Uso de AWS CLI

Ivan Santiago Forero Torres

Universidad Escuela Colombiana de ingeniería Julio Garavito

AREP (Arquitectura empresarial)

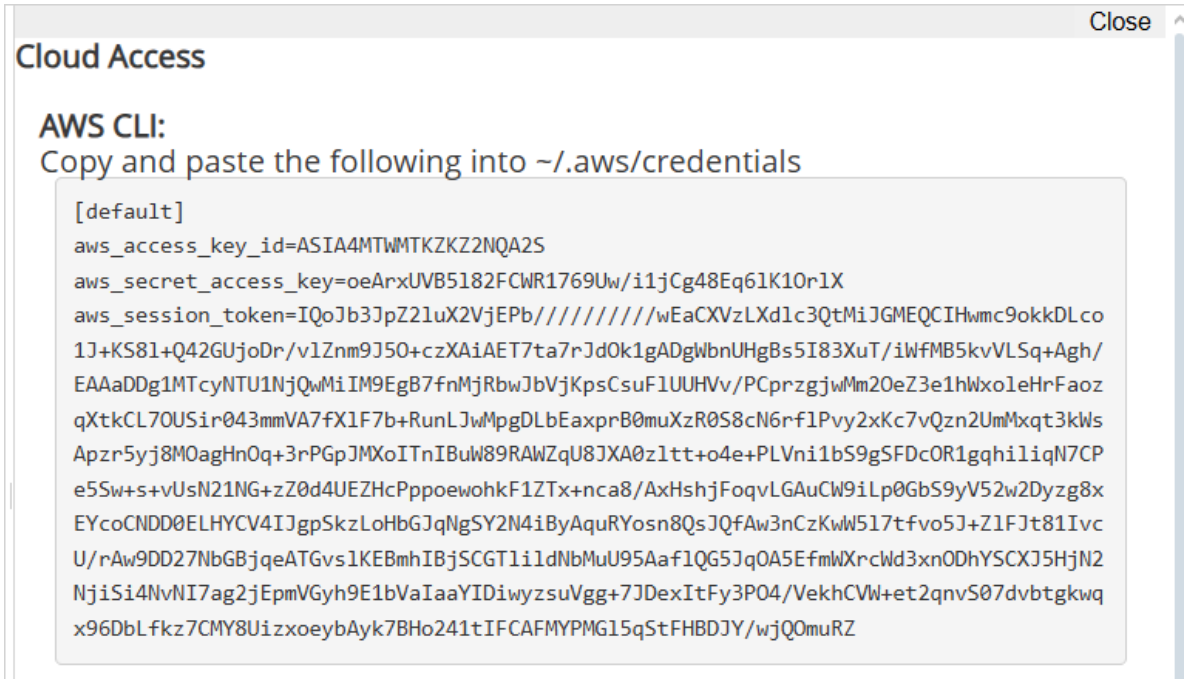
PhD. Luis Daniel Benavidez Navarro

25 de septiembre 2025

USO DE AMAZON CLI

- Primero vamos a descargar AWS CLI desde el browser, luego de tenerlo vamos a realizar los siguientes pasos.
- Vamos a escribir aws configure y se vera de la siguiente forma:

Los valores que se ven en la imagen fueron sacados de la información del laboratorio:



- Veremos que se crearon los dos archivos el de credenciales y el de configuración:

```
C:\Users\ivans>cd .aws

C:\Users\ivans\.aws>dir
 El volumen de la unidad C es Windows-SSD
 El número de serie del volumen es: 52FA-F0E0

 Directorio de C:\Users\ivans\.aws

25/09/2025  06:03 p. m.    <DIR>          .
25/09/2025  06:03 p. m.    <DIR>          ..
25/09/2025  06:03 p. m.                46 config
25/09/2025  06:03 p. m.               933 credentials
               2 archivos            979 bytes
               2 dirs  168.212.393.984 bytes libres

C:\Users\ivans\.aws>notepad config

C:\Users\ivans\.aws>notepad credentials
```

