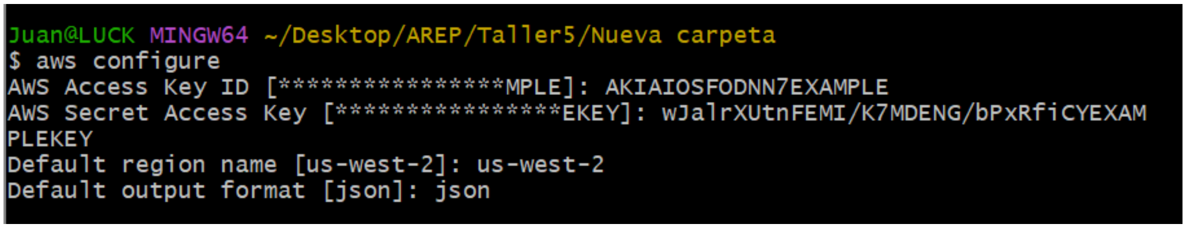
Infrastructure as Code: AWS-CLI

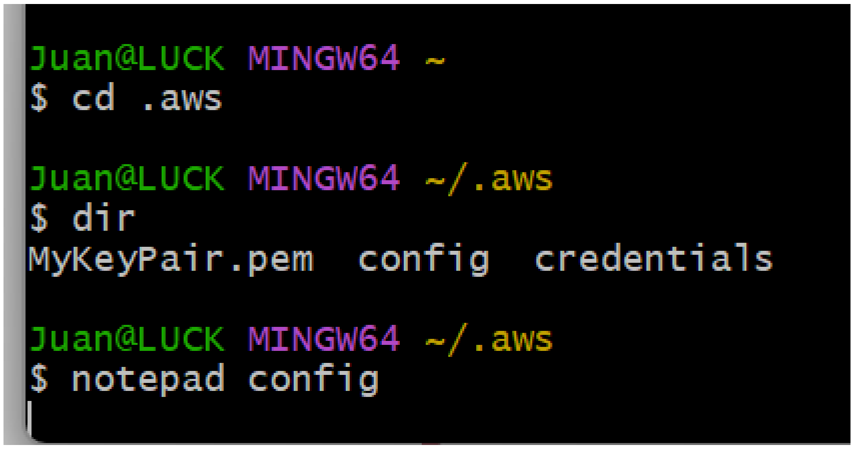
Juan David Rodriguez Rodriguez

1. We install AWS CLI
2. We configure AWS CLI with the credentials



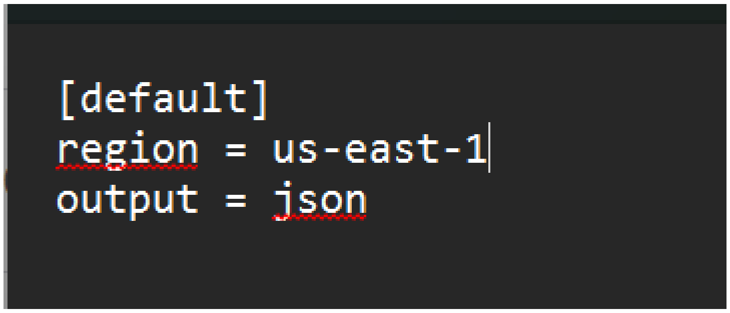
```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/Nueva carpeta
$ aws configure
AWS Access Key ID [****************MPLE]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [****************EKEY]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAM
PLEKEY
Default region name [us-west-2]: us-west-2
Default output format [json]: json
```

3. We go into the **.aws** directory and properly configure both files (**config** and **credentials**)



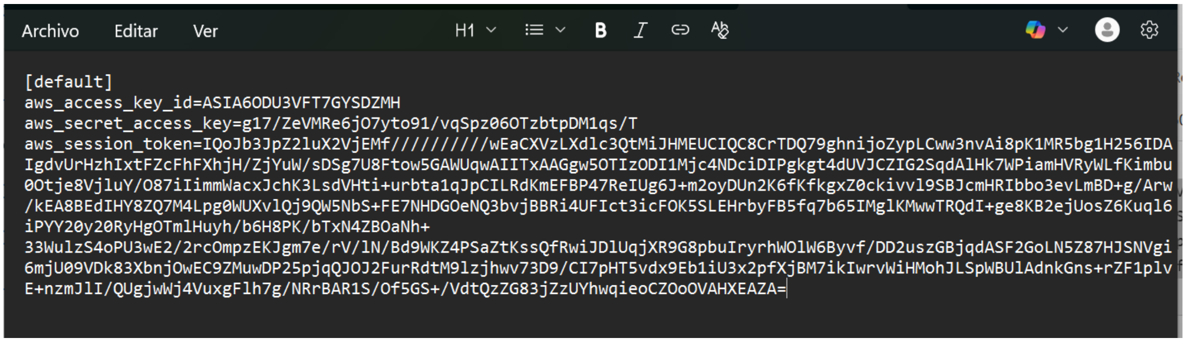
```
Juan@LUCK MINGW64 ~
$ cd .aws

Juan@LUCK MINGW64 ~/.aws
$ dir
MyKeyPair.pem  config  credentials

Juan@LUCK MINGW64 ~/.aws
$ notepad config
```

