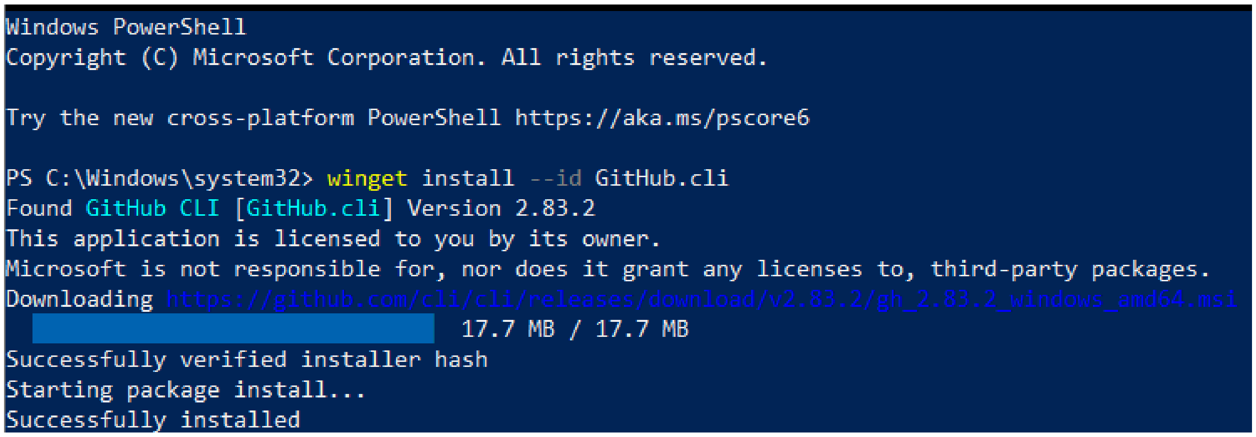
NAME: SANA TARIQ
CLASS: BSE 5B
REG.NO: 058
SUBJECT: CLOUD COMPUTING

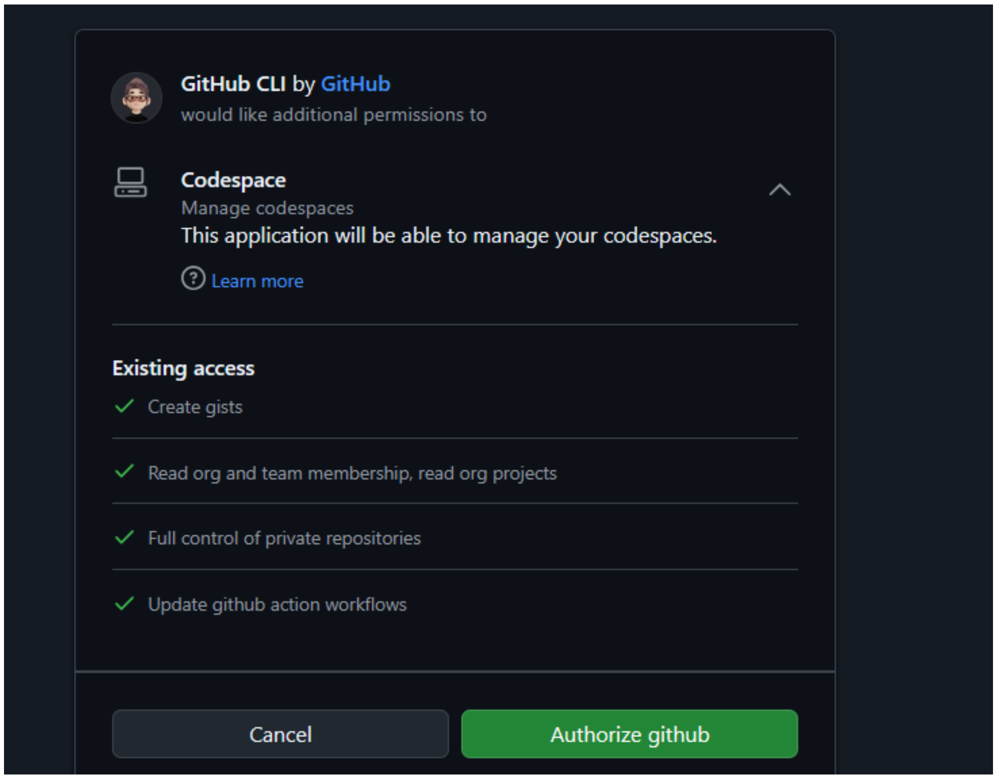
LAB#09

TASK 1:

GitHub CLI & Codespace Setup



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

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

PS C:\Windows\system32> winget install --id GitHub.cli
Found GitHub CLI [GitHub.cli] Version 2.83.2
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://github.com/cli/cli/releases/download/v2.83.2/gh_2.83.2_windows_amd64.msi
  ██████████████████████████████  17.7 MB / 17.7 MB
Successfully verified installer hash
Starting package install...
Successfully installed
```

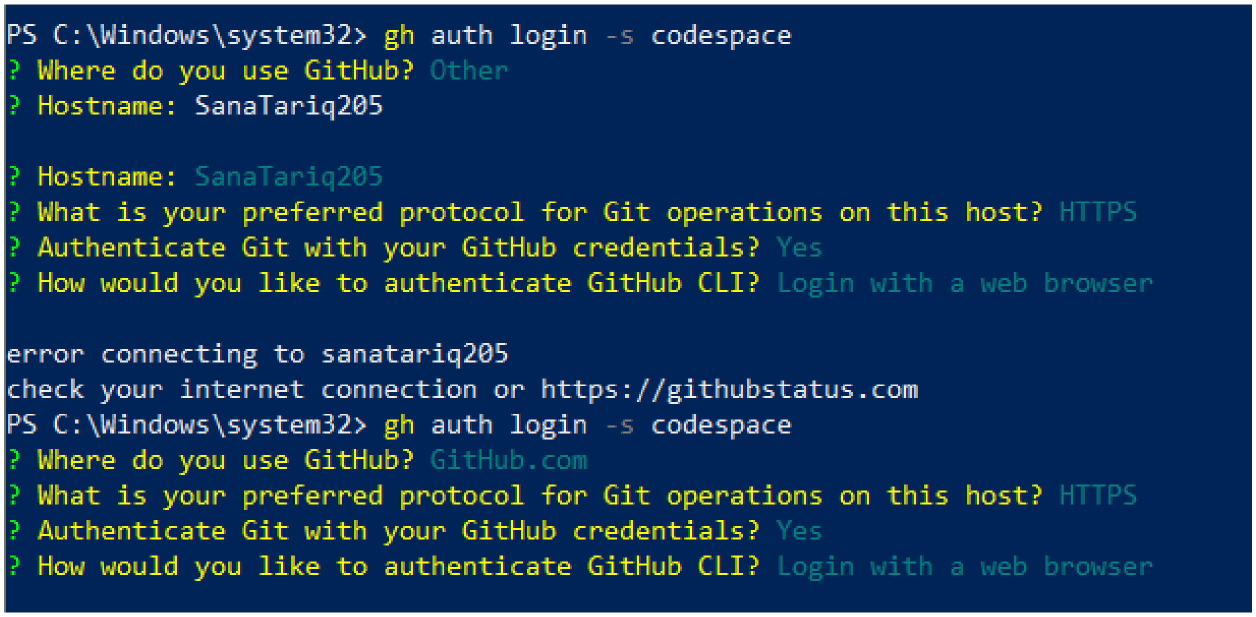


GitHub CLI by GitHub
would like additional permissions to

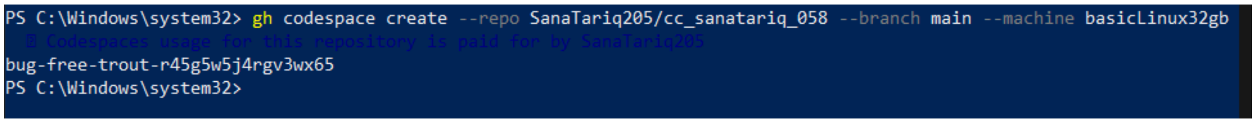
Codespace
Manage codespaces
This application will be able to manage your codespaces.

Existing access

- ✓ Create gists
- ✓ Read org and team membership, read org projects
- ✓ Full control of private repositories
- ✓ Update github action workflows



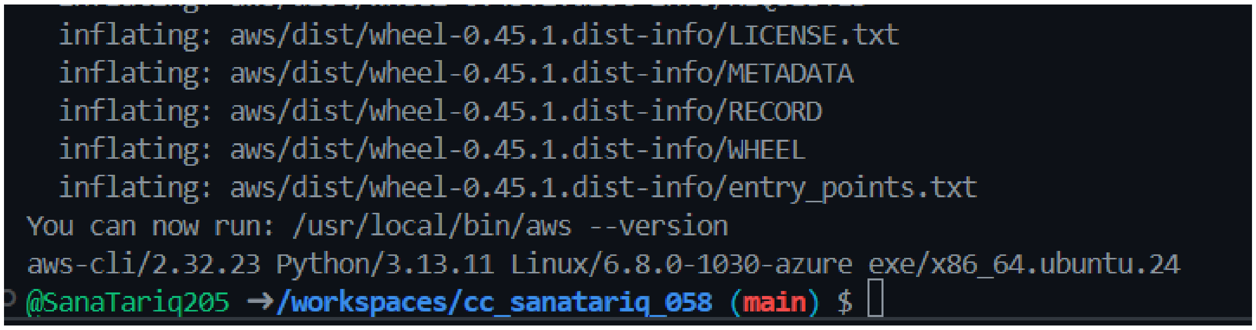
```
PS C:\Windows\system32> gh auth login -s codespace
? Where do you use GitHub? Other
? Hostname: SanaTariq205