- Verificamos el contenido de los archivos.

```
[default]
aws_access_key_id = ASIA4MTWMTKZKZ2NQA2S
aws_secret_access_key = oeArxUVB5l82FCWR1769Uw/i1jCg48Eq6lK1OrlX
aws_session_token =
IQoJb3JpZ2luX2VjEPb/////////wEaCXVzLXdlc3QtMiJGMEQCIHwmc9okkDLco1J+KS8l+Q42GUjoDr/vlZnm9J5O+cz
XAiAET7ta7rJdOk1gADgWbnUHgBs5l83XuT/iWfMB5kvVLSq+Agh/EAAaDDg1MTcyNTU1NjQwMiIM9EgB7fnMjRbwJ
bVjKpsCsuFlUUHVv/PCprzgjwMm2OeZ3e1hWxoleHrFaozqXtkCL7OUSir043mmVA7fXlF7b+RunLJwMpgDLbEaxprB0m
uXzR0S8cN6rflPvy2xKc7vQzn2UmMxqt3kWsApzr5yj8MOagHnOq+
3rPGpJMXoITnIBuW89RAWZqU8JXA0zltt+o4e+PLVni1bS9gSFDcOR1gqhiliqN7CPe5Sw+s+vUsN21NG+zZ0d4UEZHcP
ppoewohkF1ZTx+nca8/AxHshjFoqvLGAuCW9iLp0GbS9yV52w2Dyzg8xEYcoCNDD0ELHYCV4IJgpSkzLoHbGJqNgSY2N
4iByAquRYosn8QsJQfAw3nCzKwW5l7tfvo5J+ZlFJt81IvcU/rAw9DD27NbGBjqeATGvslKEBmhIBjSCGTlildNbMuU95Aaf
lQG5JqOA5EfmWXrcWd3xnODhYSCXJ5HjN2NjiSi4NvNI7ag2jEpmVGyh9E1bVaIaaYIDiwyzsuVgg+
7JDexItFy3PO4/VekhCVW+et2qnvS07dvbtgkwqx96DbLfkz7CMY8UizxoeybAyk7BHo241tIFCAFMYPMGl5qStFHBDJY/
wjQOmuRZ
```

- Creamos una llave de seguridad llamada MyKeyPair y la guardamos en

  MyKeyPair.pem, luego le damos permisos de solo lectura a la llave.

```
C:\Users\ivans\.aws>aws ec2 create-key-pair --key-name MyKeyPair --query "KeyMaterial" --output text > MyKeyPair.pem

C:\Users\ivans\.aws>notepad MyKeyPair.pem

C:\Users\ivans\.aws>chmod 400 MyKeyPair.pem
```

- Hacemos la verificación de los permisos.

```
C:\Users\ivans\.aws>ls -la
total 25
drwxr-xr-x 1 ivans 197609    0 Sep 25 18:06 .
drwxr-xr-x 1 ivans 197609    0 Sep 25 19:31 ..
-r--r--r-- 1 ivans 197609 1702 Sep 25 18:06 MyKeyPair.pem
-rw-r--r-- 1 ivans 197609   46 Sep 25 18:04 config
-rw-r--r-- 1 ivans 197609  933 Sep 25 18:03 credentials
```

- Verificamos la información que esta dentro de la llave.

```
C:\Users\ivans\.aws>aws ec2 describe-key-pairs --key-name MyKeyPair
{
    "KeyPairs": [
        {
            "KeyPairId": "key-01637af0f3bffd922",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2025-09-25T23:06:51.279000+00:00",
            "KeyName": "MyKeyPair",
            "KeyFingerprint": "4d:c1:8c:42:12:43:05:ed:c1:2f:e5:d1:ab:20:f2:17:ec:4c:99:70"
        }
    ]
}
```

- Con el primer comando lo que hacemos es listar el VPC que se encuentra por default en la cuenta, ya teniéndolo identificado el VPC creamos un nuevo grupo de seguridad.

```
C:\Users\ivans\.aws>aws ec2 describe-vpcs --filters Name=isDefault,Values=true --query "Vpcs[0].VpcId" --output text
vpc-0f5789b52fa5d380f


C:\Users\ivans\.aws>aws ec2 create-security-group --group-name my-sg-cli --description "My security group" --vpc-id vpc-0f5789b52fa5d380f
{
    "GroupId": "sg-0dc70906ea17fb0b9",
    "SecurityGroupArn": "arn:aws:ec2:us-east-1:851725556402:security-group/sg-0dc70906ea17fb0b9"
}
```

- Verificamos la información de nuestro grupo de seguridad.

```
C:\Users\ivans\.aws>aws ec2 describe-security-groups --group-ids sg-0dc70906ea17fb0b9
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0dc70906ea17fb0b9",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0f5789b52fa5d380f",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:851725556402:security-group/sg-0dc70906ea17fb0b9",
            "OwnerId": "851725556402",
            "GroupName": "my-sg-cli",
            "Description": "My security group",
            "IpPermissions": []
        }
    ]
}
```