4. Config



```
[default]
region = us-east-1
output = json
```

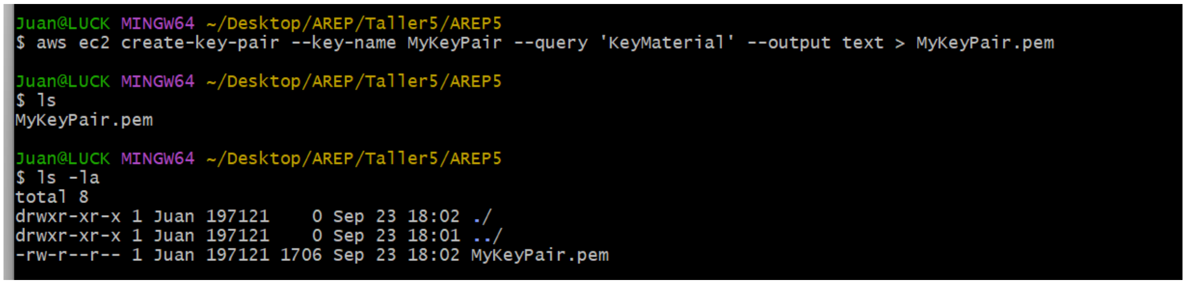
5. Credentials (found in the AWS section under **details**)



```
[default]
aws_access_key_id=ASIA6ODU3VFT7GYSDZMH
aws_secret_access_key=g17/ZeVMRe6jO7yto91/vqSpz06OTzbtpDM1qs/T
aws_session_token=IQoJb3JpZ2luX2VjEMf//////////wEaCXVzLXdlc3QtMiJHMEUCIQC8CrTDQ79ghnijoZypLCww3nvAi8pK1MR5bg1H256IDA
IgdvUrHzhIxtFZcFhFXhjH/ZjYuW/sDSg7U8Ftow5GAWUqwAIITxAAGgw5OTIzODI1Mjc4NDciDIPgkgt4dUVJCZIG2SqdAlHk7WPiamHVRyWLfKimbu
0Otje8VjluY/O87iIimmWacxJchK3LsdVHti+urbta1qJpCILRdKmEFBP47ReIUg6J+m2oyDUn2K6fKfkgxZ0ckivvl9SBJcmHRIbbo3evLmBD+g/Arw
/kEA8BEdIHY8ZQ7M4Lpg0WUXvlQj9QW5NbS+FE7NHDGOeNQ3bvjBBRi4UFIct3icFOK5SLEHrbyFB5fq7b65IMglKMwwTRQdI+ge8KB2ejUosZ6Kuql6
iPYY20y20RyHgOTmlHuyh/b6H8PK/bTxN4ZBOaNh+
33WulzS4oPU3wE2/2rcOmpzEKJgm7e/rV/lN/Bd9WKZ4PSaZtKssQfRwiJDlUqjXR9G8pbuIryrhWOlW6Byvf/DD2uszGBjqdASF2GoLN5Z87HJSNVgi
6mjU09VDk83XbnjOwEC9ZMuwDP25pjqQJOJ2FurRdtM9lzjhwv73D9/CI7pHT5vdx9Eb1iU3x2pfXjBM7ikIwrvWiHMohJLSpWBUlAdnkGns+rZF1plv
E+nzmJlI/QUgjwWj4VuxgFlh7g/NRrBAR1S/Of5GS+/VdtQzZG83jZzUYhwqieoCZOoOVAHXEAZA=
```

