LAB # 09:

CLOUD COMPUTING:

FROM:

ALINA IMAN

2023-BSE-005

SECTION A

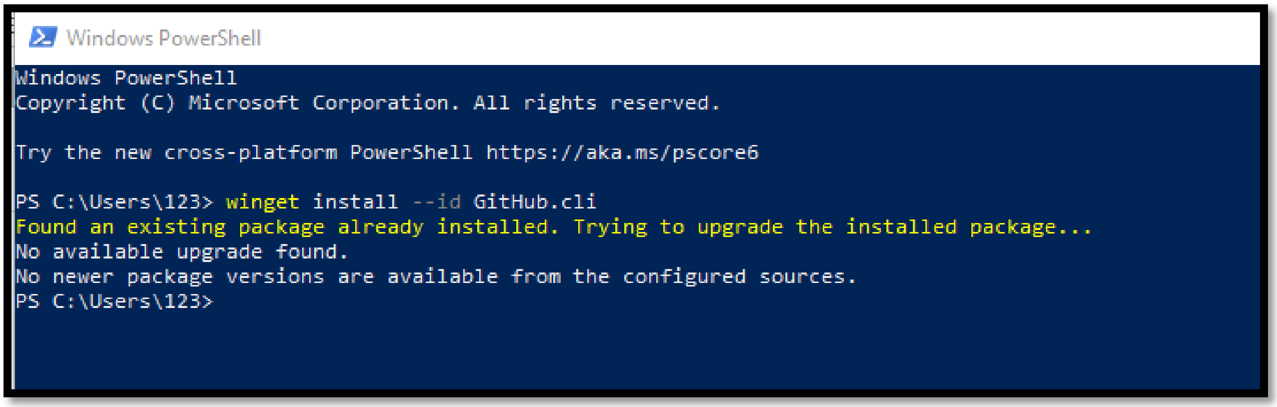
TO:

SIR SHOAIB

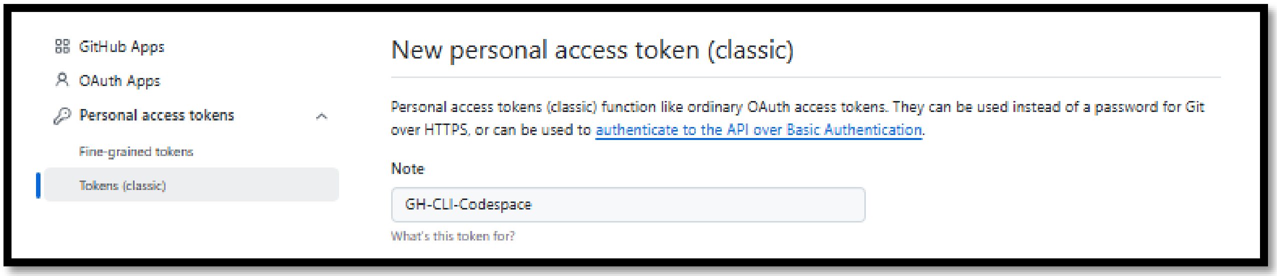
TITLE:

CODESPACES + AWS: GH CLI (CODESPACES), AWS CLI, EC2, IAM, SECURITY GROUPS, FILTERS & QUERIES

Task 1 — GitHub CLI, Codespace setup and authentication



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\123> winget install --id GitHub.cli
Found an existing package already installed. Trying to upgrade the installed package...
No available upgrade found.
No newer package versions are available from the configured sources.
PS C:\Users\123>
```



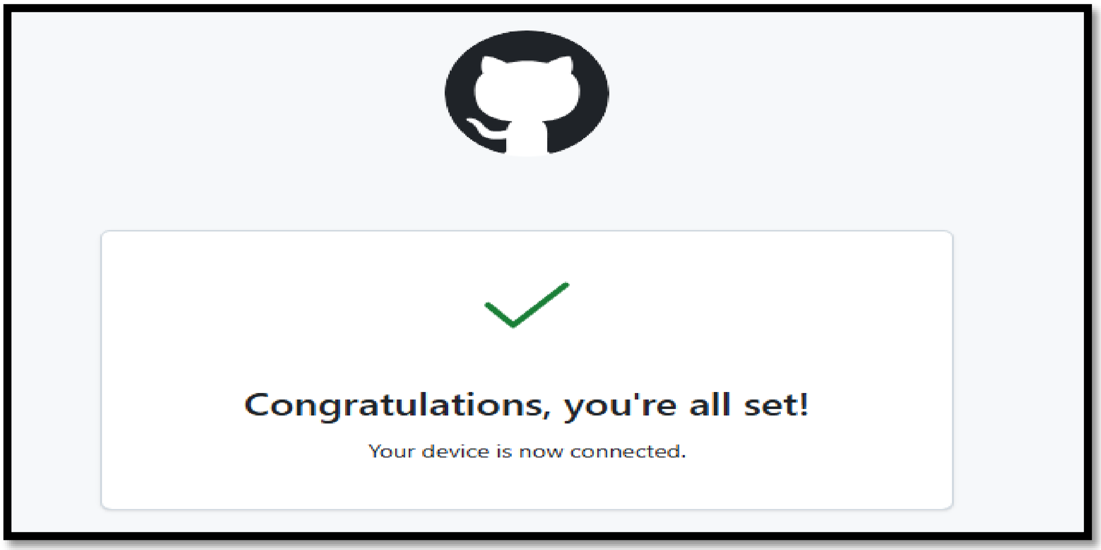
New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

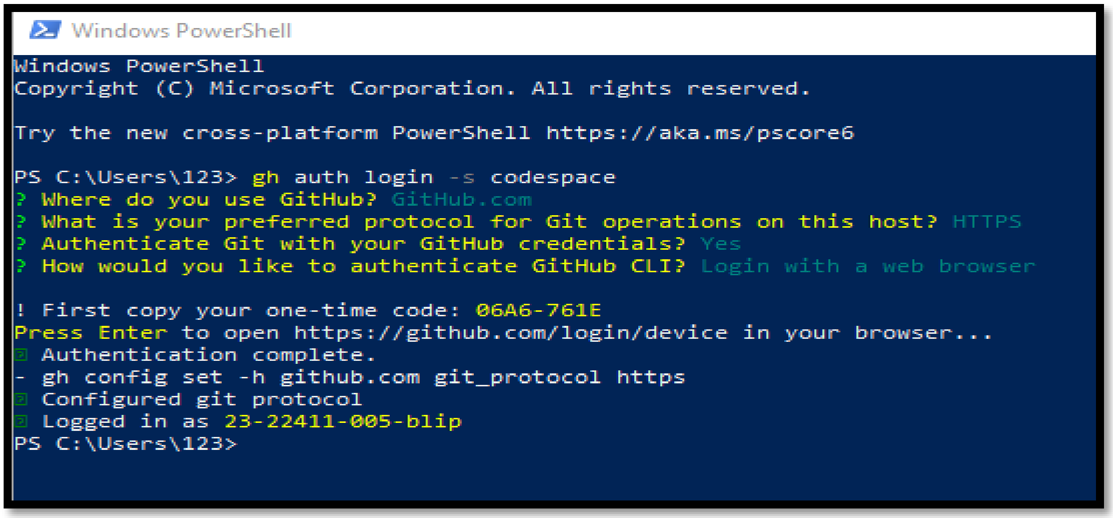
GH-CLI-Codespace

What's this token for?

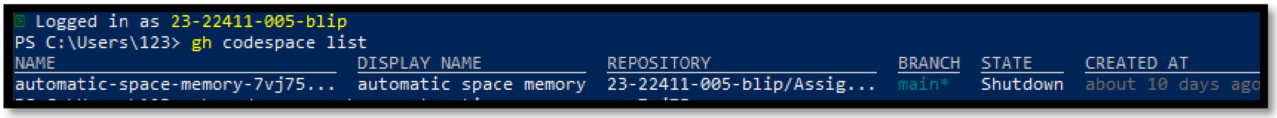


Congratulations, you're all set!

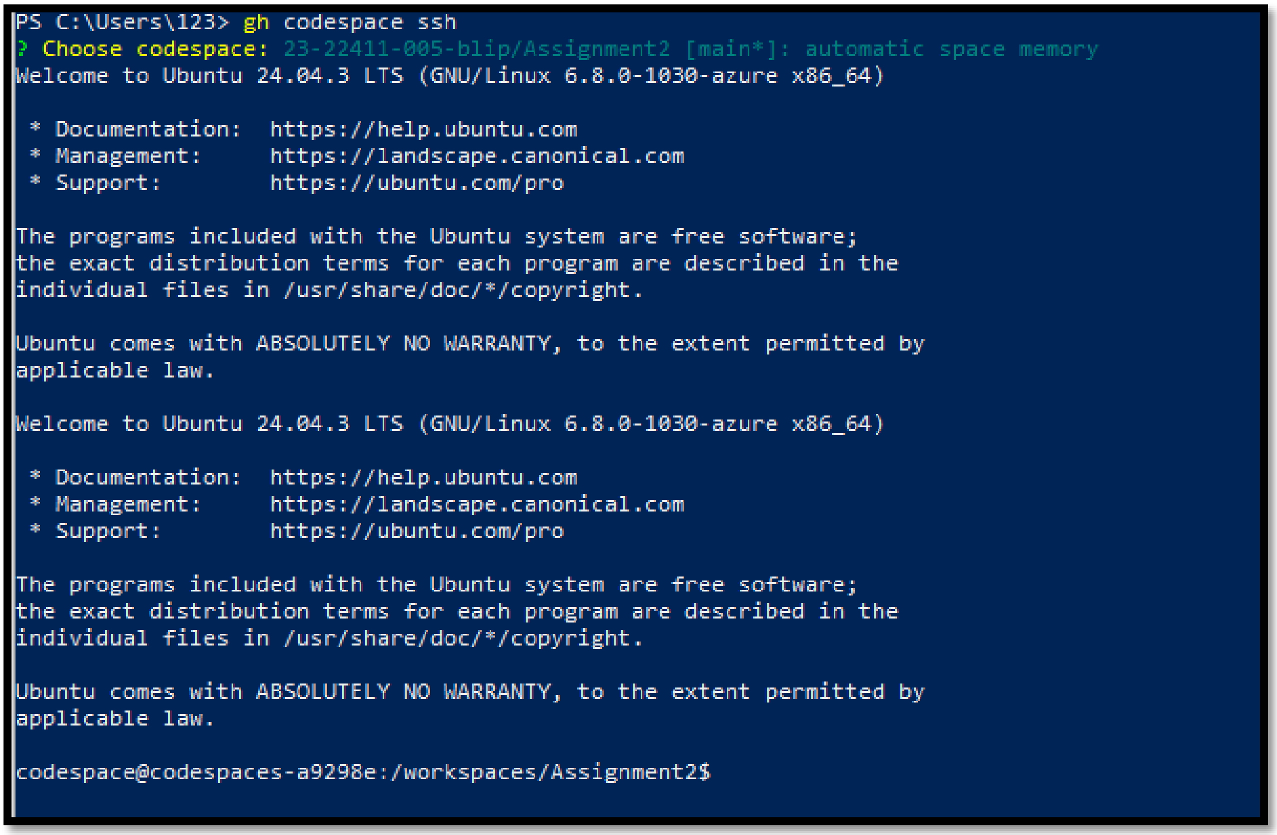
Your device is now connected.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\123> gh auth login -s codespace
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? HTTPS
? Authenticate Git with your GitHub credentials? Yes
? How would you like to authenticate GitHub CLI? Login with a web browser

! First copy your one-time code: 06A6-761E
Press Enter to open https://github.com/login/device in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as 23-22411-005-blip
PS C:\Users\123>
```