? Hostname: SanaTariq205
? 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

error connecting to sanatariq205
check your internet connection or https://githubstatus.com
PS C:\Windows\system32> 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
```



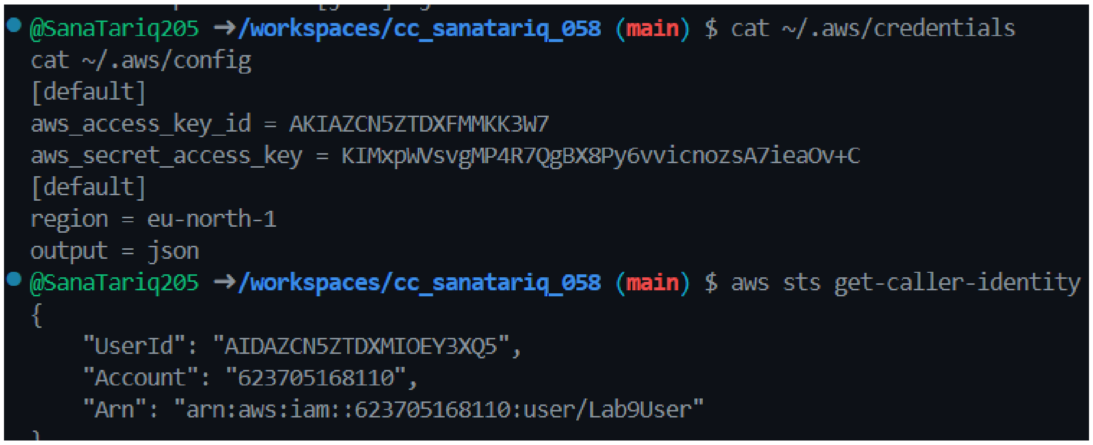
```
PS C:\Windows\system32> gh codespace create --repo SanaTariq205/cc_sanatariq_058 --branch main --machine basicLinux32gb
  ? Codespaces usage for this repository is paid for by SanaTariq205
bug-free-trout-r45g5w5j4rgv3wx65
PS C:\Windows\system32>
```

TASK 2:

Install AWS CLI in Codespace



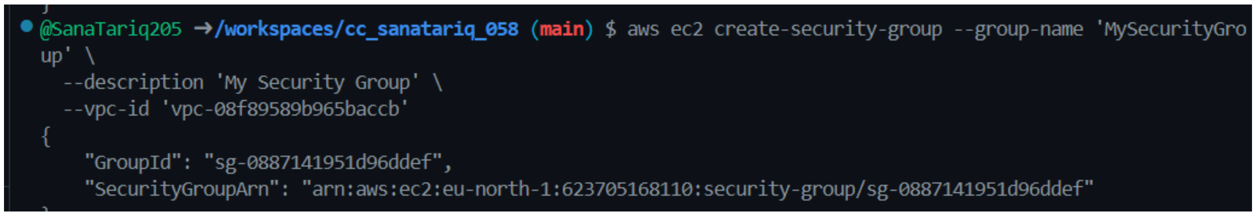
```
  inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
  inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
  inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
  inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
  inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
You can now run: /usr/local/bin/aws --version
aws-cli/2.32.23 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $
```



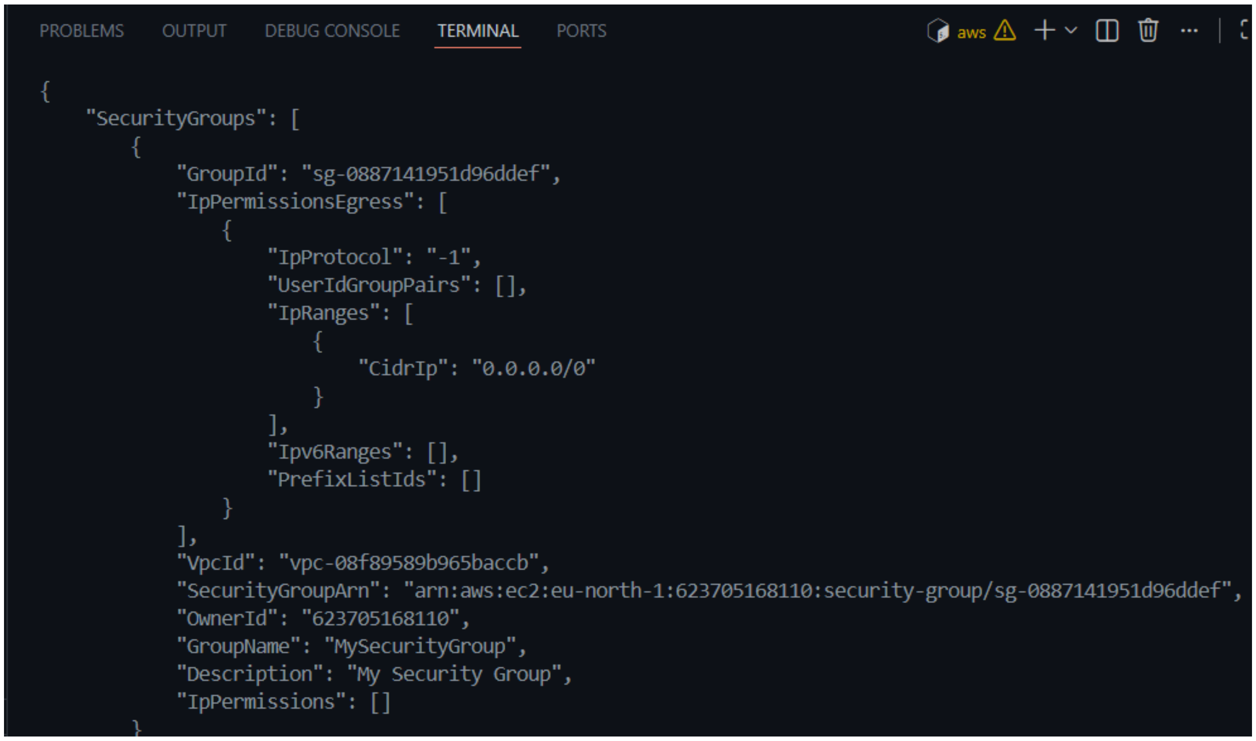
```
● @SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ cat ~/.aws/credentials
  cat ~/.aws/config
  [default]
  aws_access_key_id = AKIAZCN5ZTDXFMMKK3W7
  aws_secret_access_key = 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
  [default]
  region = eu-north-1
  output = json
● @SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws sts get-caller-identity
  {
      "UserId": "AIDAZCN5ZTDXMIOEY3XQ5",
      "Account": "623705168110",
      "Arn": "arn:aws:iam::623705168110:user/Lab9User"
```

TASK 3:

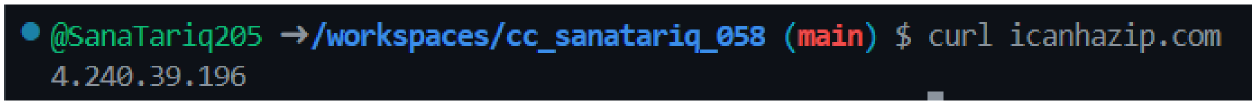
Create Security Group



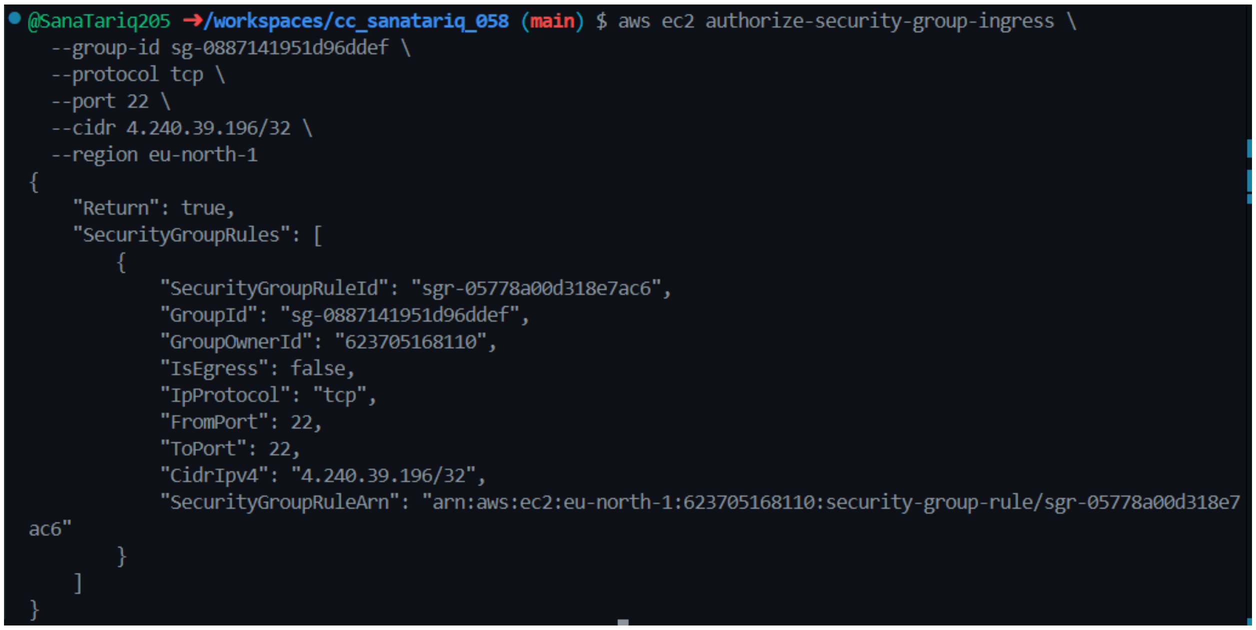
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 create-security-group --group-name 'MySecurityGro
up' \
  --description 'My Security Group' \
  --vpc-id 'vpc-08f89589b965baccb'
{
    "GroupId": "sg-0887141951d96ddef",
    "SecurityGroupArn": "arn:aws:ec2:eu-north-1:623705168110:security-group/sg-0887141951d96ddef"
```