- Con el primer comando verificamos nuestra Ip pública y con el segundo agregamos una regla de entrada a nuestro grupo de seguridad, al protocolo tcp con el puerto 3389.

```
C:\Users\ivans\.aws>curl https://checkip.amazonaws.com
181.32.31.197

C:\Users\ivans\.aws>aws ec2 authorize-security-group-ingress --group-id sg-0dc70906ea17fb0b9 --protocol tcp --port 3389 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-090e28d10320468a1",
            "GroupId": "sg-0dc70906ea17fb0b9",
            "GroupOwnerId": "851725556402",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 3389,
            "ToPort": 3389,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:851725556402:security-group-rule/sgr-090e28d10320468a1"
        }
    ]
}
```

- Agregamos una segunda regla de entrada al puerto 22.

```
C:\Users\ivans\.aws>aws ec2 authorize-security-group-ingress --group-id sg-0dc70906ea17fb0b9 --protocol tcp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0d279d7199340626f",
            "GroupId": "sg-0dc70906ea17fb0b9",
            "GroupOwnerId": "851725556402",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:851725556402:security-group-rule/sgr-0d279d7199340626f"
        }
    ]
}
```

- Con el siguiente comando lo que hacemos es listar las subnets disponibles para lanzar nuestra instancia.

```
C:\Users\ivans\.aws>aws ec2 describe-subnets --filters Name=vpc-id,Values=vpc-0f5789b52fa5d380f --query "Subnets[].{ID:SubnetId,AZ:AvailabilityZone}" --outp
ut table
-------------------------------------------
|              DescribeSubnets            |
+-------------+---------------------------+
|     AZ      |            ID             |
+-------------+---------------------------+
|  us-east-1a |  subnet-0f6d748d7a7c9ef3a  |
|  us-east-1c |  subnet-03a8b270b20105187  |
|  us-east-1d |  subnet-083e61806576a4a25  |
|  us-east-1f |  subnet-0c0efdc9e47aed6cc  |
|  us-east-1b |  subnet-071949b350b202656  |
|  us-east-1e |  subnet-0b441677ae9fc76ab  |
+-------------+---------------------------+
```

- El siguiente comando nos perfimete escoger de manera automática el último AMI de Amazon Linux

```
C:\Users\ivans\.aws>aws ssm get-parameters --names /aws/service/ami-amazon-linux-latest/al2023-ami-kernel-6.1-x86_64 --query "Parameters[0].Value" --output
text
ami-08982f1c5bf93d976
```

- El siguiente comando nos permite lanzar la instancia con los parámetros especificados.

```
C:\Users\ivans\.aws>aws ec2 run-instances --image-id ami-08982f1c5bf93d976 --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-0
dc70906ea17fb0b9 --subnet-id subnet-0f6d748d7a7c9ef3a --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=cli-demo}]"
{
```