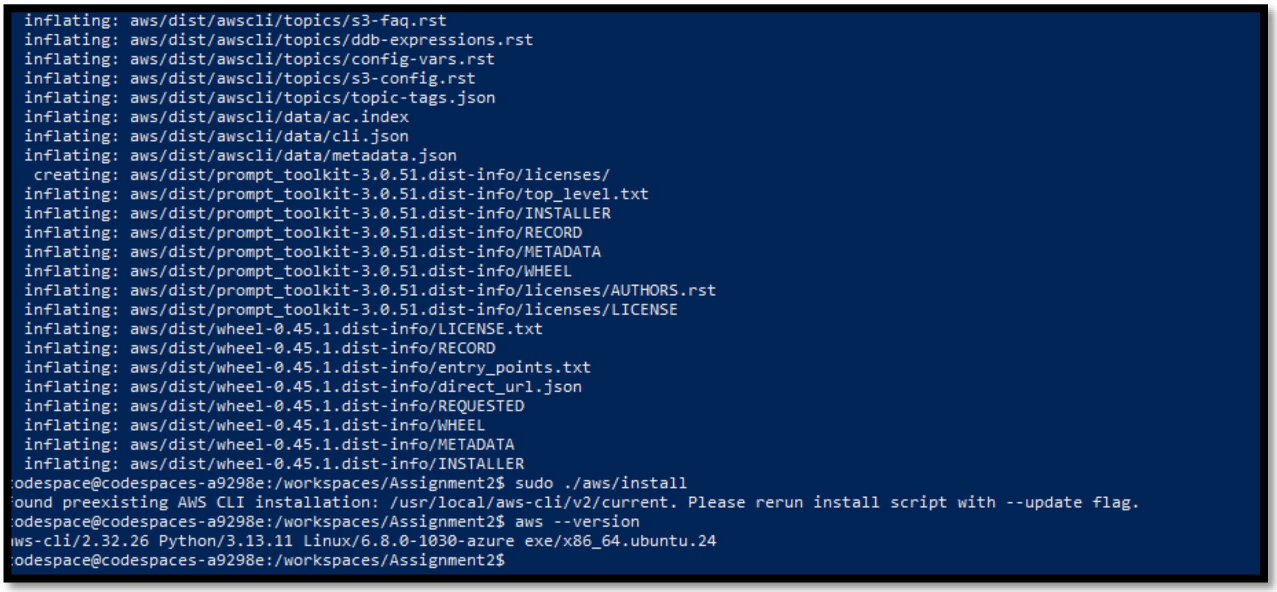


```
✓ Logged in as 23-22411-005-blip
PS C:\Users\123> gh codespace list
NAME                          DISPLAY NAME            REPOSITORY                  BRANCH  STATE     CREATED AT
automatic-space-memory-7vj75...  automatic space memory  23-22411-005-blip/Assig...  main*   Shutdown  about 10 days ago
```

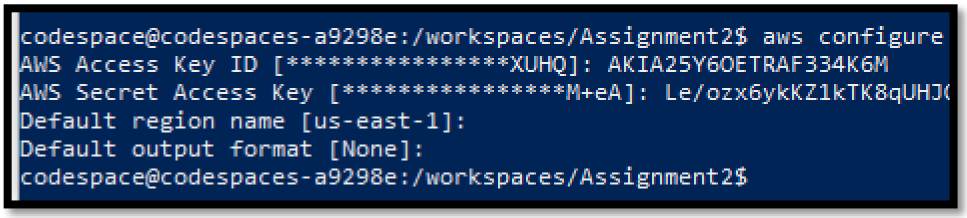


```
PS C:\Users\123> gh codespace ssh
? Choose codespace: 23-22411-005-blip/Assignment2 [main*]: automatic space memory
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

codespace@codespaces-a9298e:/workspaces/Assignment2$
```

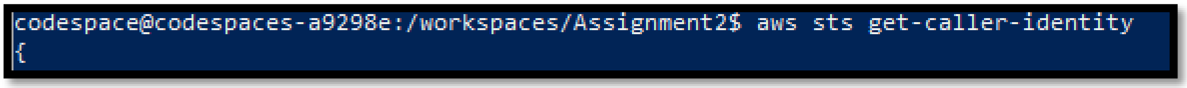
Task 2 — Install AWS CLI inside the Codespace and configure it



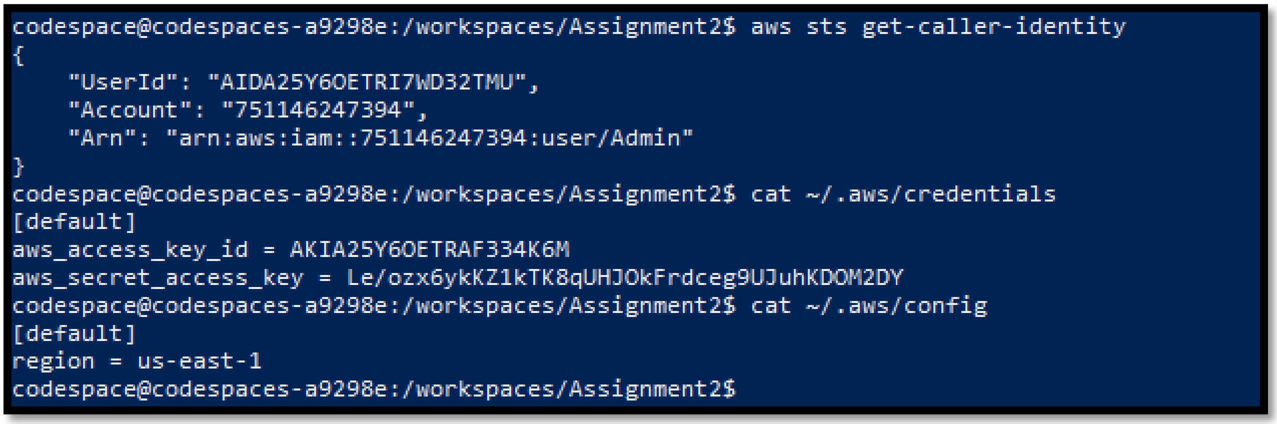
```
 inflating: aws/dist/awscli/topics/s3-faq.rst
 inflating: aws/dist/awscli/topics/ddb-expressions.rst
 inflating: aws/dist/awscli/topics/config-vars.rst
 inflating: aws/dist/awscli/topics/s3-config.rst
 inflating: aws/dist/awscli/topics/topic-tags.json
 inflating: aws/dist/awscli/data/ac.index
 inflating: aws/dist/awscli/data/cli.json
 inflating: aws/dist/awscli/data/metadata.json
  creating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/
 inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/top_level.txt
 inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/INSTALLER
 inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/RECORD
 inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/METADATA
 inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/WHEEL
 inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/AUTHORS.rst
 inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/LICENSE
 inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
 inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
 inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
 inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
 inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
 inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
 inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
 inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
odespace@codespaces-a9298e:/workspaces/Assignment2$ sudo ./aws/install
ound preexisting AWS CLI installation: /usr/local/aws-cli/v2/current. Please rerun install script with --update flag.
odespace@codespaces-a9298e:/workspaces/Assignment2$ aws --version
ws-cli/2.32.26 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
odespace@codespaces-a9298e:/workspaces/Assignment2$
```



```
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws configure
AWS Access Key ID [****************XUHQ]: AKIA25Y6OETRAF334K6M
AWS Secret Access Key [****************M+eA]: Le/ozx6ykKZ1kTK8qUHJ
Default region name [us-east-1]:
Default output format [None]:
codespace@codespaces-a9298e:/workspaces/Assignment2$
```

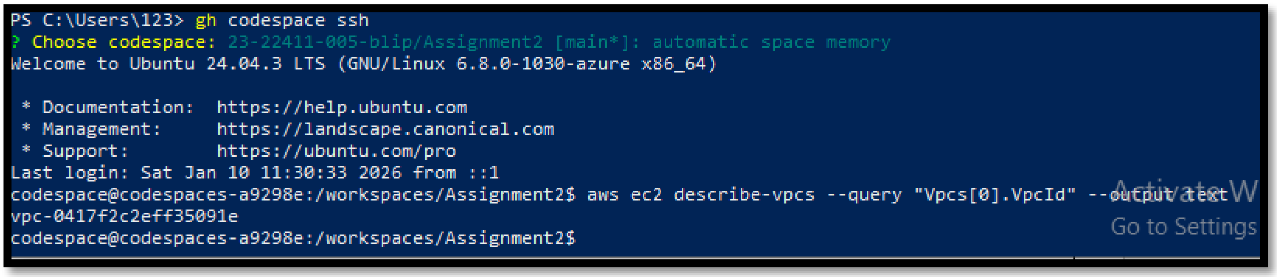


```
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws sts get-caller-identity
{
```

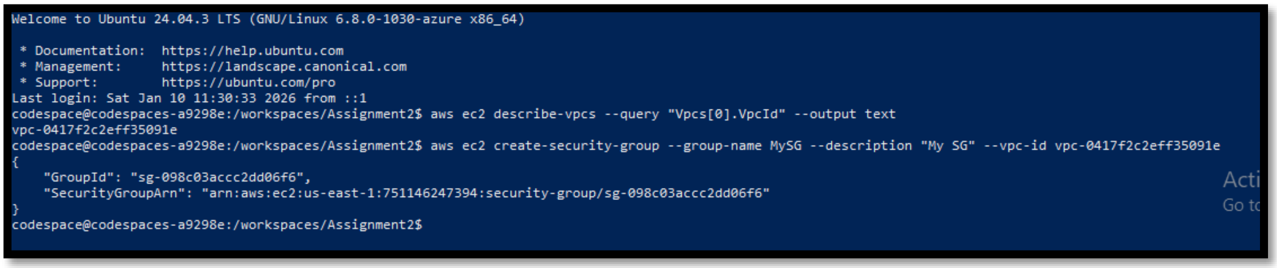


```
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws sts get-caller-identity
{
    "UserId": "AIDA25Y6OETRI7WD32TMU",
    "Account": "751146247394",
    "Arn": "arn:aws:iam::751146247394:user/Admin"
}
codespace@codespaces-a9298e:/workspaces/Assignment2$ cat ~/.aws/credentials
[default]
aws_access_key_id = AKIA25Y6OETRAF334K6M
aws_secret_access_key = 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
codespace@codespaces-a9298e:/workspaces/Assignment2$ cat ~/.aws/config
[default]
region = us-east-1
codespace@codespaces-a9298e:/workspaces/Assignment2$
```