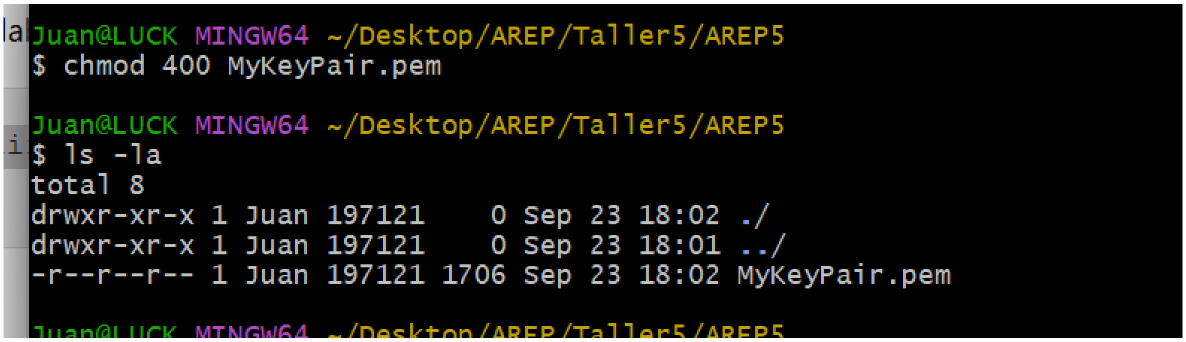
Step 1: Create a Key Pair for EC2

1. We change the folder and create a **digital key** that will allow me to securely connect to the EC2 server.



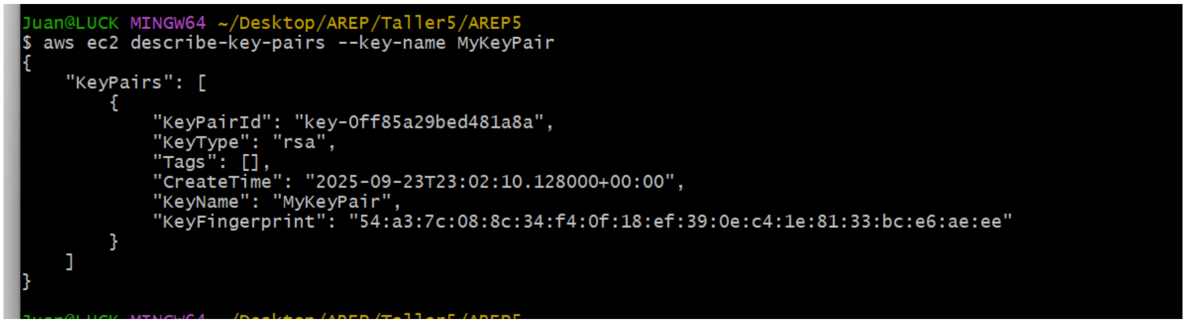
```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text > MyKeyPair.pem

Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ ls
MyKeyPair.pem

Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ ls -la
total 8
drwxr-xr-x 1 Juan 197121    0 Sep 23 18:02 ./
drwxr-xr-x 1 Juan 197121    0 Sep 23 18:01 ../
-rw-r--r-- 1 Juan 197121 1706 Sep 23 18:02 MyKeyPair.pem
```

2. We change the permissions so that only we can read the key.



```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ chmod 400 MyKeyPair.pem

Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ ls -la
total 8
drwxr-xr-x 1 Juan 197121    0 Sep 23 18:02 ./
drwxr-xr-x 1 Juan 197121    0 Sep 23 18:01 ../
-r--r--r-- 1 Juan 197121 1706 Sep 23 18:02 MyKeyPair.pem

Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
```

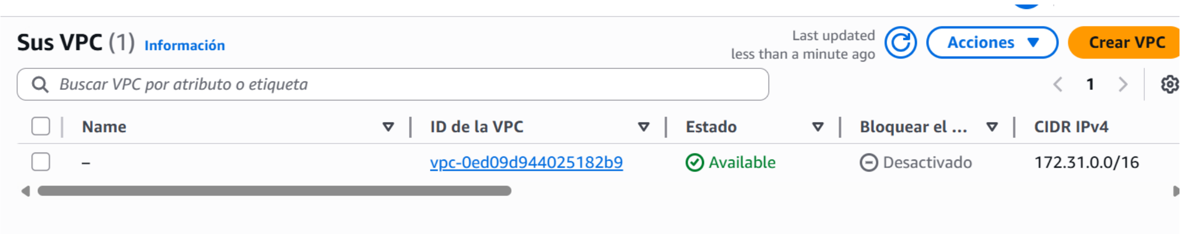
3. Check the fingerprint



```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 describe-key-pairs --key-name MyKeyPair
{
    "KeyPairs": [
        {
            "KeyPairId": "key-0ff85a29bed481a8a",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2025-09-23T23:02:10.128000+00:00",
            "KeyName": "MyKeyPair",
            "KeyFingerprint": "54:a3:7c:08:8c:34:f4:0f:18:ef:39:0e:c4:1e:81:33:bc:e6:ae:ee"
        }
    ]
}
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
```