- Veremos que se despliega toda la información.


```
{
    "ReservationId": "r-0db71ce41ee18e1b7",
    "OwnerId": "851725556402",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "a23829f1-d94b-43a8-9dd3-14d51a1f2f2a",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-09-26T03:14:13+00:00",
                        "AttachmentId": "eni-attach-0a224ca91aac7e178",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-0dc70906ea17fb0b9",
                            "GroupName": "my-sg-cli"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "0e:06:d4:d6:8c:15",
                    "NetworkInterfaceId": "eni-02bc2ed3840acc983",
                    "OwnerId": "851725556402",
                    "PrivateDnsName": "ip-172-31-41-237.ec2.internal",
                    "PrivateIpAddress": "172.31.41.237",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
```

```
            {
                "Primary": true,
                "PrivateDnsName": "ip-172-31-41-237.ec2.internal",
                "PrivateIpAddress": "172.31.41.237"
            }
        ],
        "SourceDestCheck": true,
        "Status": "in-use",
        "SubnetId": "subnet-0f6d748d7a7c9ef3a",
        "VpcId": "vpc-0f5789b52fa5d380f",
        "InterfaceType": "interface",
        "Operator": {
            "Managed": false
        }
    }
],
"RootDeviceName": "/dev/xvda",
"RootDeviceType": "ebs",
"SecurityGroups": [
    {
        "GroupId": "sg-0dc70906ea17fb0b9",
        "GroupName": "my-sg-cli"
    }
],
"SourceDestCheck": true,
"StateReason": {
    "Code": "pending",
    "Message": "pending"
},
"Tags": [
    {
        "Key": "Name",
        "Value": "cli-demo"
    }
],
"VirtualizationType": "hvm",
"CpuOptions": {
    "CoreCount": 1,
    "ThreadsPerCore": 1
},
"CapacityReservationSpecification": {
```

```
},
"CapacityReservationSpecification": {
    "CapacityReservationPreference": "open"
},
"MetadataOptions": {
    "State": "pending",
    "HttpTokens": "required",
    "HttpPutResponseHopLimit": 2,
    "HttpEndpoint": "enabled",
    "HttpProtocolIpv6": "disabled",
    "InstanceMetadataTags": "disabled"
},
"EnclaveOptions": {
    "Enabled": false
},
"BootMode": "uefi-preferred",
"PrivateDnsNameOptions": {
    "HostnameType": "ip-name",
    "EnableResourceNameDnsARecord": false,
    "EnableResourceNameDnsAAAARecord": false
},
"MaintenanceOptions": {
    "AutoRecovery": "default",
    "RebootMigration": "default"
},
"CurrentInstanceBootMode": "legacy-bios",
"Operator": {
    "Managed": false
},
"InstanceId": "i-0ccfc767f4b2e682e",
"ImageId": "ami-08982f1c5bf93d976",
"State": {
    "Code": 0,
    "Name": "pending"
},
"PrivateDnsName": "ip-172-31-41-237.ec2.internal",
"PublicDnsName": "",
"StateTransitionReason": "",
"KeyName": "MyKeyPair",
"AmiLaunchIndex": 0,
"ProductCodes": [],
```

```
                "Name": "pending"
            },
            "PrivateDnsName": "ip-172-31-41-237.ec2.internal",
            "PublicDnsName": "",
            "StateTransitionReason": "",
            "KeyName": "MyKeyPair",
            "AmiLaunchIndex": 0,
            "ProductCodes": [],
            "InstanceType": "t2.micro",
            "LaunchTime": "2025-09-26T03:14:13+00:00",
            "Placement": {
                "AvailabilityZoneId": "use1-az6",
                "GroupName": "",
                "Tenancy": "default",
                "AvailabilityZone": "us-east-1a"
            },
            "Monitoring": {
                "State": "disabled"
            },
            "SubnetId": "subnet-0f6d748d7a7c9ef3a",
            "VpcId": "vpc-0f5789b52fa5d380f",
            "PrivateIpAddress": "172.31.41.237"
        }
    ]
}
```


- El siguiente comando permite listar el dominio dado Dns público de aws


```
C:\Users\ivans\.aws>aws ec2 describe-instances --instance-ids i-0ccfc767f4b2e682e --query "Reservations[0].Instances[0].PublicDnsName" --output text
ec2-98-83-206-180.compute-1.amazonaws.com
```


- Con el siguiente comando nos conectamos a la instancia


```
C:\Users\ivans\.aws>ssh -i "C:\Users\ivans\.aws\MyKeyPair.pem" ec2-user@ec2-98-83-206-180.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```


- Luego de comprobar la conexión nos salimos, y allí listamos por el identificador las instancias que cumplen la condición t2.micro


```
C:\Users\ivans\.aws>aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query "Reservations[].Instances[].InstanceId"
[
    "i-0ccfc767f4b2e682e"
]
```

- Con los dos siguientes comandos terminamos la ejecución de la instancia y eliminamos el grupo de seguridad.

```
C:\Users\ivans\.aws>aws ec2 terminate-instances --instance-ids i-0ccfc767f4b2e682e
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-0ccfc767f4b2e682e",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}


C:\Users\ivans\.aws>aws ec2 delete-security-group --group-id sg-0dc70906ea17fb0b9
{
    "Return": true,
    "GroupId": "sg-0dc70906ea17fb0b9"
}
```

- Con los dos siguientes comandos eliminamos el par de llaves y el .pem de la maquina local.

```
C:\Users\ivans\.aws>aws ec2 delete-key-pair --key-name MyKeyPair
{
    "Return": true,
    "KeyPairId": "key-01637af0f3bffd922"
}


C:\Users\ivans\.aws>del /f "C:\Users\ivans\.aws\MyKeyPair.pem"
```