como actualizar dichas instancias.

Los comandos de actualizar la instancia se metieron en el USER_DATA_FILE temporalmente porque me daba conflicto a la hora de lanzar el script, al final del script se borran.

```
GNU nano 5.8
#!/bin/bash

# Parámetros para la instancia
IMAGE_ID="ami-053b0d53c279acc90"
INSTANCE_TYPE="t2.micro"
KEY_NAME="vockey"
SECURITY_GROUP_ID="sg-02f362db096fe62c8"
SUBNET_ID="subnet-058609a657cca92b4"

# Solicitar el nombre de la instancia
read -p "Introduce el nombre para la primera instancia: " INSTANCE_NAME_1
read -p "Introduce el nombre para la segunda instancia: " INSTANCE_NAME_2

# Definir la etiqueta a partir de lo ingresado por el usuario
TAG_SPECIFICATIONS_1="ResourceType=instance,Tags=[{Key=Name,Value=$INSTANCE_NAME_1}]"
TAG_SPECIFICATIONS_2="ResourceType=instance,Tags=[{Key=Name,Value=$INSTANCE_NAME_2}]"

# Definir el user-data en un archivo temporal
USER_DATA_FILE=$(mktemp)
echo "#!/bin/bash" > "$USER_DATA_FILE"
echo "sudo apt update" >> "$USER_DATA_FILE"
echo "sudo apt upgrade -y" >> "$USER_DATA_FILE"

# Asegurarse de que el archivo sea legible
chmod +x "$USER_DATA_FILE"

# Ejecutar el comando para lanzar la primera instancia
echo "Lanzando la primera instancia EC2..."
INSTANCE_ID_1=$(aws ec2 run-instances \
  --image-id $IMAGE_ID \
  --instance-type $INSTANCE_TYPE \
  --key-name $KEY_NAME \
  --security-group-ids $SECURITY_GROUP_ID \
  --subnet-id $SUBNET_ID \
  --tag-specifications "$TAG_SPECIFICATIONS_1" \
  --user-data file://"$USER_DATA_FILE" \
  --query 'Instances[0].InstanceId' \
  --output text)

# Comprobar si la primera instancia fue lanzada exitosamente
if [ -z "$INSTANCE_ID_1" ]; then
  echo "Error al lanzar la primera instancia.";
  exit 1
else
  echo "Primera instancia lanzada con éxito. ID de la instancia: $INSTANCE_ID_1"
fi
```

```
# Ejecutar el comando para lanzar la segunda instancia
echo "Lanzando la segunda instancia EC2..."
INSTANCE_ID_2=$(aws ec2 run-instances \
  --image-id $IMAGE_ID \
  --instance-type $INSTANCE_TYPE \
  --key-name $KEY_NAME \
  --security-group-ids $SECURITY_GROUP_ID \
  --subnet-id $SUBNET_ID \
  --tag-specifications "$TAG_SPECIFICATIONS_2" \
  --user-data file://"${USER_DATA_FILE}" \
  --query 'Instances[0].InstanceId' \
  --output text)

# Comprobar si la segunda instancia fue lanzada exitosamente
if [ -z "$INSTANCE_ID_2" ]; then
  echo "Error al lanzar la segunda instancia.";
  exit 1
else
  echo "Segunda instancia lanzada con éxito. ID de la instancia: $INSTANCE_ID_2"
fi

# Limpiar el archivo temporal de user-data
rm -f "$USER_DATA_FILE"
```

Sólo queda probar el script y listo.

```
[cloudshell-user@ip-10-134-45-7 ~]$ ./lanzar_instancia2.sh
Introduce el nombre para la primera instancia: script1
Introduce el nombre para la segunda instancia: script2
Lanzando la primera instancia EC2...
Primera instancia lanzada con éxito. ID de la instancia: i-05d70435e03c72aef
Lanzando la segunda instancia EC2...
Segunda instancia lanzada con éxito. ID de la instancia: i-0053a0b0f7af1bd22
[cloudshell-user@ip-10-134-45-7 ~]$
```

	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Comprobación de estado	Estado de la al	Zona de dispon...	DNS de IPv4 pública	Dirección IP...
<input type="checkbox"/>	Sergio-GM-P6Server	i-0fd93400e5cbac2bc	En ejecución	t2.micro	2/2 comprobaciones superadas.	Ver alarmas +	us-east-1a	ec2-3-223-127-118.co...	3.223.127.118
<input type="checkbox"/>	SergioGarcia-P6ServerII	i-0f2f0d0366a6819fa	En ejecución	t2.micro	2/2 comprobaciones superadas.	Ver alarmas +	us-east-1a	ec2-3-235-18-146.com...	3.235.18.146
<input checked="" type="checkbox"/>	script1	i-05d70435e03c72aef	En ejecución	t2.micro	Inicializando	Ver alarmas +	us-east-1a	ec2-3-220-231-188.co...	3.220.231.188
<input type="checkbox"/>	SergioGarcia-P6ServerII	i-0c5363b046b48fb46	Terminada	t2.micro	-	Ver alarmas +	us-east-1a	-	-
<input checked="" type="checkbox"/>	script2	i-0053a0b0f7af1bd22	En ejecución	t2.micro	Inicializando	Ver alarmas +	us-east-1a	ec2-54-160-48-240.co...	54.160.48.240