Step 2: Create a Security Group

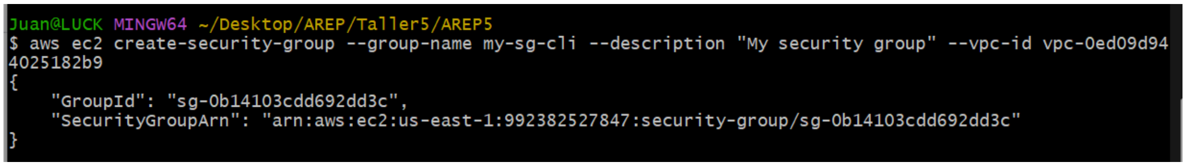
1. We check the VPCs configured in our account.



Sus VPC (1) Información
Last updated less than a minute ago
Acciones ▼
Crear VPC
Buscar VPC por atributo o etiqueta
1

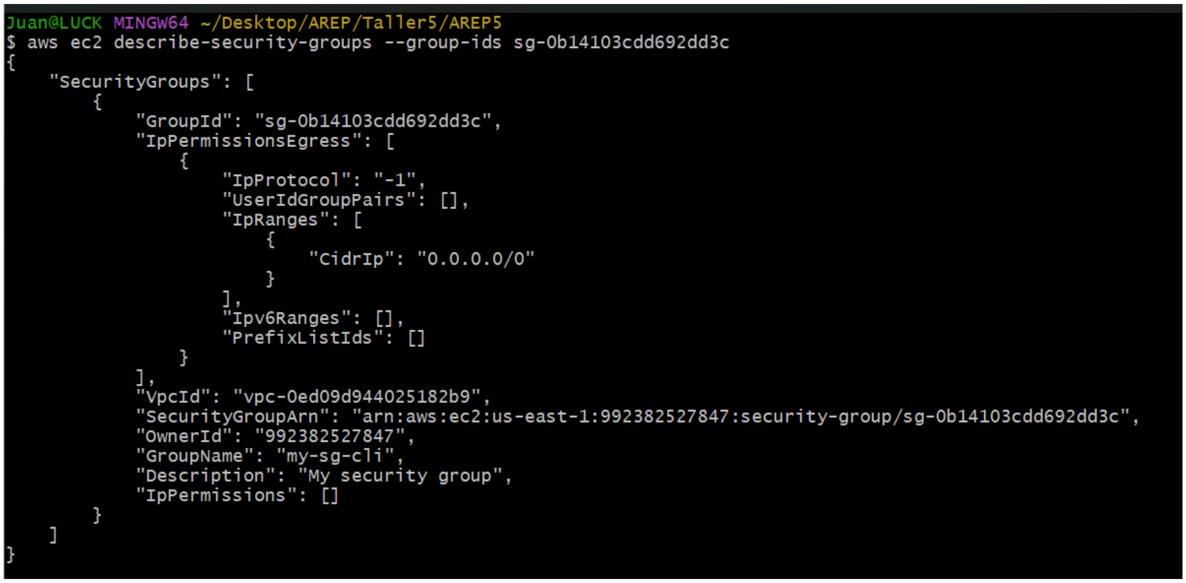
□	Name	ID de la VPC	Estado	Bloquear el ...	CIDR IPv4
□	–	vpc-0ed09d944025182b9	✓ Available	⊖ Desactivado	172.31.0.0/16

2. We run the following command with our own VPC, and this way we obtain the **GroupId**.



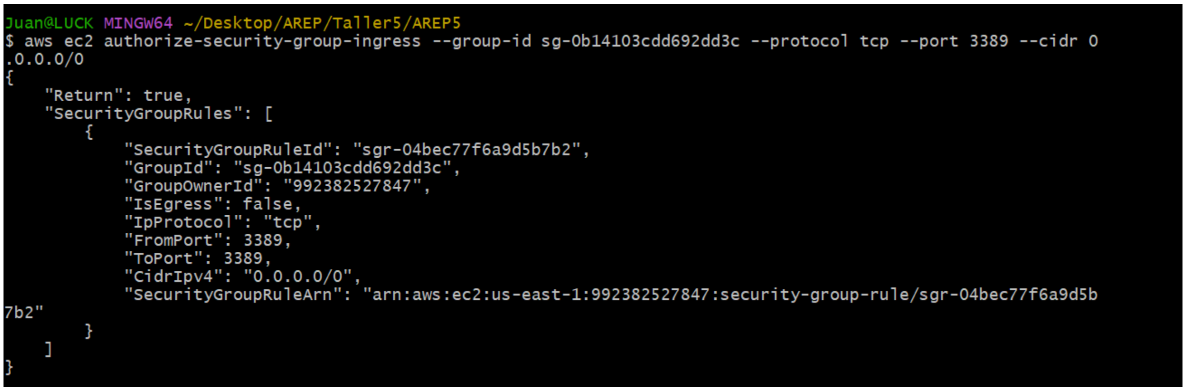
```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 create-security-group --group-name my-sg-cli --description "My security group" --vpc-id vpc-0ed09d94
4025182b9
{
    "GroupId": "sg-0b14103cdd692dd3c",
    "SecurityGroupArn": "arn:aws:ec2:us-east-1:992382527847:security-group/sg-0b14103cdd692dd3c"
}
```

