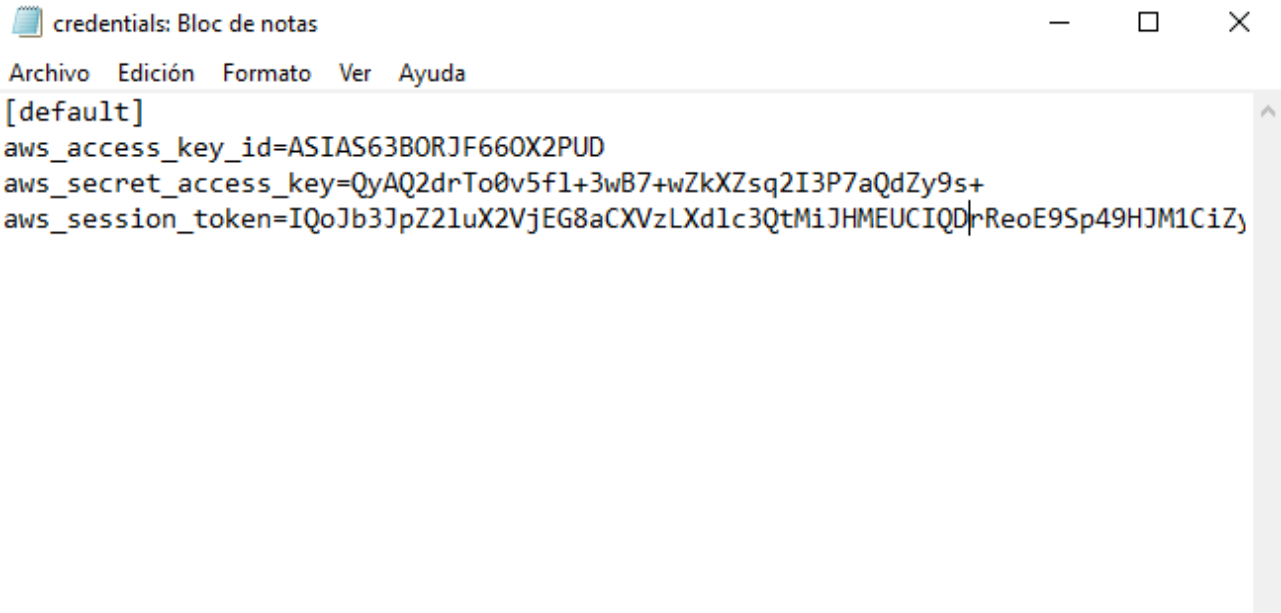


## Entregable 10

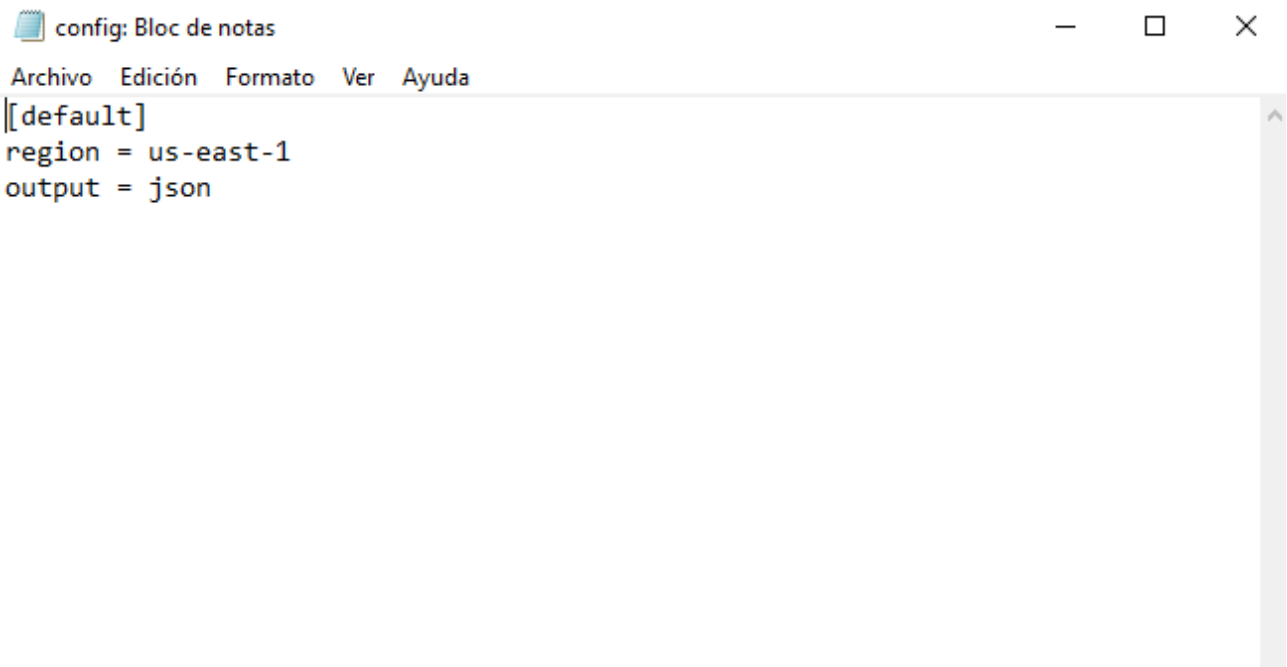
El objetivo de este desafío es crear e interactuar con algunos servicios básicos en AWS utilizando la línea de comandos (AWS CLI)

