Infrastructure as Code: AWS-CLI Primer Workshop

AWS-CLI Primer Workshop

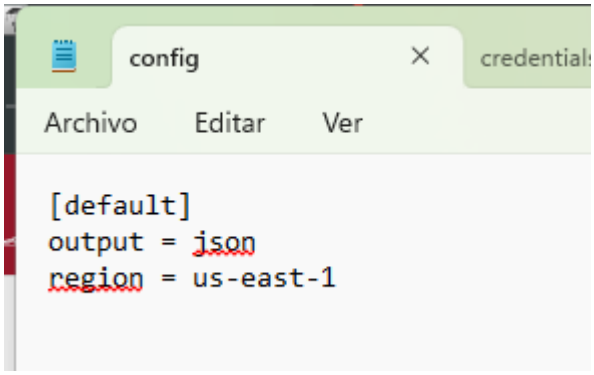
María Paula Sánchez Macías

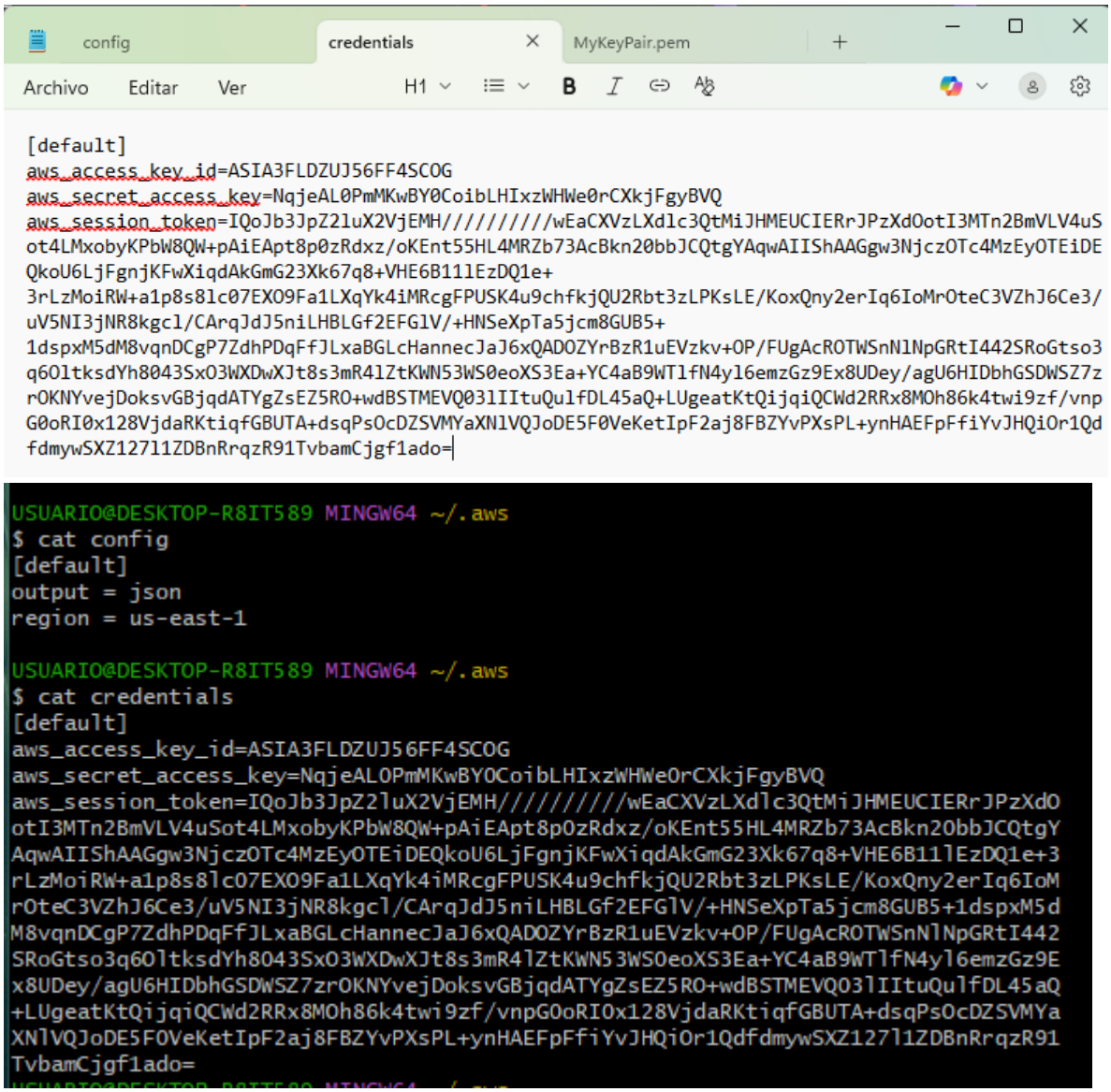
1. Instalé AWS CLI
2. **Configuré AWS CLI** con mis credenciales