3. View the list of groups.



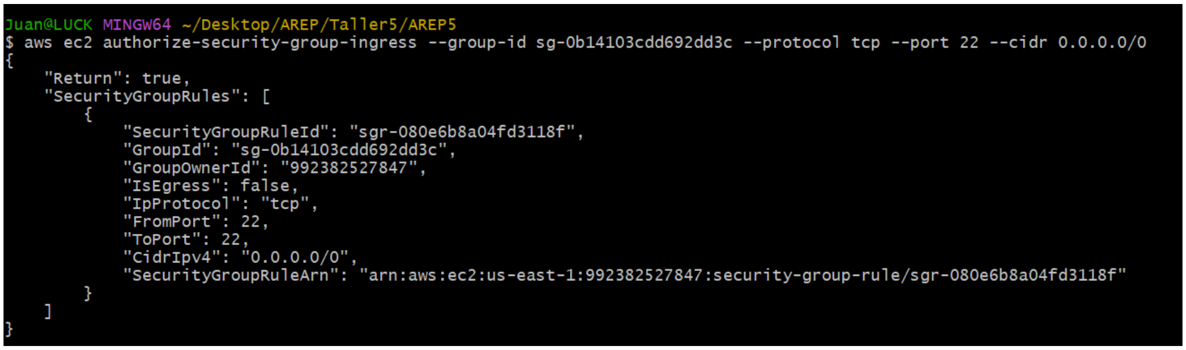
```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 describe-security-groups --group-ids sg-0b14103cdd692dd3c
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0b14103cdd692dd3c",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0ed09d944025182b9",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:992382527847:security-group/sg-0b14103cdd692dd3c",
            "OwnerId": "992382527847",
            "GroupName": "my-sg-cli",
            "Description": "My security group",
            "IpPermissions": []
        }
    ]
}
```

4. Allow RDP (port 3389):



```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 authorize-security-group-ingress --group-id sg-0b14103cdd692dd3c --protocol tcp --port 3389 --cidr 0
.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-04bec77f6a9d5b7b2",
            "GroupId": "sg-0b14103cdd692dd3c",
            "GroupOwnerId": "992382527847",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 3389,
            "ToPort": 3389,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:992382527847:security-group-rule/sgr-04bec77f6a9d5b
7b2"
        }
    ]
}
```

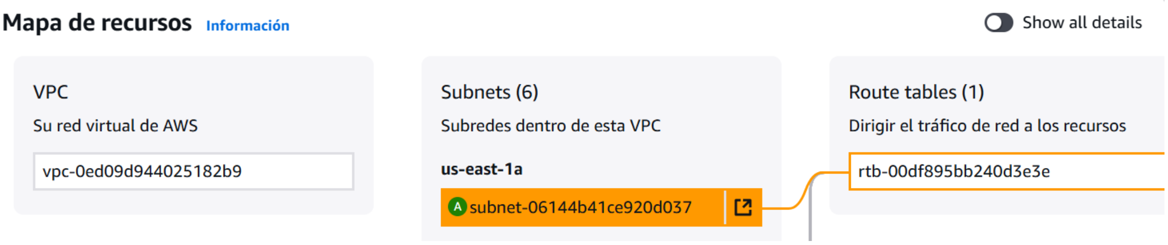
5. Allow SSH (port22)



```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 authorize-security-group-ingress --group-id sg-0b14103cdd692dd3c --protocol tcp --port 22 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-080e6b8a04fd3118f",
            "GroupId": "sg-0b14103cdd692dd3c",
            "GroupOwnerId": "992382527847",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:992382527847:security-group-rule/sgr-080e6b8a04fd3118f"
        }
    ]
}
```

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:



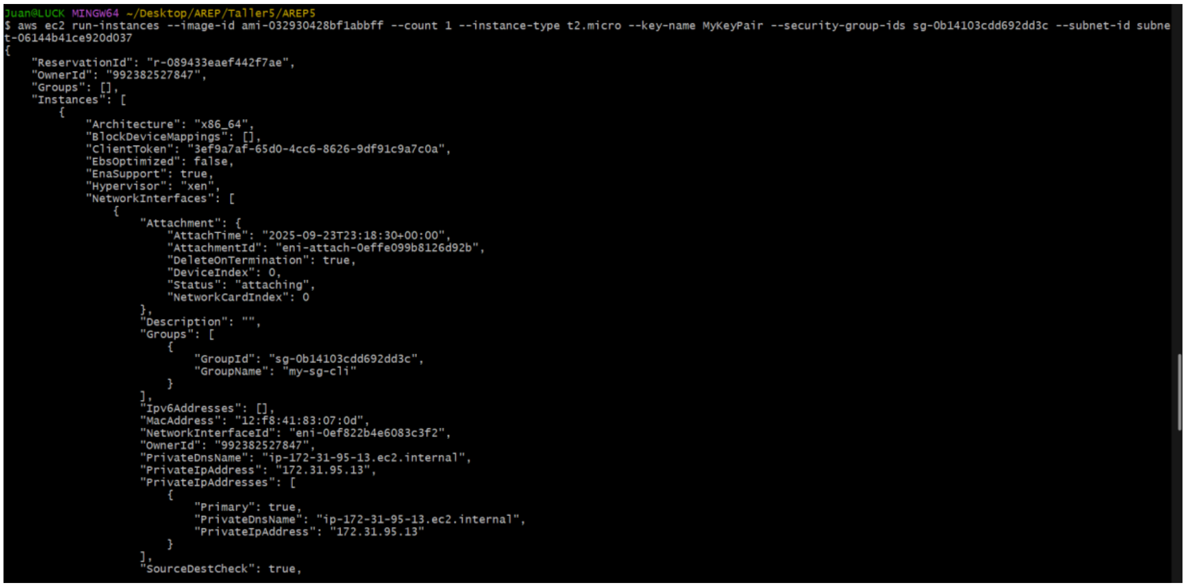
Mapa de recursos Información

Show all details

VPC
Su red virtual de AWS
vpc-0ed09d944025182b9

Subnets (6)
Subredes dentro de esta VPC
us-east-1a
subnet-06144b41ce920d037

Route tables (1)
Dirigir el tráfico de red a los recursos
rtb-00df895bb240d3e3e

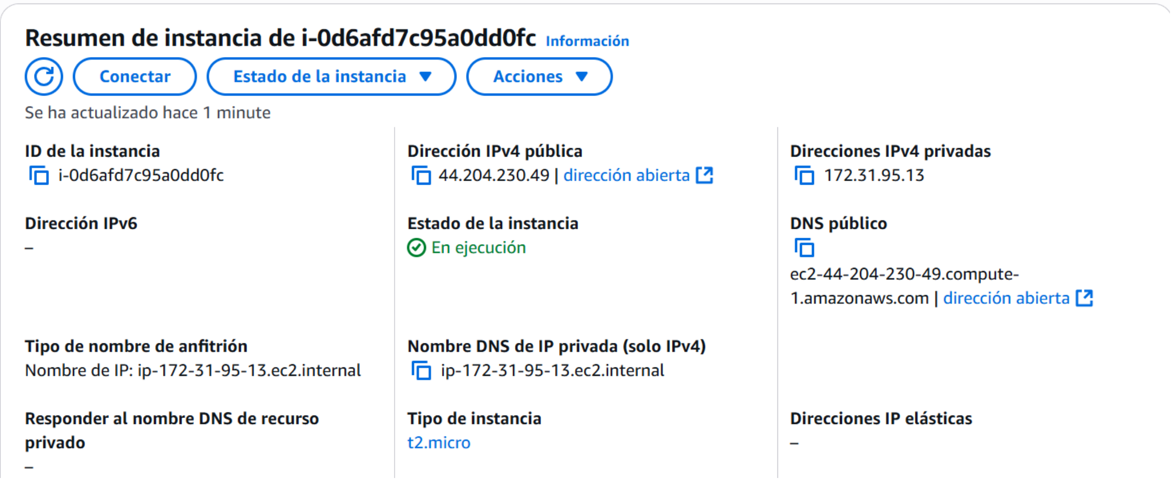


```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 run-instances --image-id ami-032930428bf1abbff --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-0b14103cdd692dd3c --subnet-id subne
t-06144b41ce920d037
{
    "ReservationId": "r-089433eaef442f7ae",
    "OwnerId": "992382527847",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "3ef9a7af-65d0-4cc6-8626-9df91c9a7c0a",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-09-23T23:18:30+00:00",
                        "AttachmentId": "eni-attach-0effe099b8126d92b",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-0b14103cdd692dd3c",
                            "GroupName": "my-sg-cli"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "12:f8:41:83:07:0d",
                    "NetworkInterfaceId": "eni-0ef822b4e6083c3f2",
                    "OwnerId": "992382527847",
                    "PrivateDnsName": "ip-172-31-95-13.ec2.internal",
                    "PrivateIpAddress": "172.31.95.13",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-172-31-95-13.ec2.internal",
                            "PrivateIpAddress": "172.31.95.13"
                        }
                    ],
                    "SourceDestCheck": true,
```