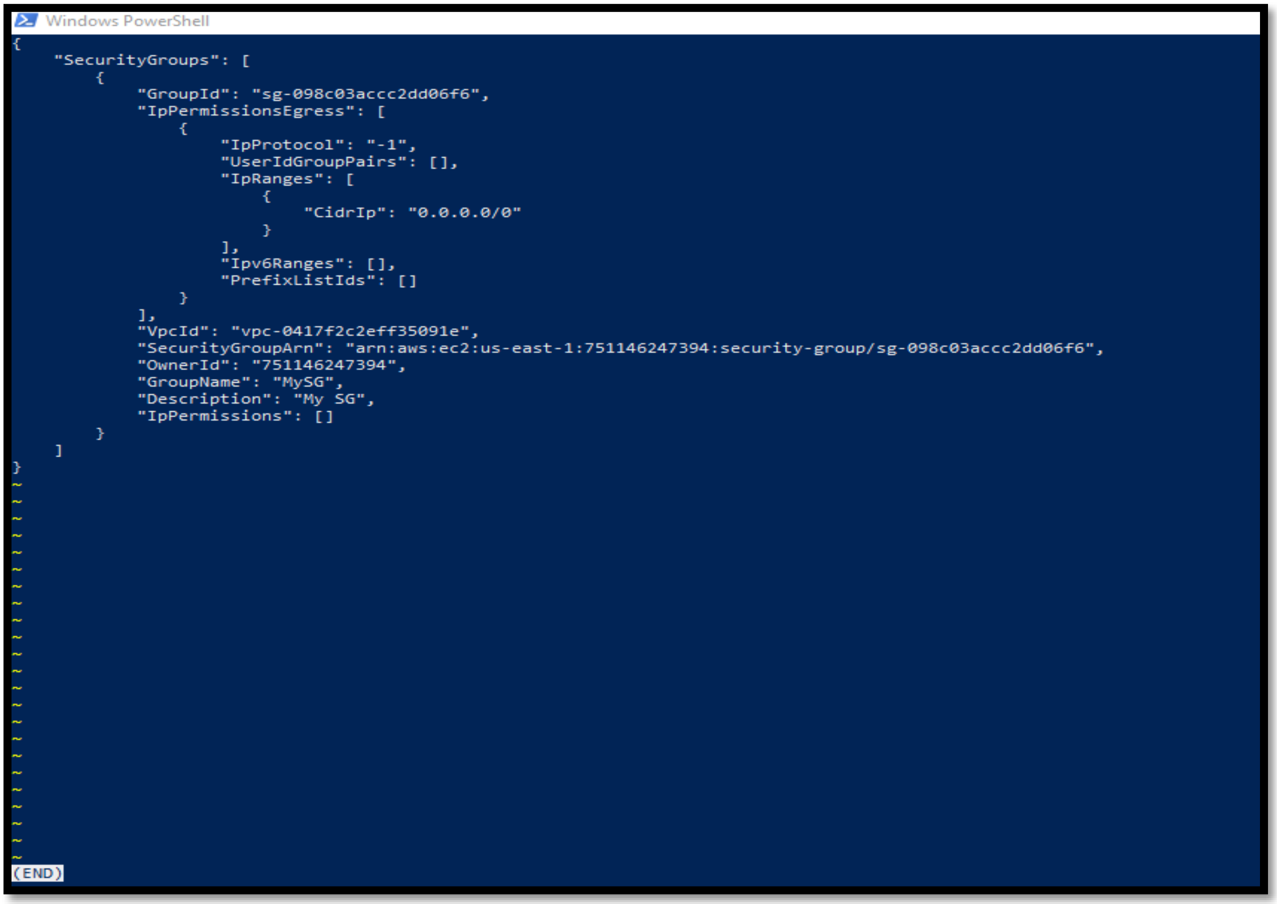
Task 3 — Create security group and add ingress rules using Codespace IP



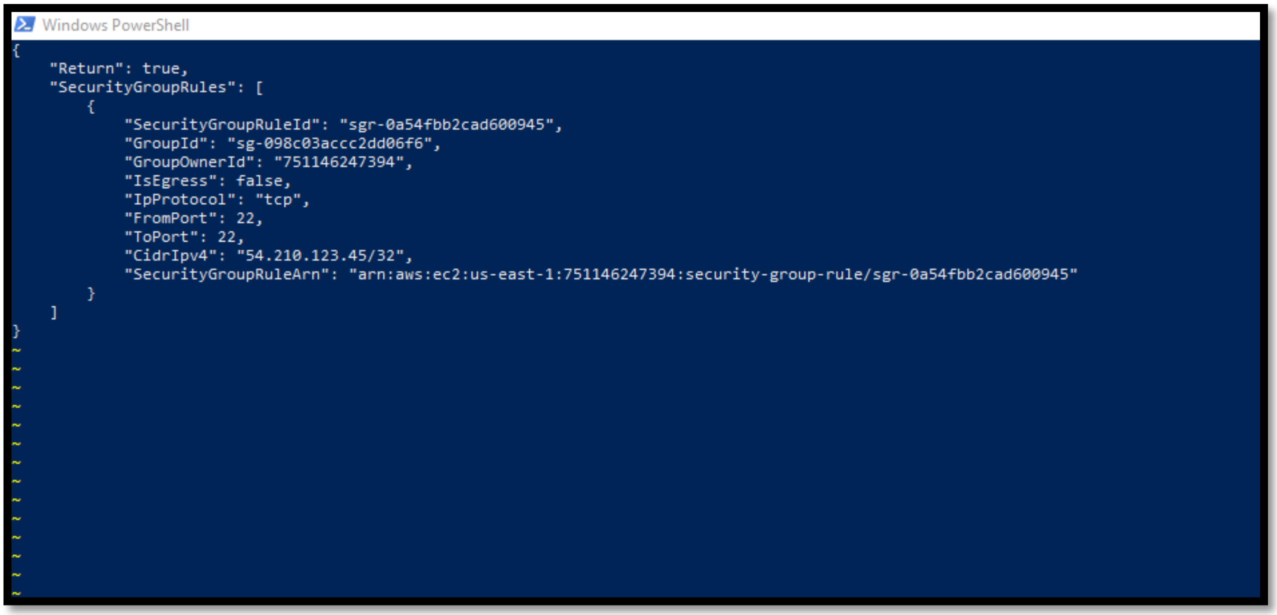
```
PS C:\Users\123> gh codespace ssh
? Choose codespace: 23-22411-005-blip/Assignment2 [main*]: automatic space memory
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro
Last login: Sat Jan 10 11:30:33 2026 from ::1
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws ec2 describe-vpcs --query "Vpcs[0].VpcId" --output text
vpc-0417f2c2eff35091e
codespace@codespaces-a9298e:/workspaces/Assignment2$
```



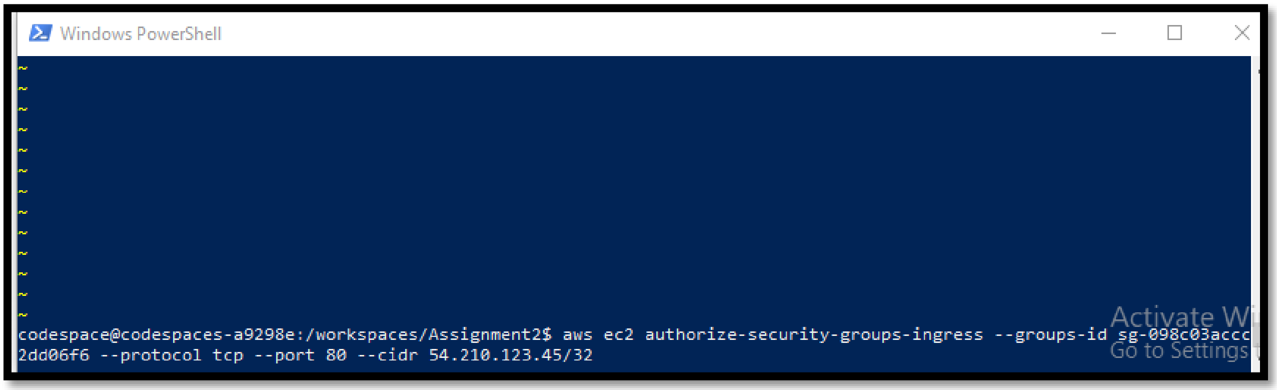
```
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro
Last login: Sat Jan 10 11:30:33 2026 from ::1
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws ec2 describe-vpcs --query "Vpcs[0].VpcId" --output text
vpc-0417f2c2eff35091e
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws ec2 create-security-group --group-name MySG --description "My SG" --vpc-id vpc-0417f2c2eff35091e
{
    "GroupId": "sg-098c03accc2dd06f6",
    "SecurityGroupArn": "arn:aws:ec2:us-east-1:751146247394:security-group/sg-098c03accc2dd06f6"
}
codespace@codespaces-a9298e:/workspaces/Assignment2$
```



```
{
    "SecurityGroups": [
        {
            "GroupId": "sg-098c03accc2dd06f6",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0417f2c2eff35091e",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:751146247394:security-group/sg-098c03accc2dd06f6",
            "OwnerId": "751146247394",
            "GroupName": "MySG",
            "Description": "My SG",
            "IpPermissions": []
        }
    ]
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
(END)
```