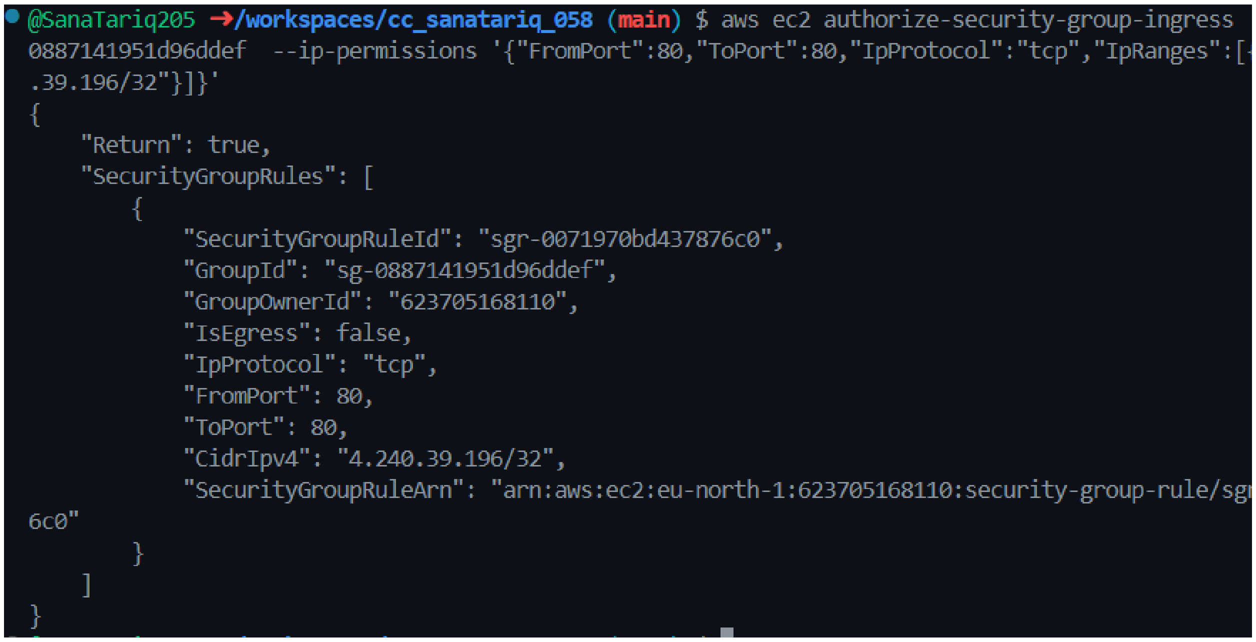
```
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0887141951d96ddef",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-08f89589b965baccb",
            "SecurityGroupArn": "arn:aws:ec2:eu-north-1:623705168110:security-group/sg-0887141951d96ddef",
            "OwnerId": "623705168110",
            "GroupName": "MySecurityGroup",
            "Description": "My Security Group",
            "IpPermissions": []
        }
```



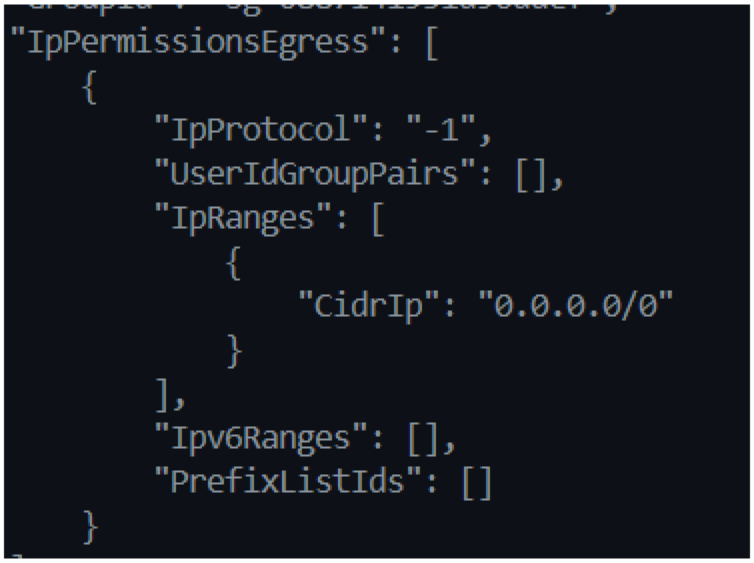
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ curl icanhazip.com
4.240.39.196
```



```
@SanaTariq205 ➜ /workspaces/cc_sanatariq_058 (main) $ aws ec2 authorize-security-group-ingress \
  --group-id sg-0887141951d96ddef \
  --protocol tcp \
  --port 22 \
  --cidr 4.240.39.196/32 \
  --region eu-north-1
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-05778a00d318e7ac6",
            "GroupId": "sg-0887141951d96ddef",
            "GroupOwnerId": "623705168110",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "4.240.39.196/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:eu-north-1:623705168110:security-group-rule/sgr-05778a00d318e7
ac6"
        }
    ]
}
```



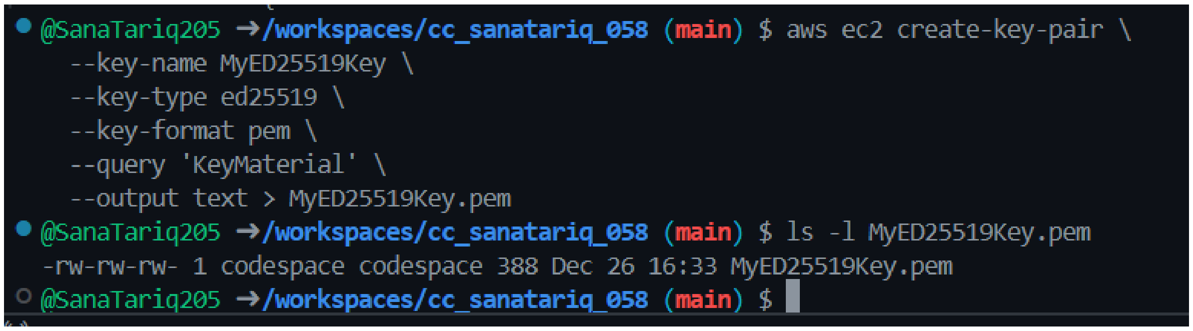
```
@SanaTariq205 ➜ /workspaces/cc_sanatariq_058 (main) $ aws ec2 authorize-security-group-ingress
0887141951d96ddef  --ip-permissions '{"FromPort":80,"ToPort":80,"IpProtocol":"tcp","IpRanges":[{
.39.196/32"}]}'
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0071970bd437876c0",
            "GroupId": "sg-0887141951d96ddef",
            "GroupOwnerId": "623705168110",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "4.240.39.196/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:eu-north-1:623705168110:security-group-rule/sgr
6c0"
        }
    ]
}
```



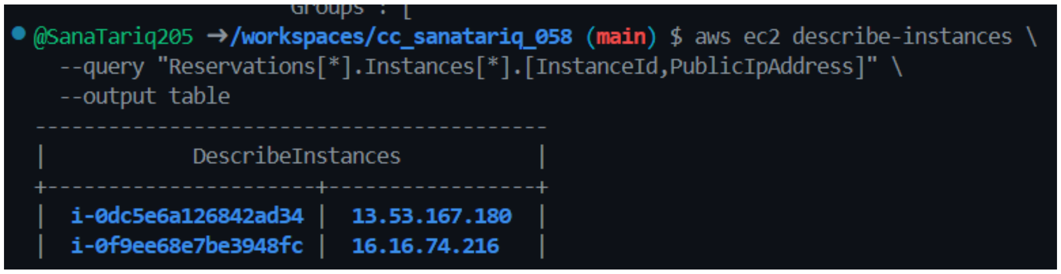
```
"IpPermissionsEgress": [
    {
        "IpProtocol": "-1",
        "UserIdGroupPairs": [],
        "IpRanges": [
            {
                "CidrIp": "0.0.0.0/0"
            }
        ],
        "Ipv6Ranges": [],
        "PrefixListIds": []
    }
```

TASK 4:

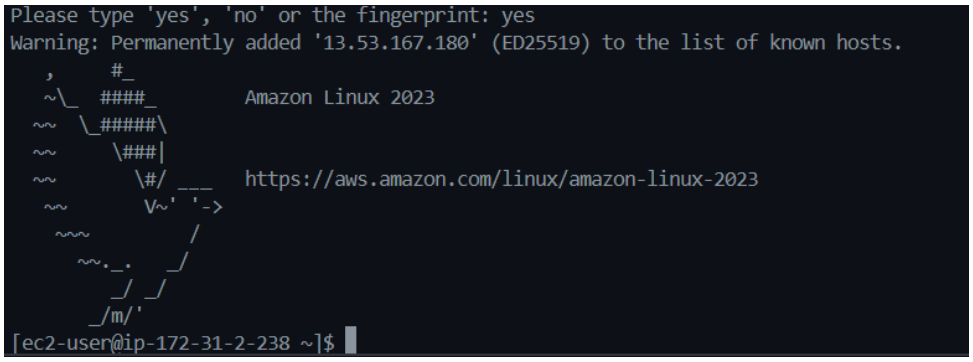
Create Key Pair & Launch EC2



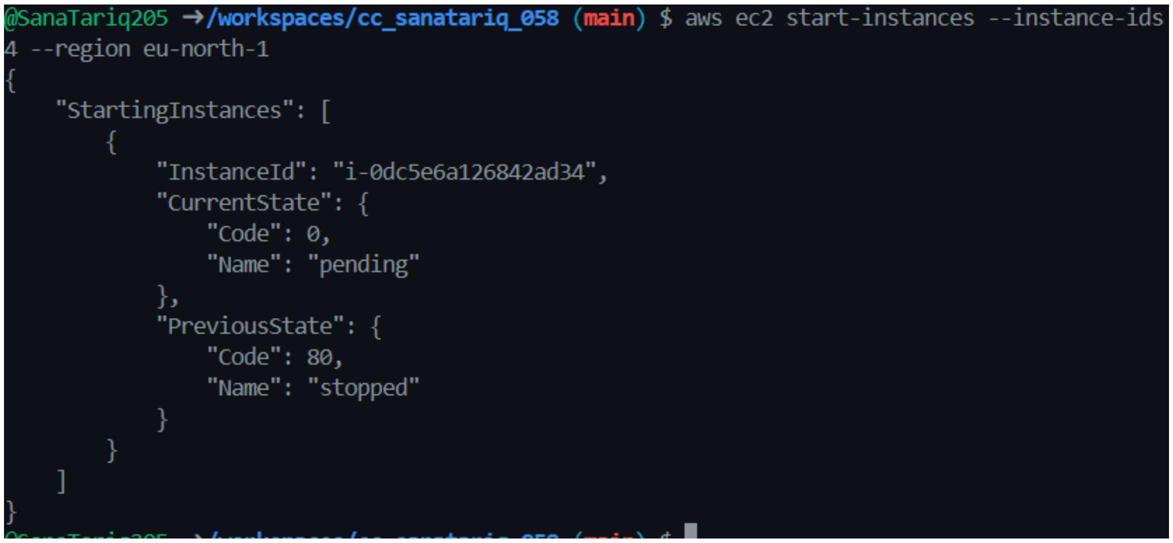
```
● @SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 create-key-pair \
    --key-name MyED25519Key \
    --key-type ed25519 \
    --key-format pem \
    --query 'KeyMaterial' \
    --output text > MyED25519Key.pem
● @SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ ls -l MyED25519Key.pem
  -rw-rw-rw- 1 codespace codespace 388 Dec 26 16:33 MyED25519Key.pem
○ @SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ 
```