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Videos/AREP5
$ aws configure
AWS Access Key ID [****************MPLE]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [****************EKEY]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAM
PLEKEY
Default region name [None]: us-west-2
Default output format [USUARIO@DESKTOP-R8IT589 MINGW64 ~]: json
```

3. Ingresé al directorio .aws y configuré correctamente ambos archivos (config y credentials)

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Videos/AREP5
$ cd

USUARIO@DESKTOP-R8IT589 MINGW64 ~
$ cd .aws

USUARIO@DESKTOP-R8IT589 MINGW64 ~/.aws
$ dir
config  credentials
```





Step 1: Create a Key Pair for EC2

1. Cambié la carpeta y creé una "llave digital" que me permitirá conectarme de forma segura a mi servidor EC2.

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text > MyKeyPair.pe
m
ls
MyKeyPair.pem

USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ ls -la
total 8
drwxr-xr-x 1 USUARIO 197121    0 Sep 23 11:34 ./
drwxr-xr-x 1 USUARIO 197121    0 Sep 23 11:28 ../
-rw-r--r-- 1 USUARIO 197121 1702 Sep 23 11:35 MyKeyPair.pem
```

2. Cambié los permisos para que solo yo pueda leer la llave

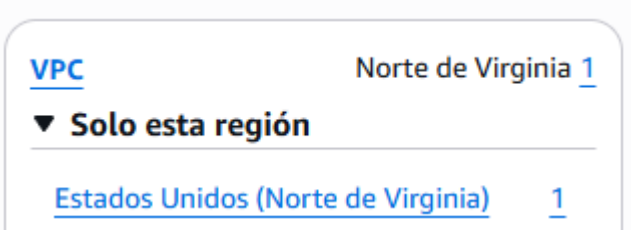
```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ chmod 400 MyKeyPair.pem

USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ ls -la
total 8
drwxr-xr-x 1 USUARIO 197121    0 Sep 23 11:34 ./
drwxr-xr-x 1 USUARIO 197121    0 Sep 23 11:28 ../
-r--r--r-- 1 USUARIO 197121 1702 Sep 23 11:35 MyKeyPair.pem
```

3. Check the fingerprint

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 describe-key-pairs --key-name MyKeyPair
{
    "KeyPairs": [
        {
            "KeyPairId": "key-02ff3bb3315400d4d",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2025-09-23T16:35:43.882000+00:00",
            "KeyName": "MyKeyPair",
            "KeyFingerprint": "5b:be:34:03:34:90:65:74:41:64:1c:93:db:12:4a:ac:96:ac:e4:fe"
        }
    ]
}
```

Step 2: Create a Security Group

1. **Revisé VPCs configured in your account**




```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 create-security-group --group-name my-sg-cli --description "My security group" --vpc-id vpc-0e6a25b4409618b7f
{
    "GroupId": "sg-009e3436f77ac4998",
    "SecurityGroupArn": "arn:aws:ec2:us-east-1:767397831291:security-group/sg-009e3436f77ac4998"
}
```

2. Ver la lista de grupos

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 describe-security-groups --group-ids sg-009e3436f77ac4998
{
    "SecurityGroups": [
        {
            "GroupId": "sg-009e3436f77ac4998",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0e6a25b4409618b7f",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:767397831291:security-group/sg-009e3436f77ac4998",
            "OwnerId": "767397831291",
            "GroupName": "my-sg-cli",
            "Description": "My security group",
            "IpPermissions": []
        }
    ]
}
```

3. Allow RDP (port 3389):

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 authorize-security-group-ingress --group-id sg-009e3436f77ac4998 --protocol tcp --port 3389 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0fbbc770b5af3545f",
            "GroupId": "sg-009e3436f77ac4998",
            "GroupOwnerId": "767397831291",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 3389,
            "ToPort": 3389,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:767397831291:security-group-rule/sgr-0fbbc770b5af3545f"
        }
    ]
}
```