Step 4: Connect to the Instance

1. To connect to the instance, we need to obtain the public DNS, which can be found in the **instance section**. We then access the instance we are creating

(the one that appears without a name), and there we find the public DNS.



Resumen de instancia de i-0d6afd7c95a0dd0fc Información

Se ha actualizado hace 1 minute

ID de la instancia
i-0d6afd7c95a0dd0fc

Dirección IPv4 pública
44.204.230.49 | dirección abierta

Direcciones IPv4 privadas
172.31.95.13

Dirección IPv6
–

Estado de la instancia
En ejecución

DNS público
ec2-44-204-230-49.compute-1.amazonaws.com | dirección abierta

Tipo de nombre de anfitrión
Nombre de IP: ip-172-31-95-13.ec2.internal

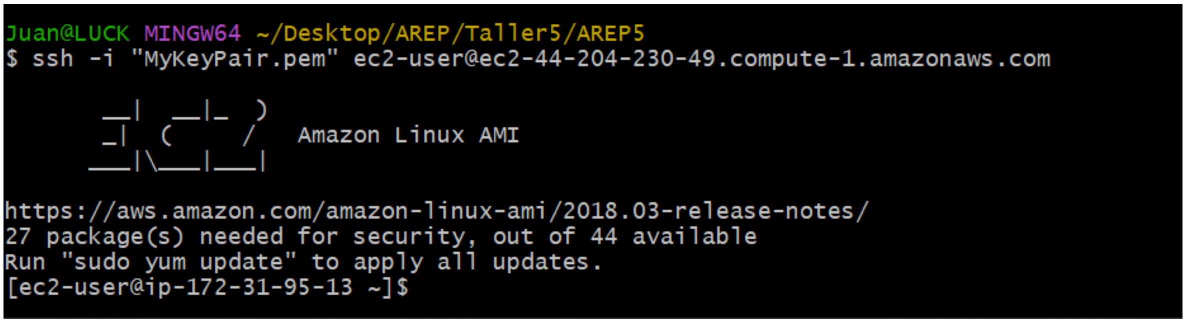
Nombre DNS de IP privada (solo IPv4)
ip-172-31-95-13.ec2.internal

Responder al nombre DNS de recurso privado
–

Tipo de instancia
t2.micro

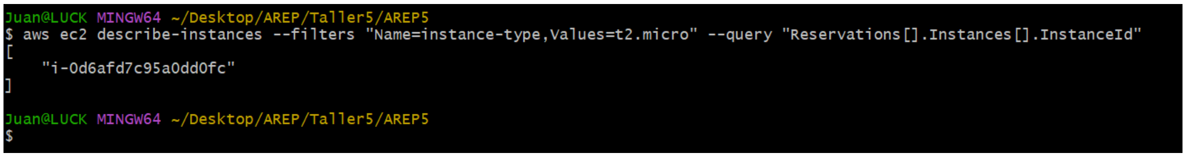
Direcciones IP elásticas
–

2. With the public DNS, we can now access the instance’s CLI.



```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ ssh -i "MyKeyPair.pem" ec2-user@ec2-44-204-230-49.compute-1.amazonaws.com

       __|  __|_  )
       _|  (     /   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-172-31-95-13 ~]$
```

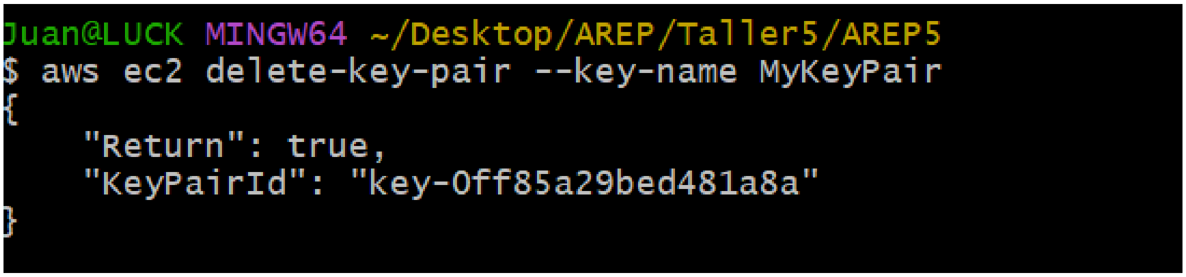
Step 5: List Your Instances



```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query "Reservations[].Instances[].InstanceId"
[
    "i-0d6afd7c95a0dd0fc"
]

Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$
```