```
● @SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 describe-instances \
    --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress]" \
    --output table
  ------------------------------------------
  |           DescribeInstances            |
  +----------------------+-----------------+
  |  i-0dc5e6a126842ad34 |  13.53.167.180  |
  |  i-0f9ee68e7be3948fc |  16.16.74.216   |
```



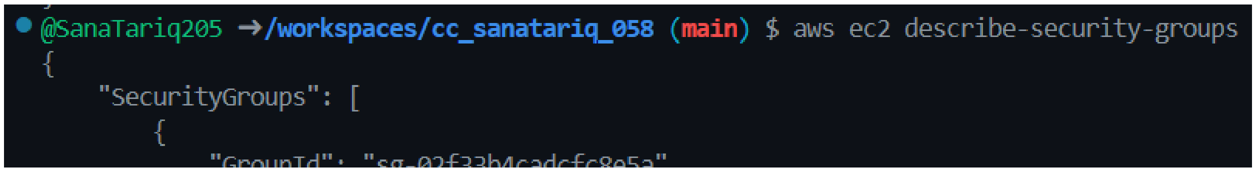
```
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '13.53.167.180' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-2-238 ~]$ 
```



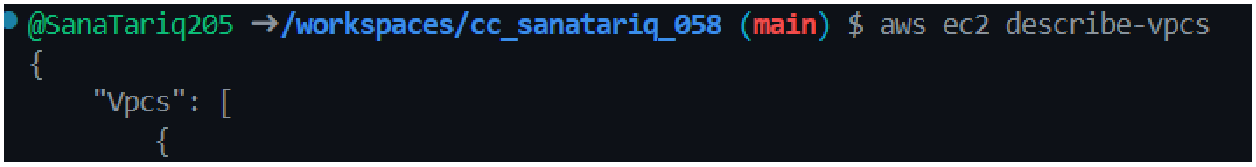
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 start-instances --instance-ids
4 --region eu-north-1
{
    "StartingInstances": [
        {
            "InstanceId": "i-0dc5e6a126842ad34",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

TASK 5:

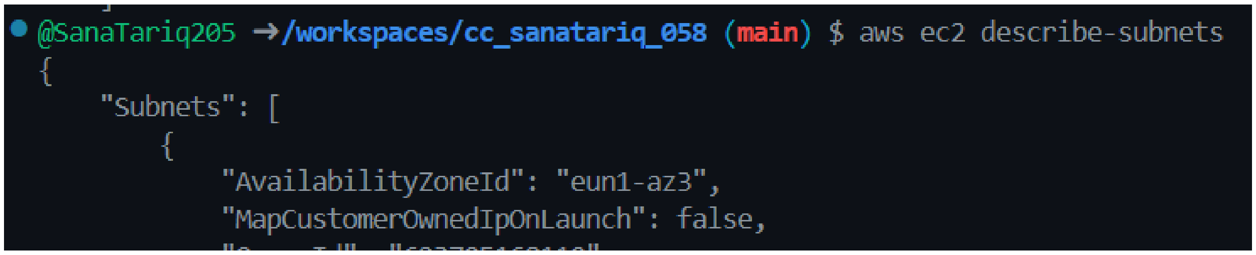
Describe Commands



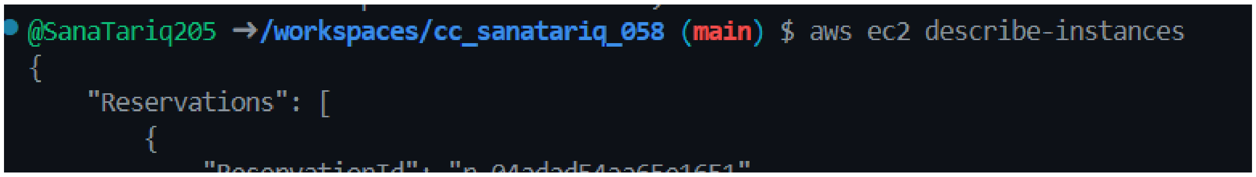
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 describe-security-groups
{
    "SecurityGroups": [
        {
```