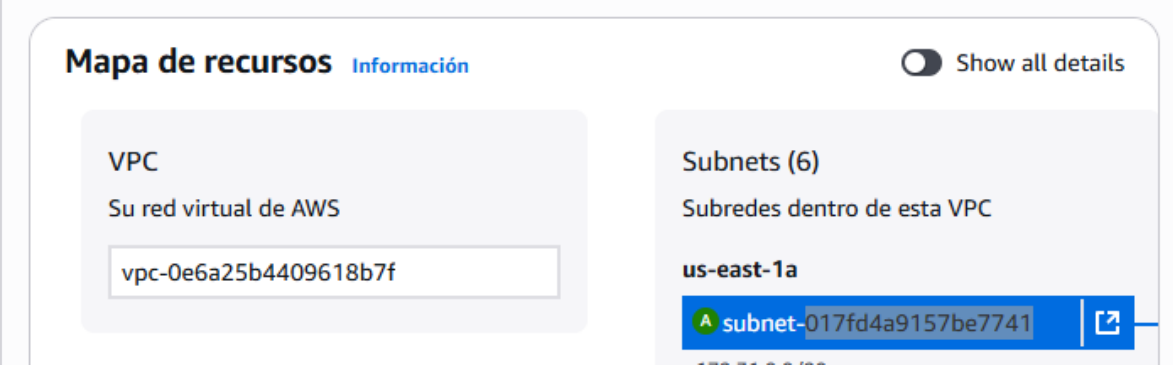
4. Allow SSH (port 22):

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 authorize-security-group-ingress --group-id sg-009e3436f77ac4998 --protocol tcp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0bbcff5fb7d360f11",
            "GroupId": "sg-009e3436f77ac4998",
            "GroupOwnerId": "767397831291",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:767397831291:security-group-rule/sgr-0bbcff5fb7d360f11"
        }
    ]
}
```

Step 3: Create the Instance

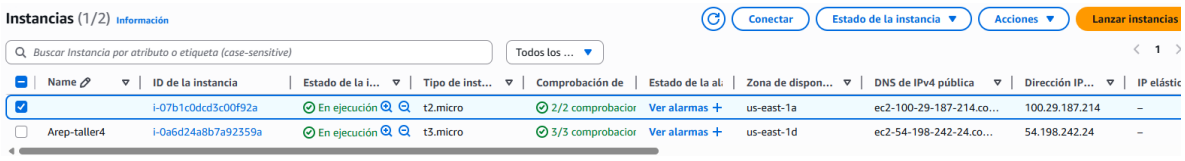
1. Before creating the instance, ensure you have a subnet configured.

   Run the following command to launch a **t2.micro** instance:



```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 authorize-security-group-ingress --group-id sg-009e3436f77ac4998 --protocol tcp --port 3389 --cidr 0.0.0.0/0
{
    "SecurityGroups": [
        {
            "GroupId": "sg-009e3436f77ac4998",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0e6a25b4409618b7f",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:767397831291:security-group/sg-009e3436f77ac4998",
            "OwnerId": "767397831291",
            "GroupName": "my-sg-cli",
            "Description": "My security group",
            "IpPermissions": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 3389,
                    "ToPort": 3389,
                    "UserIdGroupPairs": [],
{
    "ReservationId": "r-07147a262ee4eb390",
    "OwnerId": "767397831291",
    "Groups": [],
```

Step 4: Connect to the Instance




```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ ssh -i "MyKeyPair.pem" ec2-user@ec2-100-29-187-214.compute-1.amazonaws.com
The authenticity of host 'ec2-100-29-187-214.compute-1.amazonaws.com (100.29.187.214)' can't be established.
ED25519 key fingerprint is SHA256:tn665KMLBYtzOuJvxearetWEfGfPsJfm5tNykx2qzlI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-100-29-187-214.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
27 package(s) needed for security, out of 44 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-0-131 ~]$
```

Step 5: List Your Instances

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query "Reservations[].Instances[].InstanceId"
[
    "i-07b1c0dcd3c00f92a"
]
```

Step 6: Clean Up

1. Delete the key pair:

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 delete-key-pair --key-name MyKeyPair
{
    "Return": true,
    "KeyPairId": "key-02ff3bb3315400d4d"
}
```

2. Delete the security group and Terminate the instance

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 terminate-instances --instance-ids i-07b1c0dcd3c00f92a
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-07b1c0dcd3c00f92a",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

```
USUARIO@DESKTOP-R8IT589 MINGW64 ~/Documents/AREP5
$ aws ec2 delete-security-group --group-id sg-009e3436f77ac4998
{
    "Return": true,
    "GroupId": "sg-009e3436f77ac4998"
}
```