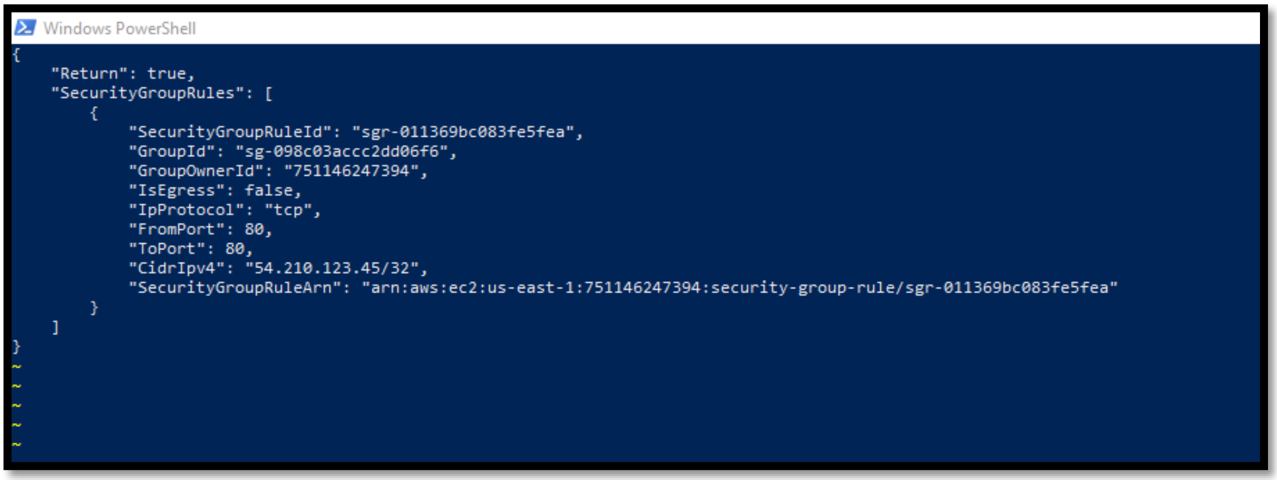


```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0a54fbb2cad600945",
            "GroupId": "sg-098c03accc2dd06f6",
            "GroupOwnerId": "751146247394",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "54.210.123.45/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:751146247394:security-group-rule/sgr-0a54fbb2cad600945"
        }
    ]
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

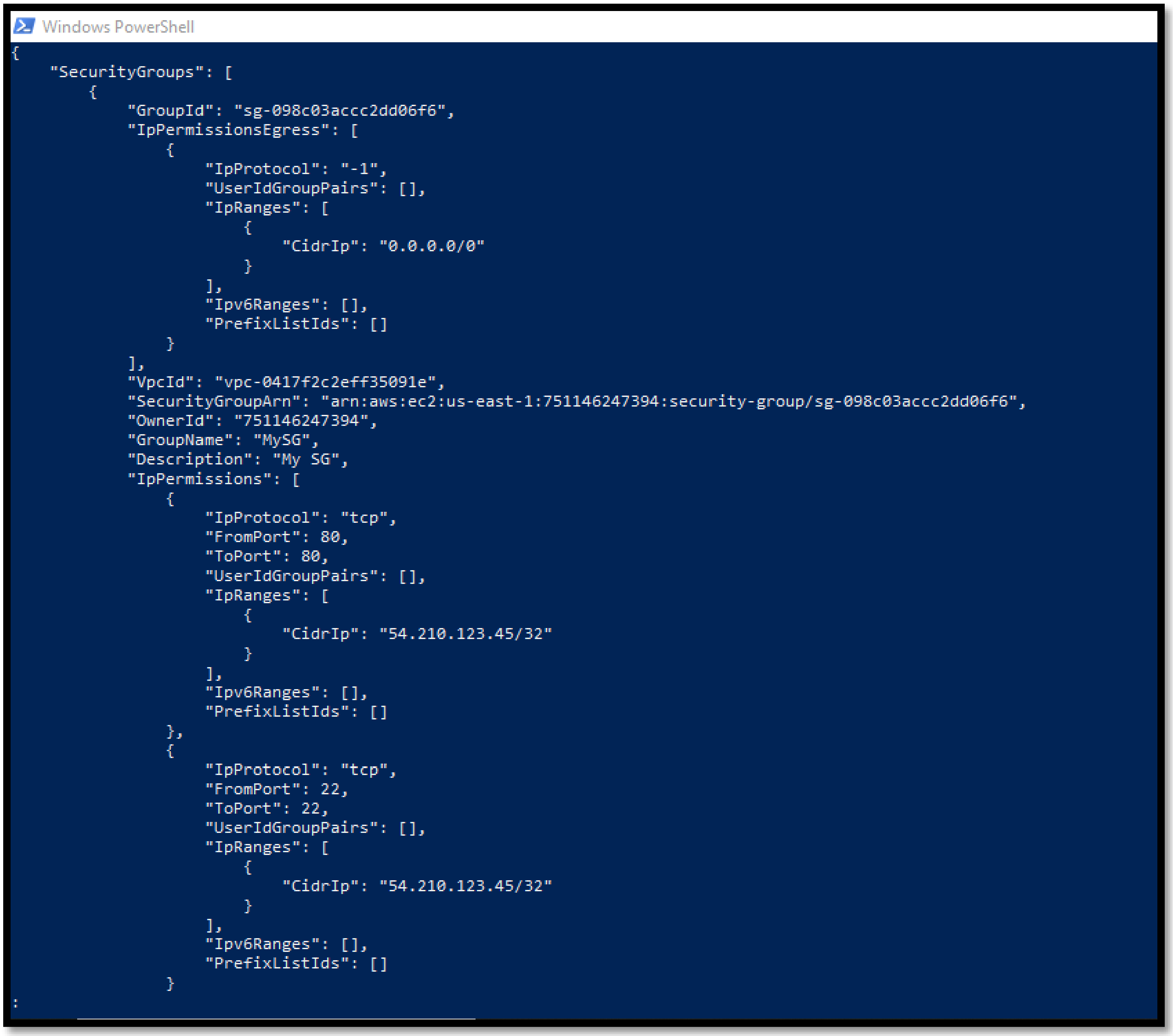


```
~
~
~
~
~
~
~
~
~
~
~
~
~
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws ec2 authorize-security-groups-ingress --groups-id sg-098c03accc
2dd06f6 --protocol tcp --port 80 --cidr 54.210.123.45/32
```

Activate Wi
Go to Settings

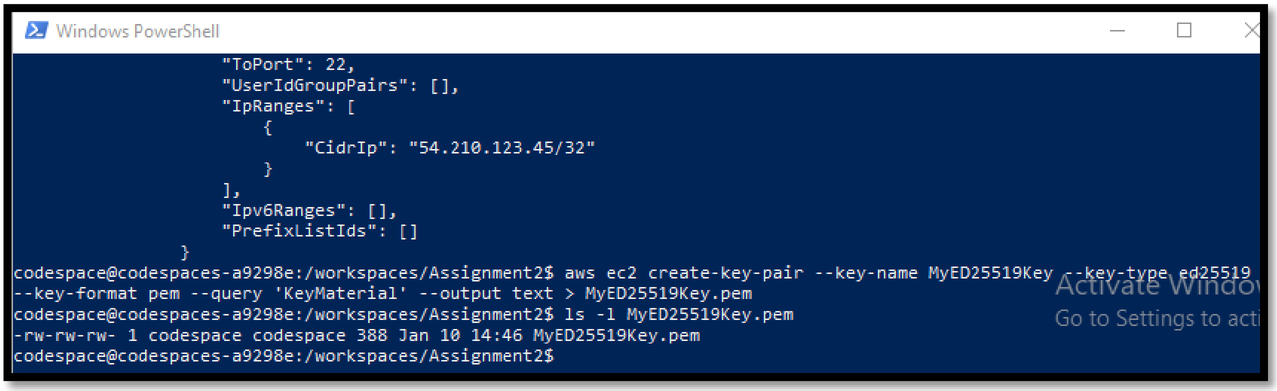


```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-011369bc083fe5fea",
            "GroupId": "sg-098c03accc2dd06f6",
            "GroupOwnerId": "751146247394",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "54.210.123.45/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:751146247394:security-group-rule/sgr-011369bc083fe5fea"
        }
    ]
}
~
~
~
~
~
```



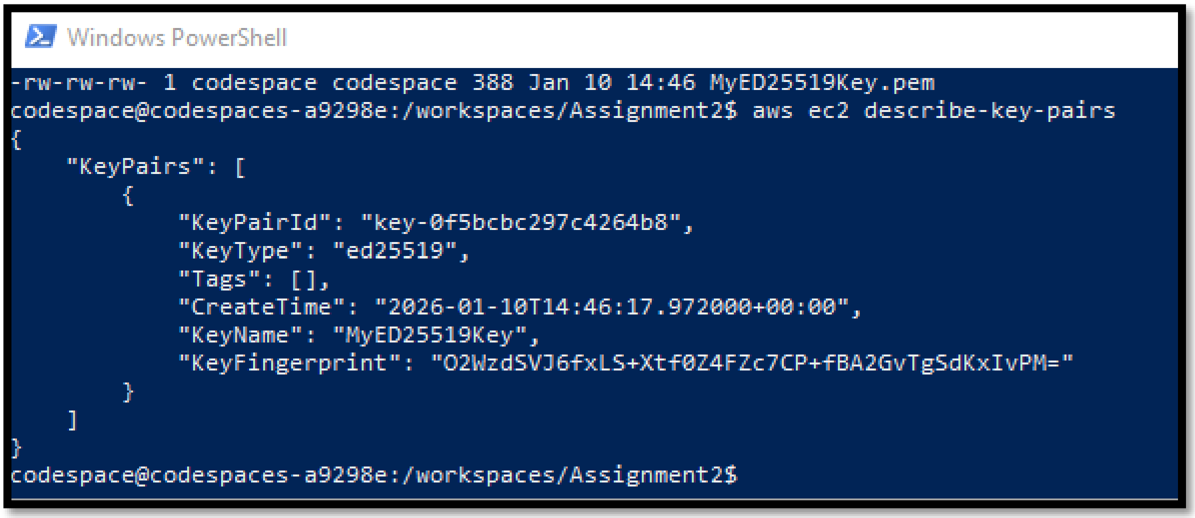
```
{
    "SecurityGroups": [
        {
            "GroupId": "sg-098c03accc2dd06f6",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0417f2c2eff35091e",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:751146247394:security-group/sg-098c03accc2dd06f6",
            "OwnerId": "751146247394",
            "GroupName": "MySG",
            "Description": "My SG",
            "IpPermissions": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "54.210.123.45/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "54.210.123.45/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
:
```

Task 4 — Create a key pair, describe key pairs, and launch EC2 instance

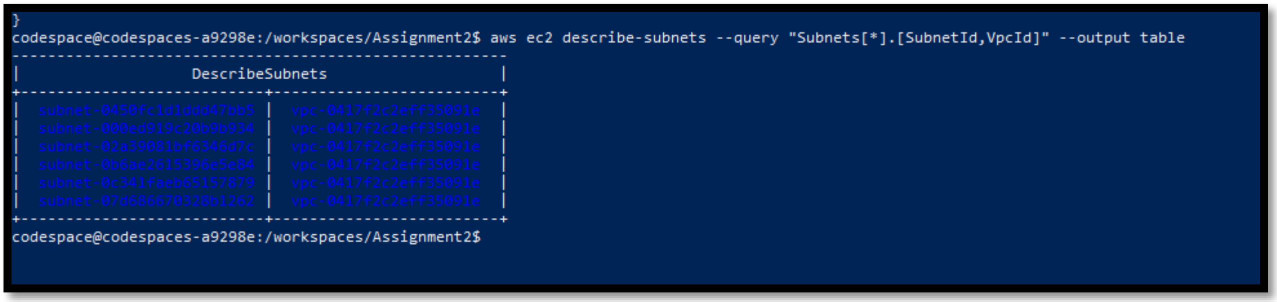