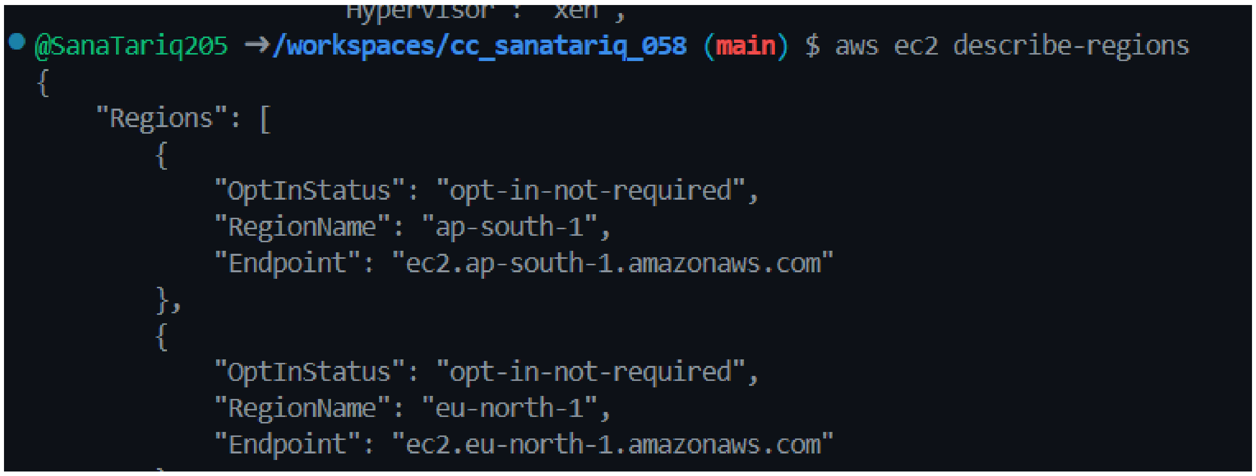
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 describe-vpcs
{
    "Vpcs": [
        {
```



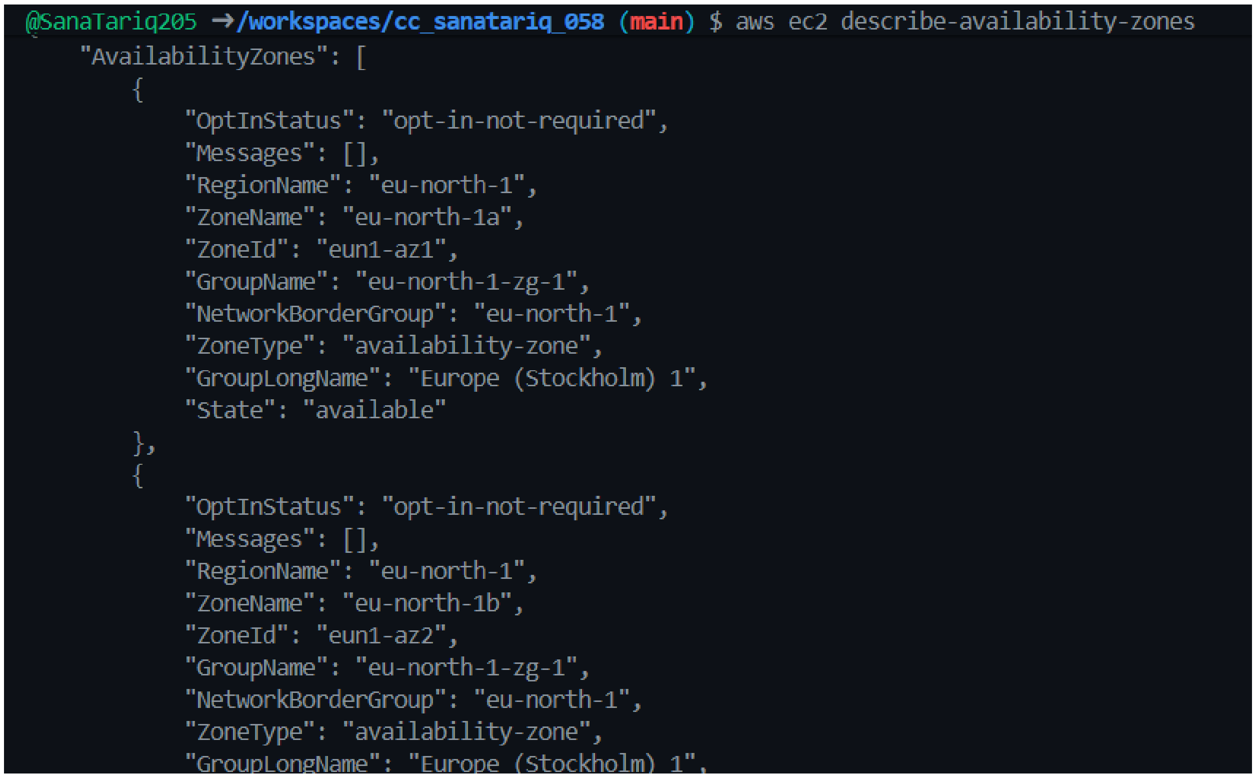
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 describe-subnets
{
    "Subnets": [
        {
            "AvailabilityZoneId": "eun1-az3",
            "MapCustomerOwnedIpOnLaunch": false,
```



```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 describe-instances
{
    "Reservations": [
        {
```



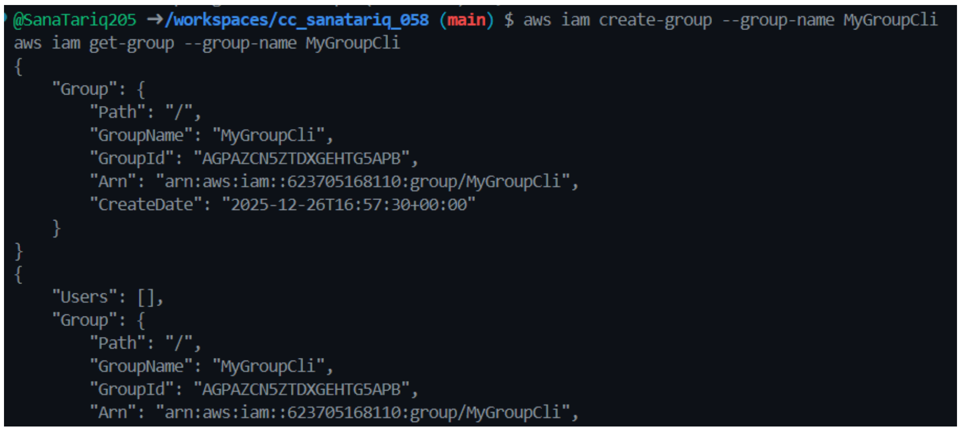
```
        "Hypervisor": "xen",
@SanaTariq205 ➜ /workspaces/cc_sanatariq_058 (main) $ aws ec2 describe-regions
{
    "Regions": [
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "ap-south-1",
            "Endpoint": "ec2.ap-south-1.amazonaws.com"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "eu-north-1",
            "Endpoint": "ec2.eu-north-1.amazonaws.com"
```