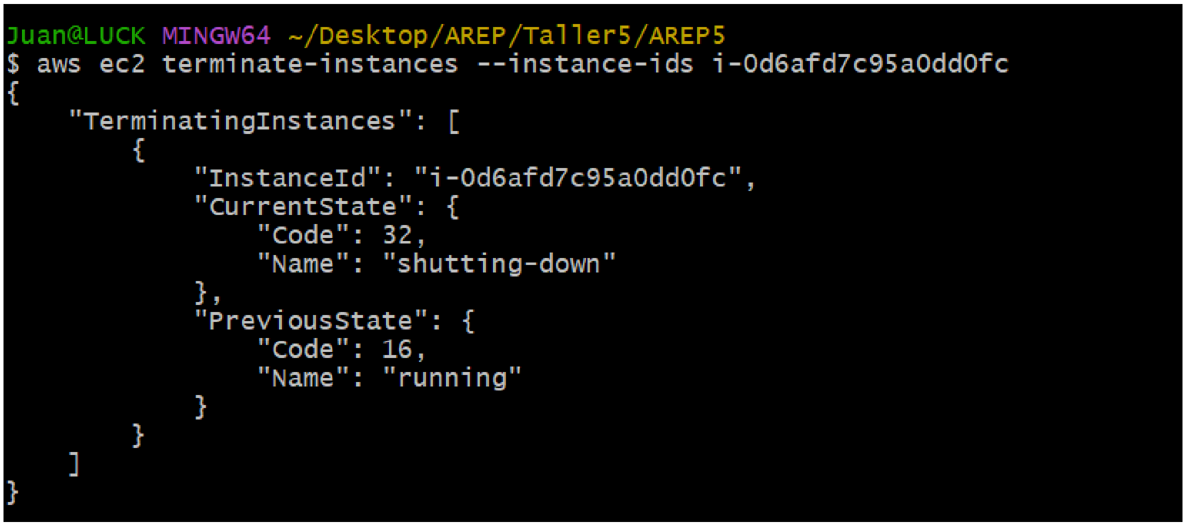
Step 6: Clean Up

1. Delete the key pair:

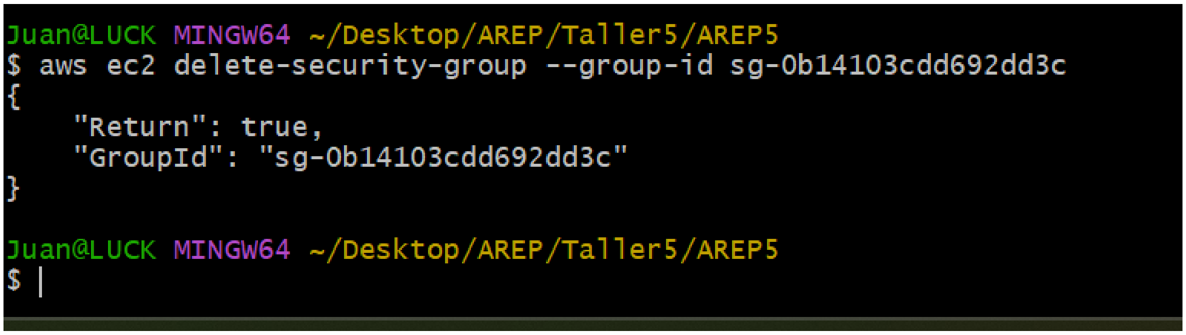


```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 delete-key-pair --key-name MyKeyPair
{
    "Return": true,
    "KeyPairId": "key-0ff85a29bed481a8a"
}
```

2. Delete the security group and Terminate the instance



```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 terminate-instances --instance-ids i-0d6afd7c95a0dd0fc
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-0d6afd7c95a0dd0fc",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```



```
Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ aws ec2 delete-security-group --group-id sg-0b14103cdd692dd3c
{
    "Return": true,
    "GroupId": "sg-0b14103cdd692dd3c"
}

Juan@LUCK MINGW64 ~/Desktop/AREP/Taller5/AREP5
$ |
```