```
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "54.210.123.45/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws ec2 create-key-pair --key-name MyED25519Key --key-type ed25519
--key-format pem --query 'KeyMaterial' --output text > MyED25519Key.pem
codespace@codespaces-a9298e:/workspaces/Assignment2$ ls -l MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Jan 10 14:46 MyED25519Key.pem
codespace@codespaces-a9298e:/workspaces/Assignment2$
```

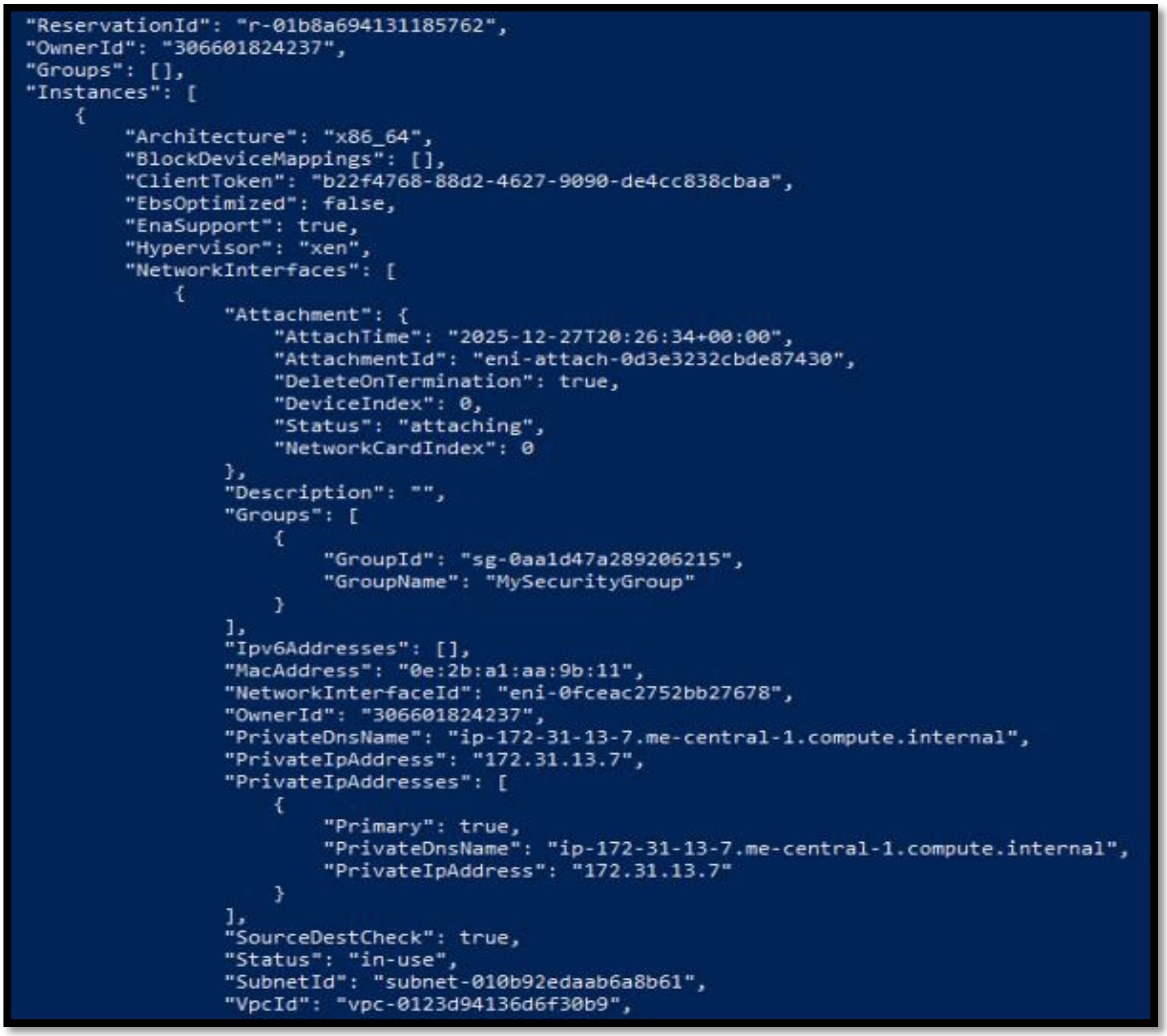
Activate Windo
Go to Settings to acti



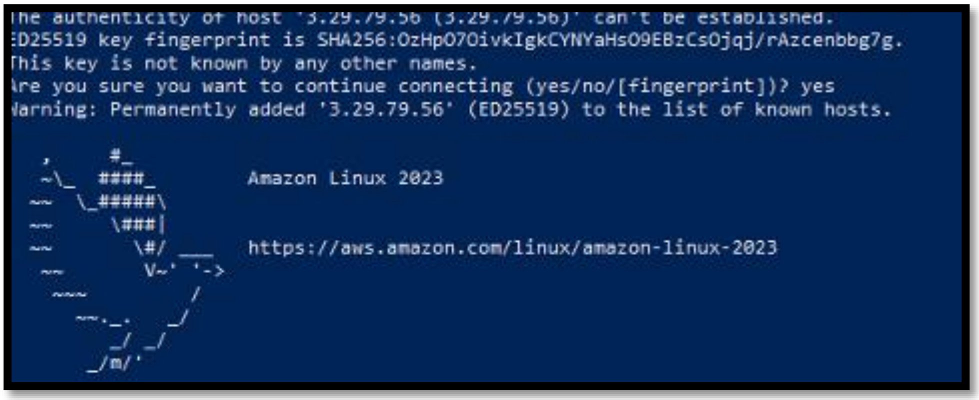
```
-rw-rw-rw- 1 codespace codespace 388 Jan 10 14:46 MyED25519Key.pem
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws ec2 describe-key-pairs
{
    "KeyPairs": [
        {
            "KeyPairId": "key-0f5bcbc297c4264b8",
            "KeyType": "ed25519",
            "Tags": [],
            "CreateTime": "2026-01-10T14:46:17.972000+00:00",
            "KeyName": "MyED25519Key",
            "KeyFingerprint": "O2WzdSVJ6fxLS+Xtf0Z4FZc7CP+fBA2GvTgSdKxIvPM="
        }
    ]
}
codespace@codespaces-a9298e:/workspaces/Assignment2$
```



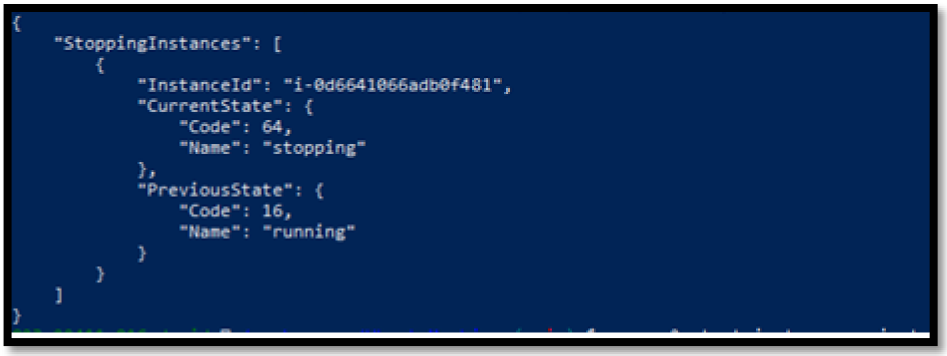
```
}
codespace@codespaces-a9298e:/workspaces/Assignment2$ aws ec2 describe-subnets --query "Subnets[*].[SubnetId,VpcId]" --output table
-------------------------------------------------------
|                   DescribeSubnets                   |
+---------------------------+-------------------------+
|  subnet-0450fc1d1ddd47bb5 |  vpc-0417f2c2eff35091e  |
|  subnet-000ed919c20b9b934 |  vpc-0417f2c2eff35091e  |
|  subnet-02a39081bf6346d7c |  vpc-0417f2c2eff35091e  |
|  subnet-0b6ae2615396e5e84 |  vpc-0417f2c2eff35091e  |
|  subnet-0c341faeb65157879 |  vpc-0417f2c2eff35091e  |
|  subnet-07d686670328b1262 |  vpc-0417f2c2eff35091e  |
+---------------------------+-------------------------+
codespace@codespaces-a9298e:/workspaces/Assignment2$
```



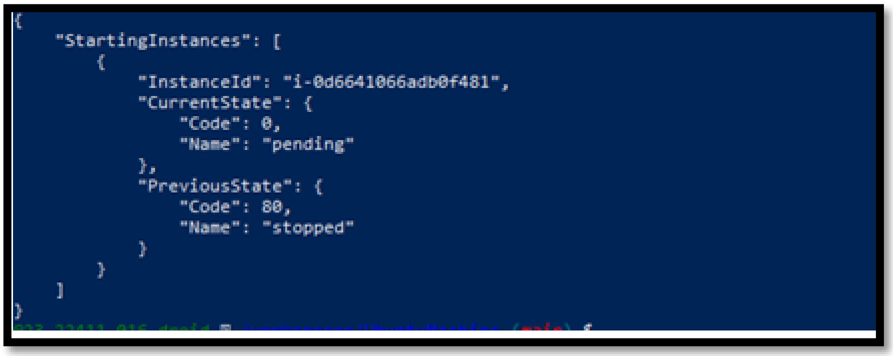
```
"ReservationId": "r-01b8a694131185762",
"OwnerId": "306601824237",
"Groups": [],
"Instances": [
    {
        "Architecture": "x86_64",
        "BlockDeviceMappings": [],
        "ClientToken": "b22f4768-88d2-4627-9090-de4cc838cbaa",
        "EbsOptimized": false,
        "EnaSupport": true,
        "Hypervisor": "xen",
        "NetworkInterfaces": [
            {
                "Attachment": {
                    "AttachTime": "2025-12-27T20:26:34+00:00",
                    "AttachmentId": "eni-attach-0d3e3232cbde87430",
                    "DeleteOnTermination": true,
                    "DeviceIndex": 0,
                    "Status": "attaching",
                    "NetworkCardIndex": 0
                },
                "Description": "",
                "Groups": [
                    {
                        "GroupId": "sg-0aa1d47a289206215",
                        "GroupName": "MySecurityGroup"
                    }
                ],
                "Ipv6Addresses": [],
                "MacAddress": "0e:2b:a1:aa:9b:11",
                "NetworkInterfaceId": "eni-0fceac2752bb27678",
                "OwnerId": "306601824237",
                "PrivateDnsName": "ip-172-31-13-7.me-central-1.compute.internal",
                "PrivateIpAddress": "172.31.13.7",
                "PrivateIpAddresses": [
                    {
                        "Primary": true,
                        "PrivateDnsName": "ip-172-31-13-7.me-central-1.compute.internal",
                        "PrivateIpAddress": "172.31.13.7"
                    }
                ],
                "SourceDestCheck": true,
                "Status": "in-use",
                "SubnetId": "subnet-010b92edaab6a8b61",
                "VpcId": "vpc-0123d94136d6f30b9",
```



```
The authenticity of host '3.29.79.56 (3.29.79.56)' can't be established.
ED25519 key fingerprint is SHA256:OzHpO7OivkIgkCYNYaHsO9EBzCsOjqj/rAzcenbbg7g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.29.79.56' (ED25519) to the list of known hosts.