Una vez instalada la herramienta AWS CLI, procedemos a configurarla, primero en el archivo credentials asignamos el access key, secret key y session token



```
credentials: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
[default]
aws_access_key_id=ASIAS63BORJF660X2PUD
aws_secret_access_key=QyAQ2drTo0v5f1+3wB7+wZkXZsq2I3P7aQdZy9s+
aws_session_token=IQoJb3JpZ2luX2VjEG8aCXVzLXdlc3QtMiJHMEUCIQDrReoE9Sp49HJM1CiZy
```

Luego hacemos lo mismo con el archivo config, agregando la region y el output



```
config: Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
[default]
region = us-east-1
output = json
```

Teniendo un usuario creado en AWS con los permisos correspondientes se procede a crear una instancia de EC2 dentro del free tier, junto a los tags correspondientes

```
C:\Users\Jonathan>aws ec2 run-instances --image-id ami-0c02fb55956c7d316 --count 1 --instance-type t2.micro --key-name vockey --tag-specifications "ResourceType=instance,Tags=[{Key=Owner,Value='Jonathan Sosa'}, {Key=Email,Value='jonathanezequielrosa@hotmail.com'}, {Key=Team,Value='Grupo-Random'}, {Key=ProyectoGrupo-1,Value='Actividad-AWS'}]"
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0c02fb55956c7d316",
      "InstanceId": "i-08e195f4267ba8c3b",
      "InstanceType": "t2.micro",
      "KeyName": "vockey",
      "LaunchTime": "2024-09-22T16:15:08+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1a",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-34-152.ec2.internal",
      "PrivateIpAddress": "172.31.34.152",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-0d1a6076c2cabd97f",
      "VpcId": "vpc-0953171176b6a7804",
    }
  ]
}
```

Creamos tambien el script de usuario que luego correremos al lanzar la instancia, asi instalando apache2



script.sh: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
```

## Procedemos a lanzar la instancia

```
C:\Windows\system32\cmd.exe - aws ec2 run-instances --image-id ami-0c02fb55956c7d316 --count 1 --instance-type t2.micro --user-data file://C:\Users\Jonathan\Desktop\Entregable10\script.sh --key-name vockey

C:\Users\Jonathan>aws ec2 run-instances --image-id ami-0c02fb55956c7d316 --count 1 --instance-type t2.micro --user-data file://C:\Users\Jonathan\Desktop\Entregable10\script.sh --key-name vockey
{
  "Groups": [],
  "Instances": [
    {
      "AmiLaunchIndex": 0,
      "ImageId": "ami-0c02fb55956c7d316",
      "InstanceId": "i-0b18aa044c8e9f139",
      "InstanceType": "t2.micro",
      "KeyName": "vockey",
      "LaunchTime": "2024-09-22T21:39:09+00:00",
      "Monitoring": {
        "State": "disabled"
      },
      "Placement": {
        "AvailabilityZone": "us-east-1a",
        "GroupName": "",
        "Tenancy": "default"
      },
      "PrivateDnsName": "ip-172-31-36-245.ec2.internal",
      "PrivateIpAddress": "172.31.36.245",
      "ProductCodes": [],
      "PublicDnsName": "",
      "State": {
        "Code": 0,
        "Name": "pending"
      },
      "StateTransitionReason": "",
      "SubnetId": "subnet-0d29a23cad1bd3506",
      "VpcId": "vpc-0ef22c6b79eb15048",
    }
  ]
}
```

Habiendo instalado apache en la instancia procedemos a crear y configurar el security group que asociaremos a la misma