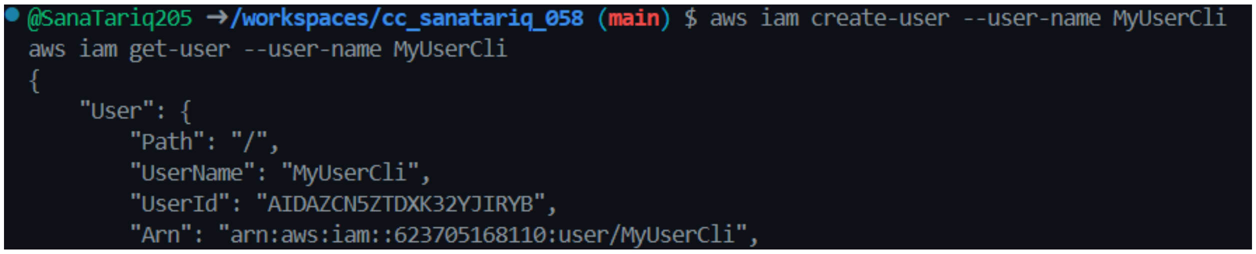
```
@SanaTariq205 ➜ /workspaces/cc_sanatariq_058 (main) $ aws ec2 describe-availability-zones
    "AvailabilityZones": [
        {
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "eu-north-1",
            "ZoneName": "eu-north-1a",
            "ZoneId": "eun1-az1",
            "GroupName": "eu-north-1-zg-1",
            "NetworkBorderGroup": "eu-north-1",
            "ZoneType": "availability-zone",
            "GroupLongName": "Europe (Stockholm) 1",
            "State": "available"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "eu-north-1",
            "ZoneName": "eu-north-1b",
            "ZoneId": "eun1-az2",
            "GroupName": "eu-north-1-zg-1",
            "NetworkBorderGroup": "eu-north-1",
            "ZoneType": "availability-zone",
            "GroupLongName": "Europe (Stockholm) 1",
```

TASK 6:

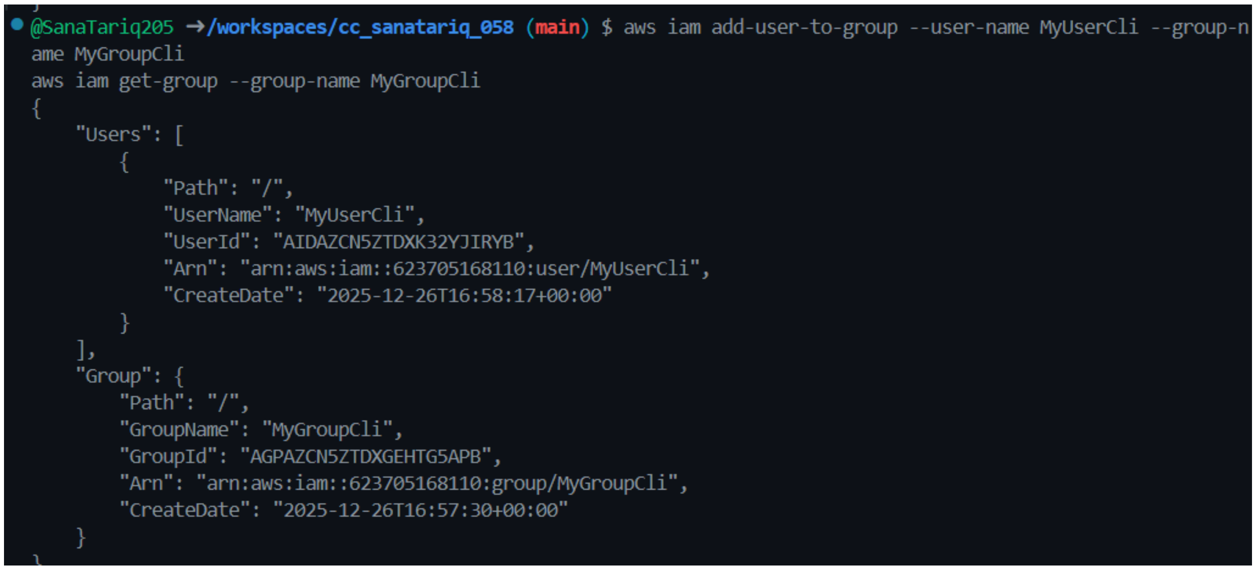
IAM Users & Groups



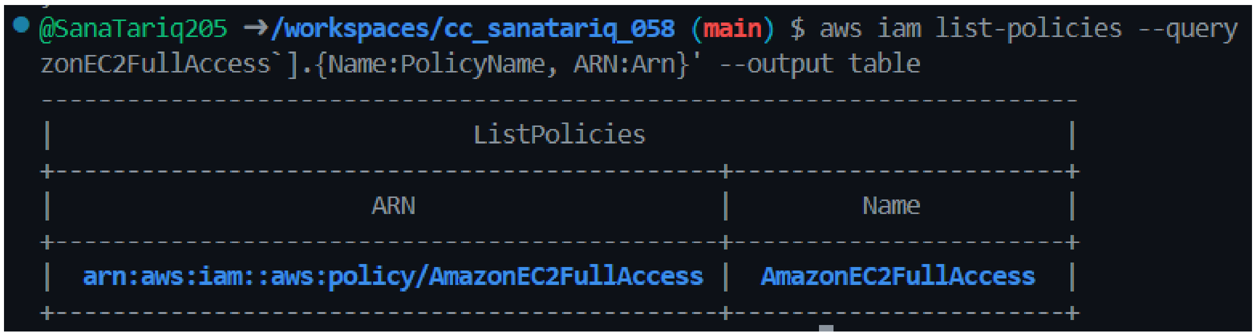
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws iam create-group --group-name MyGroupCli
aws iam get-group --group-name MyGroupCli
{
    "Group": {
        "Path": "/",
        "GroupName": "MyGroupCli",
        "GroupId": "AGPAZCN5ZTDXGEHTG5APB",
        "Arn": "arn:aws:iam::623705168110:group/MyGroupCli",
        "CreateDate": "2025-12-26T16:57:30+00:00"
    }
}
{
    "Users": [],
    "Group": {
        "Path": "/",
        "GroupName": "MyGroupCli",
        "GroupId": "AGPAZCN5ZTDXGEHTG5APB",
        "Arn": "arn:aws:iam::623705168110:group/MyGroupCli",
```



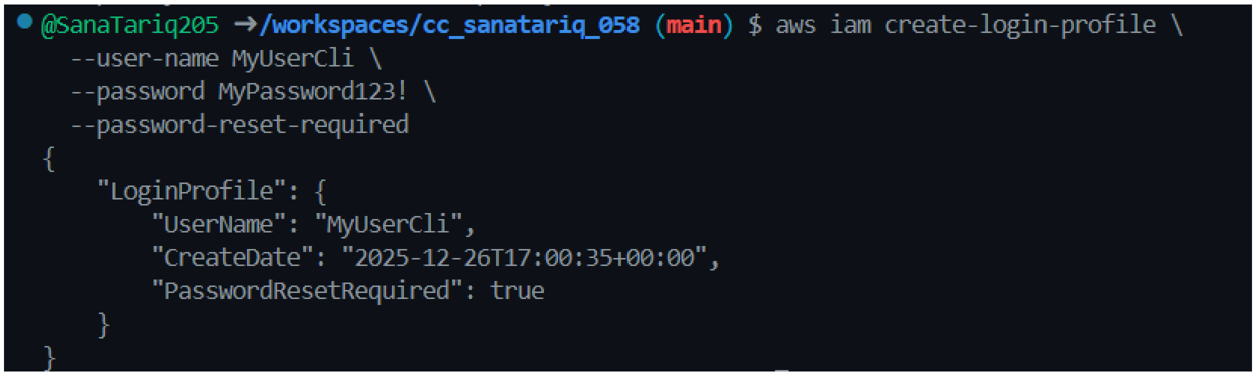
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws iam create-user --user-name MyUserCli
aws iam get-user --user-name MyUserCli
{
    "User": {
        "Path": "/",
        "UserName": "MyUserCli",
        "UserId": "AIDAZCN5ZTDXK32YJIRYB",
        "Arn": "arn:aws:iam::623705168110:user/MyUserCli",
```



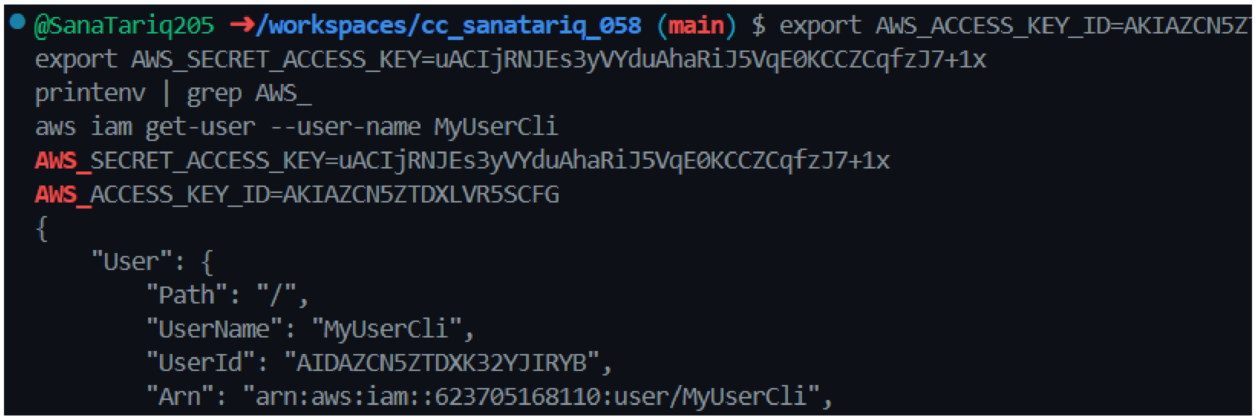
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws iam add-user-to-group --user-name MyUserCli --group-n
ame MyGroupCli
aws iam get-group --group-name MyGroupCli
{
    "Users": [
        {
            "Path": "/",
            "UserName": "MyUserCli",
            "UserId": "AIDAZCN5ZTDXK32YJIRYB",
            "Arn": "arn:aws:iam::623705168110:user/MyUserCli",
            "CreateDate": "2025-12-26T16:58:17+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "MyGroupCli",
        "GroupId": "AGPAZCN5ZTDXGEHTG5APB",
        "Arn": "arn:aws:iam::623705168110:group/MyGroupCli",
        "CreateDate": "2025-12-26T16:57:30+00:00"
    }
```



```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws iam list-policies --query
zonEC2FullAccess`].{Name:PolicyName, ARN:Arn}' --output table
-----------------------------------------------------------------------
|                            ListPolicies                             |
+----------------------------------------------+----------------------+
|                     ARN                      |         Name         |
+----------------------------------------------+----------------------+
|  arn:aws:iam::aws:policy/AmazonEC2FullAccess |  AmazonEC2FullAccess |
+----------------------------------------------+----------------------+
```



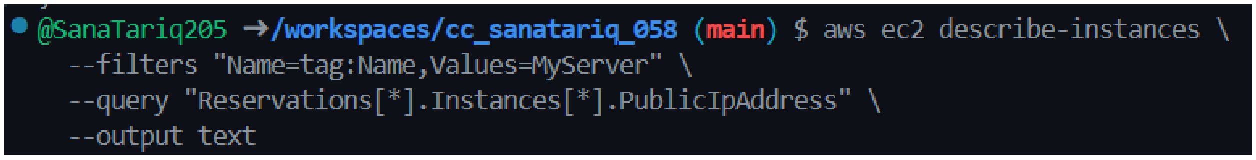
```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws iam create-login-profile \
  --user-name MyUserCli \
  --password MyPassword123! \
  --password-reset-required
{
    "LoginProfile": {
        "UserName": "MyUserCli",
        "CreateDate": "2025-12-26T17:00:35+00:00",
        "PasswordResetRequired": true
    }
}
```



```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ export AWS_ACCESS_KEY_ID=AKIAZCN5Z
export AWS_SECRET_ACCESS_KEY=uACIjRNJEs3yVYduAhaRiJ5VqE0KCCZCqfzJ7+1x
printenv | grep AWS_
aws iam get-user --user-name MyUserCli
AWS_SECRET_ACCESS_KEY=uACIjRNJEs3yVYduAhaRiJ5VqE0KCCZCqfzJ7+1x
AWS_ACCESS_KEY_ID=AKIAZCN5ZTDXLVR5SCFG
{
    "User": {
        "Path": "/",
        "UserName": "MyUserCli",
        "UserId": "AIDAZCN5ZTDXK32YJIRYB",
        "Arn": "arn:aws:iam::623705168110:user/MyUserCli",
```

TASK 7 & 8:

Filters and Queries



```
@SanaTariq205 →/workspaces/cc_sanatariq_058 (main) $ aws ec2 describe-instances \
  --filters "Name=tag:Name,Values=MyServer" \
  --query "Reservations[*].Instances[*].PublicIpAddress" \
  --output text
```