   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
```

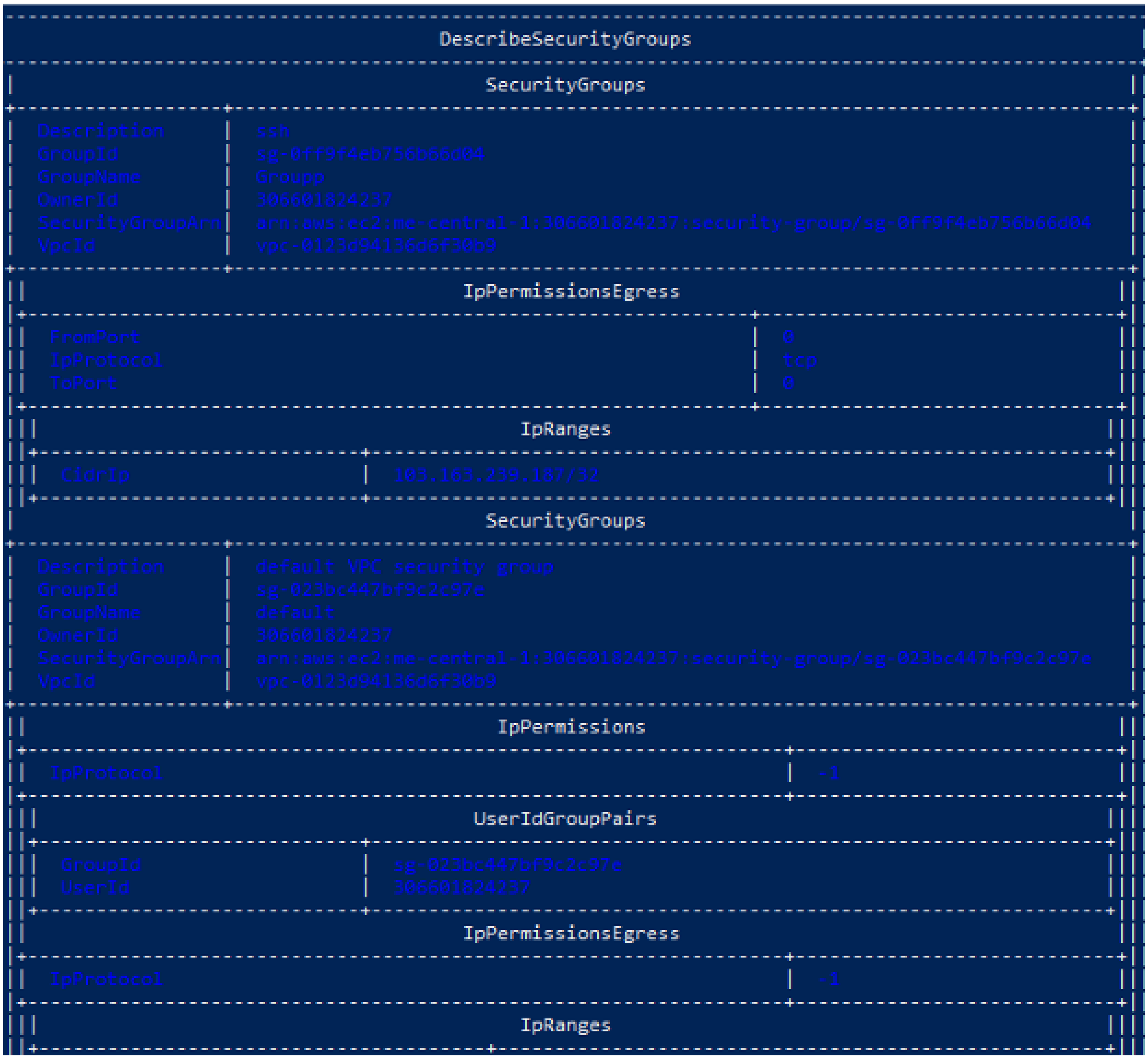


```
{
    "StoppingInstances": [
        {
            "InstanceId": "i-0d6641066adb0f481",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

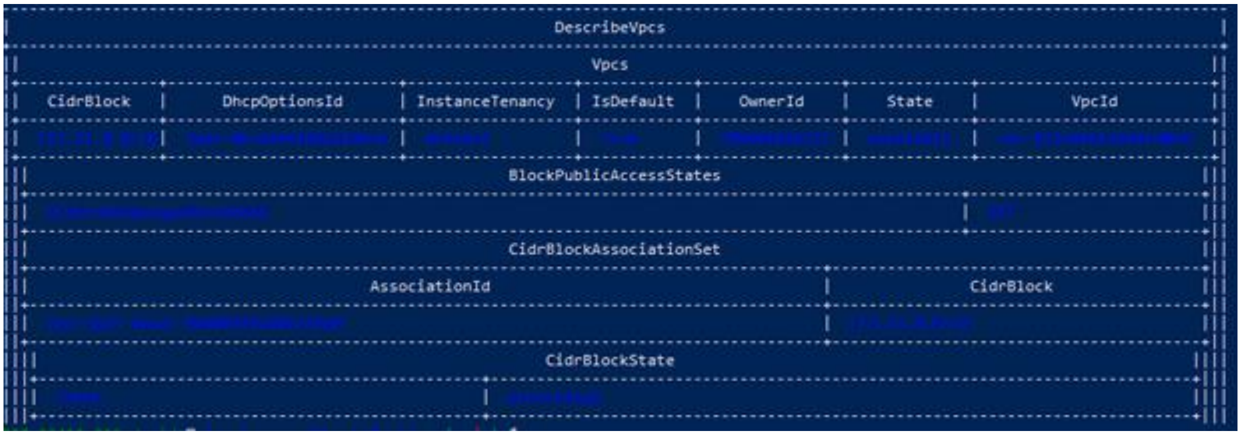


```
{
    "StartingInstances": [
        {
            "InstanceId": "i-0d6641066adb0f481",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

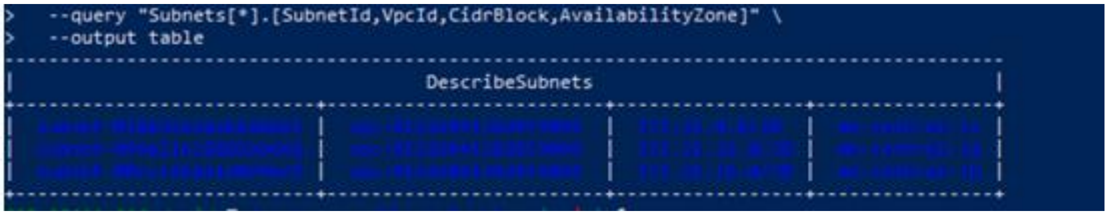
Task 5 — Understand AWS describe-* commands



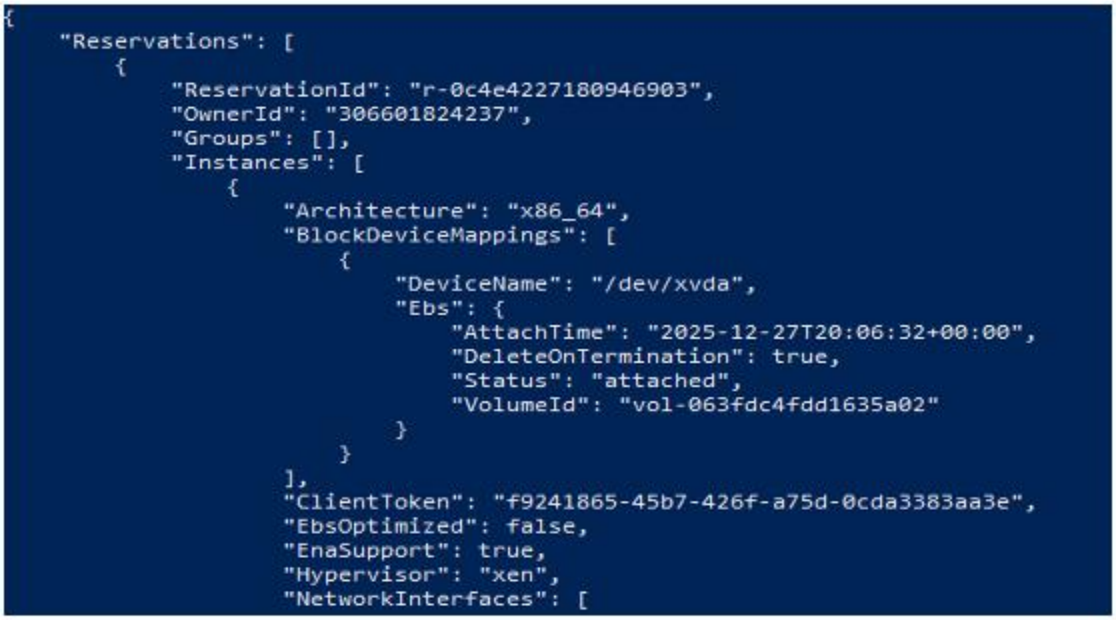
```
---------------------------------------------------------------------------------------------------
|                                      DescribeSecurityGroups                                       |
---------------------------------------------------------------------------------------------------
|                                          SecurityGroups                                         |
+-------------------+-----------------------------------------------------------------------------+
|  Description      |  ssh                                                                        |
|  GroupId          |  sg-0ff9f4eb756b66d04                                                       |
|  GroupName        |  Groupp                                                                     |
|  OwnerId          |  306601824237                                                               |
|  SecurityGroupArn |  arn:aws:ec2:me-central-1:306601824237:security-group/sg-0ff9f4eb756b66d04  |
|  VpcId            |  vpc-0123d94136d6f30b9                                                      |
+-------------------+-----------------------------------------------------------------------------+
||                                      IpPermissionsEgress                                      ||
|+----------------------------------------------------------------------+------------------------+|
||  FromPort                                                            |  0                     ||
||  IpProtocol                                                          |  tcp                   ||
||  ToPort                                                              |  0                     ||
|+----------------------------------------------------------------------+------------------------+|
|||                                          IpRanges                                           |||
||+------------------------------+-------------------------------------------------------------+||
|||  CidrIp                      |  103.163.239.187/32                                         |||
||+------------------------------+-------------------------------------------------------------+||
|                                          SecurityGroups                                         |
+-------------------+-----------------------------------------------------------------------------+
|  Description      |  default VPC security group                                                 |
|  GroupId          |  sg-023bc447bf9c2c97e                                                       |
|  GroupName        |  default                                                                    |
|  OwnerId          |  306601824237                                                               |
|  SecurityGroupArn |  arn:aws:ec2:me-central-1:306601824237:security-group/sg-023bc447bf9c2c97e  |
|  VpcId            |  vpc-0123d94136d6f30b9                                                      |
+-------------------+-----------------------------------------------------------------------------+
||                                         IpPermissions                                         ||
|+-------------------------------------------------------------------------+---------------------+|
||  IpProtocol                                                             |  -1                 ||
|+-------------------------------------------------------------------------+---------------------+|
|||                                      UserIdGroupPairs                                       |||
||+------------------------------+-------------------------------------------------------------+||
|||  GroupId                     |  sg-023bc447bf9c2c97e                                       |||
|||  UserId                      |  306601824237                                               |||
||+------------------------------+-------------------------------------------------------------+||
||                                      IpPermissionsEgress                                      ||
|+-------------------------------------------------------------------------+---------------------+|
||  IpProtocol                                                             |  -1                 ||
|+-------------------------------------------------------------------------+---------------------+|
|||                                          IpRanges                                           |||
||+---------------------------------------------+----------------------------------------------+||
```



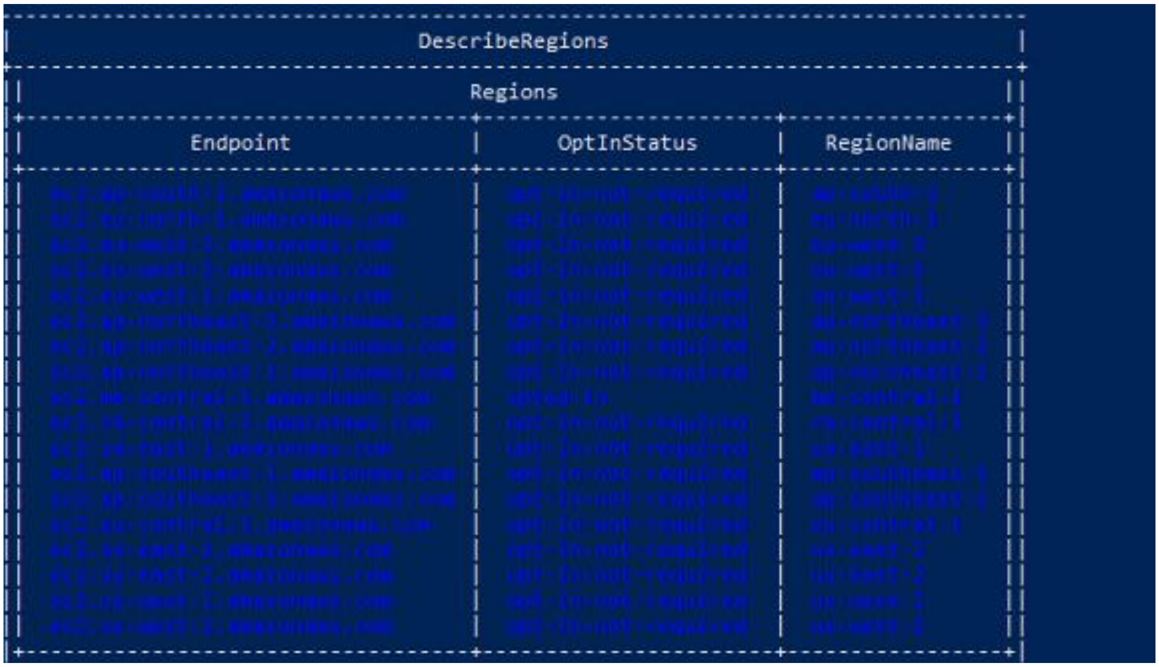
```
------------------------------------------------------------------------------------------------------------
|                                               DescribeVpcs                                               |
+----------------------------------------------------------------------------------------------------------+
||                                                   Vpcs                                                 ||
|+-------------+-------------------+-------------------+-----------+------------+------------+-------------+|
||  CidrBlock  |   DhcpOptionsId   | InstanceTenancy   | IsDefault |  OwnerId   |   State    |    VpcId    ||
|+-------------+-------------------+-------------------+-----------+------------+------------+-------------+|
||  [illegible]| [illegible]       | [illegible]       | [illegible]| [illegible]| [illegible]| [illegible] ||
|+-------------+-------------------+-------------------+-----------+------------+------------+-------------+|
|||                                        BlockPublicAccessStates                                       |||
||+-------------------------------------------------------------------------------------+----------------+||
||| [illegible]                                                                         | [illegible]    |||
||+-------------------------------------------------------------------------------------+----------------+||
|||                                        CidrBlockAssociationSet                                       |||
||+---------------------------------------------------------------------+--------------------------------+||
|||                            AssociationId                            |           CidrBlock            |||
||+---------------------------------------------------------------------+--------------------------------+||
||| [illegible]                                                         | [illegible]                    |||
||+---------------------------------------------------------------------+--------------------------------+||
||||                                           CidrBlockState                                           ||||
|||+---------------------------------------+------------------------------------------------------------+|||
|||| [illegible]                           | [illegible]                                                ||||
|||+---------------------------------------+------------------------------------------------------------+|||
```



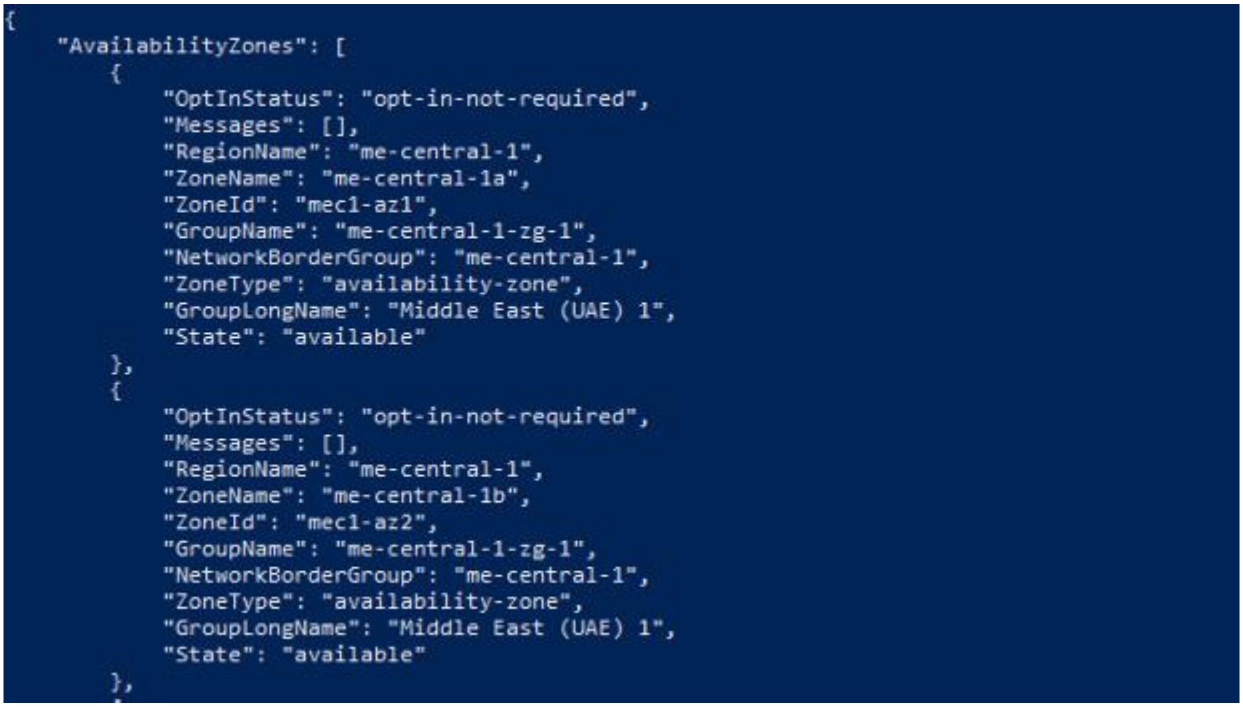
```
>   --query "Subnets[*].[SubnetId,VpcId,CidrBlock,AvailabilityZone]" \
>   --output table
-------------------------------------------------------------------------------------------------
|                                        DescribeSubnets                                        |
+-----------------------------+------------------------------+-------------------+---------------+
|  [illegible]                |  [illegible]                 |  [illegible]      |  [illegible]  |
|  [illegible]                |  [illegible]                 |  [illegible]      |  [illegible]  |
|  [illegible]                |  [illegible]                 |  [illegible]      |  [illegible]  |
+-----------------------------+------------------------------+-------------------+---------------+
```