```
C:\Users\Jonathan>aws ec2 create-security-group --group-name my-sg --description "Security group for HTTP, HTTPS, SSH"
{
  "GroupId": "sg-0e72263cccb1e6d4"
}
```

```
C:\Users\Jonathan>aws ec2 authorize-security-group-ingress --group-name my-sg --protocol tcp --port 80 --cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-08e4ef95da1f3bde3",
      "GroupId": "sg-0e72263cccb1e6d4",
      "GroupOwnerId": "203678452299",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 80,
      "ToPort": 80,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
```

```
C:\Users\Jonathan>aws ec2 authorize-security-group-ingress --group-name my-sg --protocol tcp --port 443 --cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-094b3fd2a7a85cac2",
      "GroupId": "sg-0e72263cccb1e6d4",
      "GroupOwnerId": "203678452299",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 443,
      "ToPort": 443,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
```

```
C:\Users\Jonathan>aws ec2 authorize-security-group-ingress --group-name my-sg --protocol tcp --port 22 --cidr 0.0.0.0/0
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-091dfd35a2ce744e8",
      "GroupId": "sg-0e72263ccccc1e6d4",
      "GroupOwnerId": "203678452299",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "CidrIpv4": "0.0.0.0/0"
    }
  ]
}
```

De esta manera queda configurado con reglas para permitir tráfico HTTP (80), HTTPS (443), y SSH (22)

Descargamos el .pem del sandbox y procedemos a probar la conexión

[illegible]

vemos que la conexión fue exitosa.

Luego creamos un bucket

```
C:\Users\Jonathan>aws s3api create-bucket --bucket my-unique-bucket-name-9-22 --region us-east-1
{
  "Location": "/my-unique-bucket-name-9-22"
}

C:\Users\Jonathan>
```

y le subimos un archivo

```
C:\Users\Jonathan>aws s3 cp "C:\Users\Jonathan\Desktop\Entregable10.pdf" s3://my-unique-bucket-name-9-22/  
upload: Desktop\Entregable10.pdf to s3://my-unique-bucket-name-9-22/Entregable10.pdf  
  
C:\Users\Jonathan>
```

Creamos un volumen

```
C:\Users\Jonathan>aws ec2 create-volume --size 2 --availability-zone us-east-1a
{
  "AvailabilityZone": "us-east-1a",
  "CreateTime": "2024-09-23T00:08:32+00:00",
  "Encrypted": false,
  "Size": 2,
  "SnapshotId": "",
  "State": "creating",
  "VolumeId": "vol-0553b1f052234a01b",
  "Iops": 100,
  "Tags": [],
  "VolumeType": "gp2",
  "MultiAttachEnabled": false
}
```

lo asociamos

```
C:\Users\Jonathan>aws ec2 attach-volume --volume-id "vol-0553b1f052234a01b" --instance-id "i-09eddb0319ec74721" --device /dev/xvdf
{
  "AttachTime": "2024-09-23T00:49:53.620000+00:00",
  "Device": "/dev/xvdf",
  "InstanceId": "i-09eddb0319ec74721",
  "State": "attaching",
  "VolumeId": "vol-0553b1f052234a01b"
}
```

Y finalmente lo formateamos y montamos en la carpeta /desafios

```
[ec2-user@ip-10-0-0-133 ~]$ sudo mkfs -t ext4 /dev/xvdf
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: 086548d6-6fad-452f-b121-dbc8bbe27e68
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[ec2-user@ip-10-0-0-133 ~]$ sudo mkdir /desafios
[ec2-user@ip-10-0-0-133 ~]$ sudo mount /dev/xvdf /desafios
[ec2-user@ip-10-0-0-133 ~]$ echo '/dev/xvdf /desafios ext4 defaults 0 0' | sudo tee -a /etc/fstab
/dev/xvdf /desafios ext4 defaults 0 0
```

Copiamos el archivo del bucket al nuevo volumen

```
[ec2-user@ip-10-0-0-133 ~]$ aws s3 cp s3://my-unique-bucket-name-9-22/Entregable10.pdf /desafios/file.pdf
download: s3://my-unique-bucket-name-9-22/Entregable10.pdf to ../../desafios/file.pdf
[ec2-user@ip-10-0-0-133 ~]$ ls /desafios/
file.pdf
```

Para finalizar limpiamos los recursos con los comandos:

```
aws ec2 terminate-instances --instance-ids <instance-id>
```

```
aws ec2 delete-volume --volume-id <volume-id>
```

```
aws s3 rb s3://my-unique-bucket-name --force
```