```
{
    "Reservations": [
        {
            "ReservationId": "r-0c4e4227180946903",
            "OwnerId": "306601824237",
            "Groups": [],
            "Instances": [
                {
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2025-12-27T20:06:32+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-063fdc4fdd1635a02"
                            }
                        }
                    ],
                    "ClientToken": "f9241865-45b7-426f-a75d-0cda3383aa3e",
                    "EbsOptimized": false,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "NetworkInterfaces": [
```

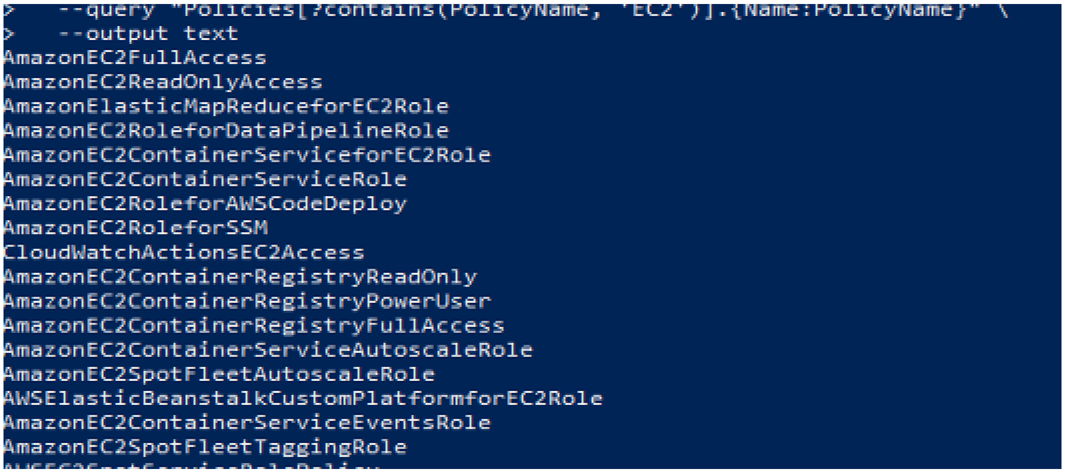


```
----------------------------------------------------------------------------------
|                                 DescribeRegions                                |
+--------------------------------------------------------------------------------+
||                                    Regions                                   ||
|+-----------------------------------+-----------------------+------------------+|
||              Endpoint             |      OptInStatus      |    RegionName    ||
|+-----------------------------------+-----------------------+------------------+|
||  ec2.ap-south-1.amazonaws.com     |  opt-in-not-required  |  ap-south-1      ||
||  ec2.eu-north-1.amazonaws.com     |  opt-in-not-required  |  eu-north-1      ||
||  ec2.eu-west-3.amazonaws.com      |  opt-in-not-required  |  eu-west-3       ||
||  ec2.eu-west-2.amazonaws.com      |  opt-in-not-required  |  eu-west-2       ||
||  ec2.eu-west-1.amazonaws.com      |  opt-in-not-required  |  eu-west-1       ||
||  ec2.ap-northeast-3.amazonaws.com |  opt-in-not-required  |  ap-northeast-3  ||
||  ec2.ap-northeast-2.amazonaws.com |  opt-in-not-required  |  ap-northeast-2  ||
||  ec2.ap-northeast-1.amazonaws.com |  opt-in-not-required  |  ap-northeast-1  ||
||  ec2.me-central-1.amazonaws.com   |  opted-in             |  me-central-1    ||
||  ec2.ca-central-1.amazonaws.com   |  opt-in-not-required  |  ca-central-1    ||
||  ec2.sa-east-1.amazonaws.com      |  opt-in-not-required  |  sa-east-1       ||
||  ec2.ap-southeast-1.amazonaws.com |  opt-in-not-required  |  ap-southeast-1  ||
||  ec2.ap-southeast-2.amazonaws.com |  opt-in-not-required  |  ap-southeast-2  ||
||  ec2.eu-central-1.amazonaws.com   |  opt-in-not-required  |  eu-central-1    ||
||  ec2.us-east-1.amazonaws.com      |  opt-in-not-required  |  us-east-1       ||
||  ec2.us-east-2.amazonaws.com      |  opt-in-not-required  |  us-east-2       ||
||  ec2.us-west-1.amazonaws.com      |  opt-in-not-required  |  us-west-1       ||
||  ec2.us-west-2.amazonaws.com      |  opt-in-not-required  |  us-west-2       ||
|+-----------------------------------+-----------------------+------------------+|
```

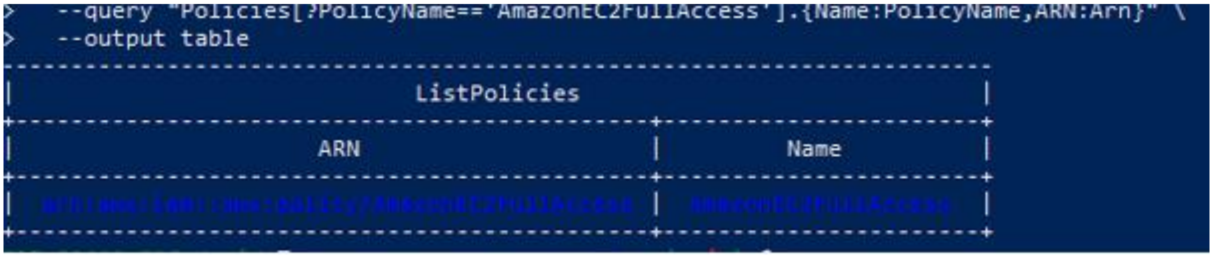


```
{
    "AvailabilityZones": [
        {
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "me-central-1",
            "ZoneName": "me-central-1a",
            "ZoneId": "mec1-az1",
            "GroupName": "me-central-1-zg-1",
            "NetworkBorderGroup": "me-central-1",
            "ZoneType": "availability-zone",
            "GroupLongName": "Middle East (UAE) 1",
            "State": "available"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "me-central-1",
            "ZoneName": "me-central-1b",
            "ZoneId": "mec1-az2",
            "GroupName": "me-central-1-zg-1",
            "NetworkBorderGroup": "me-central-1",
            "ZoneType": "availability-zone",
            "GroupLongName": "Middle East (UAE) 1",
            "State": "available"
        },
```

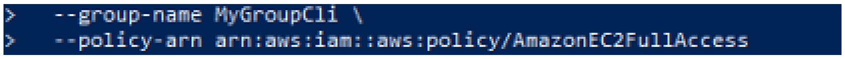
Task 6 — IAM: create group, user, attach policies, create console login & keys



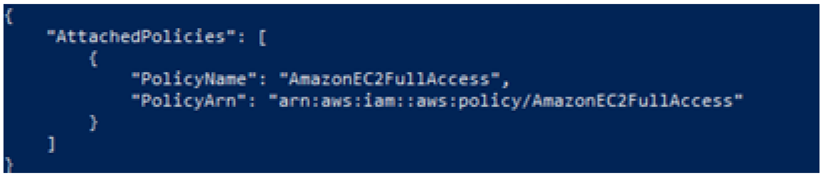
```
>   --query "Policies[?contains(PolicyName, 'EC2')].{Name:PolicyName}" \
>   --output text
AmazonEC2FullAccess
AmazonEC2ReadOnlyAccess
AmazonElasticMapReduceforEC2Role
AmazonEC2RoleforDataPipelineRole
AmazonEC2ContainerServiceforEC2Role
AmazonEC2ContainerServiceRole
AmazonEC2RoleforAWSCodeDeploy
AmazonEC2RoleforSSM
CloudWatchActionsEC2Access
AmazonEC2ContainerRegistryReadOnly
AmazonEC2ContainerRegistryPowerUser
AmazonEC2ContainerRegistryFullAccess
AmazonEC2ContainerServiceAutoscaleRole
AmazonEC2SpotFleetAutoscaleRole
AWSElasticBeanstalkCustomPlatformforEC2Role
AmazonEC2ContainerServiceEventsRole
AmazonEC2SpotFleetTaggingRole
```



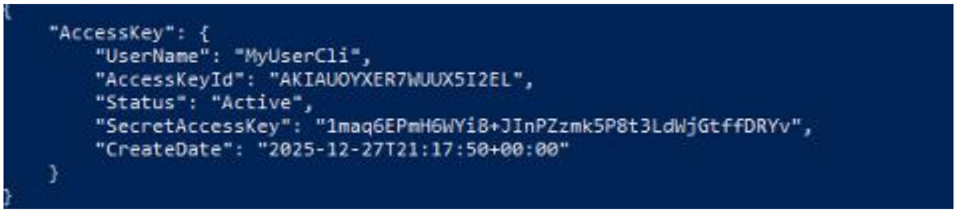
```
>   --query "Policies[?PolicyName=='AmazonEC2FullAccess'].{Name:PolicyName,ARN:Arn}" \
>   --output table
-------------------------------------------------------------------------
|                             ListPolicies                              |
+-----------------------------------------------+-----------------------+
|                      ARN                      |         Name          |
+-----------------------------------------------+-----------------------+
|  arn:aws:iam::aws:policy/AmazonEC2FullAccess  |  AmazonEC2FullAccess  |
+-----------------------------------------------+-----------------------+
```



```
>   --group-name MyGroupCli \
>   --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
```

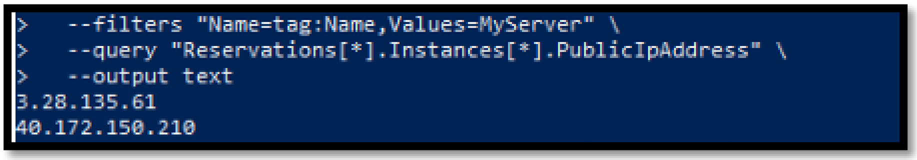


```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonEC2FullAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
        }
    ]
}
```

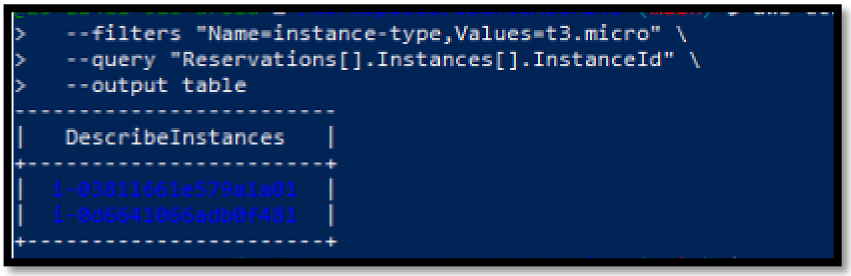


```
{
    "AccessKey": {
        "UserName": "MyUserCli",
        "AccessKeyId": "AKIAUOYXER7WUUX5I2EL",
        "Status": "Active",
        "SecretAccessKey": "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",
        "CreateDate": "2025-12-27T21:17:50+00:00"
    }
}
```

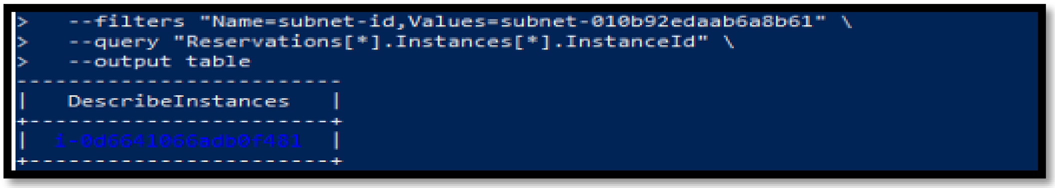
Task 7 — Filters: query with filters to find instances and their attributes



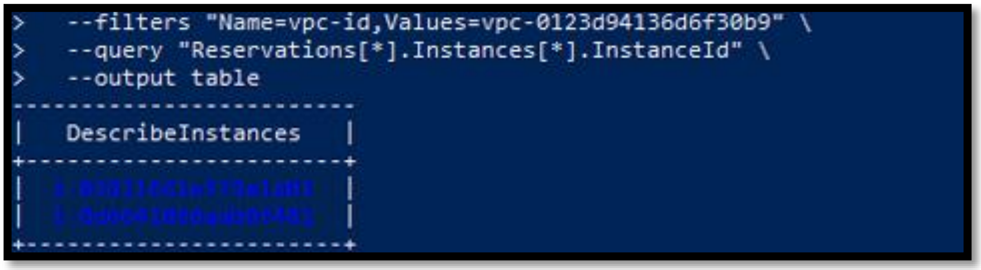
```
>   --filters "Name=tag:Name,Values=MyServer" \
>   --query "Reservations[*].Instances[*].PublicIpAddress" \
>   --output text
3.28.135.61
40.172.150.210
```



```
>   --filters "Name=instance-type,Values=t3.micro" \
>   --query "Reservations[].Instances[].InstanceId" \
>   --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-03811661e579a1a01  |
|  i-0d6641066adb0f481  |
+-----------------------+
```

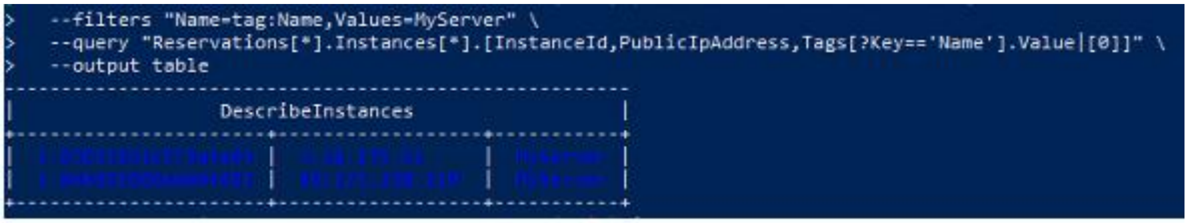


```
>   --filters "Name=subnet-id,Values=subnet-010b92edaab6a8b61" \
>   --query "Reservations[*].Instances[*].InstanceId" \
>   --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-0d6641066adb0f481  |
+-----------------------+
```

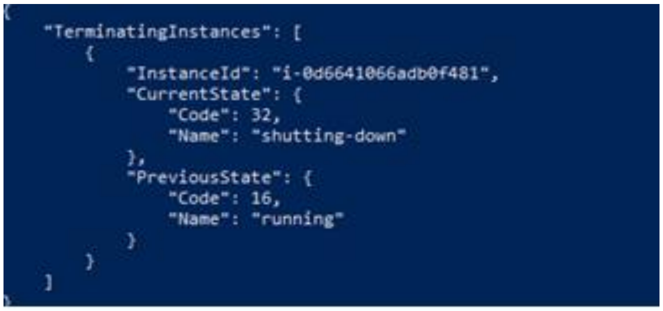


```
>   --filters "Name=vpc-id,Values=vpc-0123d94136d6f30b9" \
>   --query "Reservations[*].Instances[*].InstanceId" \
>   --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-03811661e579a1a01  |
|  i-0d6641066adb0f481  |
+-----------------------+
```

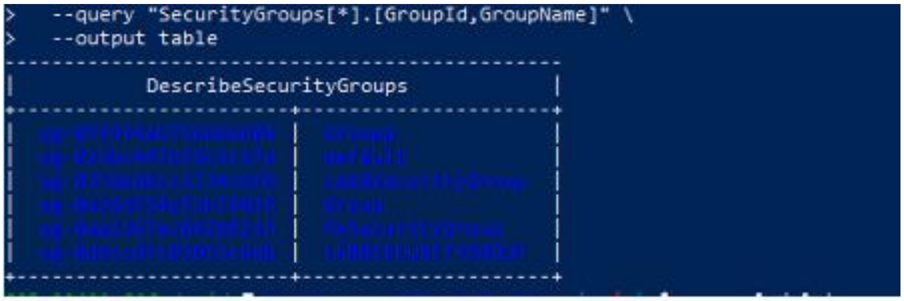
Task 8 — Use --query to format outputs for reporting Examples (run each and take a screenshot immediately after):



```
>   --filters "Name=tag:Name,Values=MyServer" \
>   --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress,Tags[?Key=='Name'].Value|[0]]" \
>   --output table
-------------------------------------------------------
|                  DescribeInstances                  |
+---------------------+-------------------+-----------+
|  [illegible]        |  [illegible]      | [illegible] |
|  [illegible]        |  [illegible]      | [illegible] |
+---------------------+-------------------+-----------+
```



```
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-0d6641066adb0f481",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
```



```
>   --query "SecurityGroups[*].[GroupId,GroupName]" \
>   --output table
----------------------------------------------
|           DescribeSecurityGroups           |
+------------------------+-------------------+
|  [illegible]           |  [illegible]      |
|  [illegible]           |  default          |
|  [illegible]           |  [illegible]      |
|  [illegible]           |  [illegible]      |
|  [illegible]           |  [illegible]      |
|  [illegible]           |  [illegible]      |
+------------------------+-------------------+
```