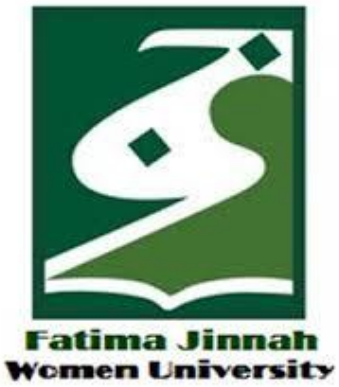


CLOUD COMPUTING LAB: 09

Submitted By:

Seerat Fatima

Registration. No:

2023-BSE-060

Submitted To:

Engr. Shoaib

Section:

5B

Task 1 — GitHub CLI, Codespace setup and authentication

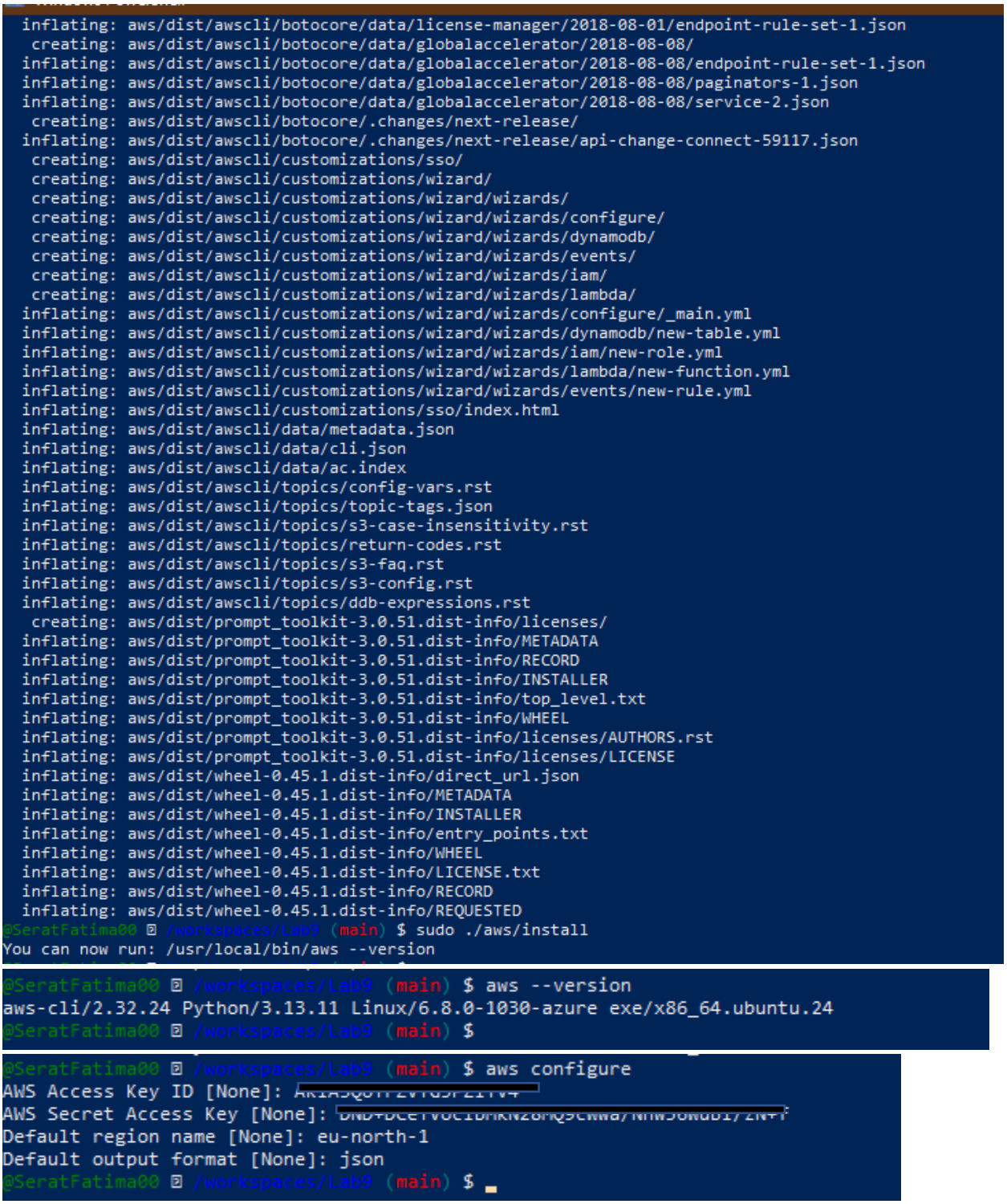
```
PS C:\Users\Waseem> winget install --id GitHub.cli
The `msstore` source requires that you view the following agreements before using.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
The source requires the current machine's 2-letter geographic region to be sent to the backend service to function properly (ex. "US").

Do you agree to all the source agreements terms?
[Y] Yes  [N] No: Y
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
PS C:\Users\Waseem>
```

```
PS C:\Users\Waseem> 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? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.
? Paste your authentication token: ****************************************
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as SeratFatima00
PS C:\Users\Waseem>
```

```
PS C:\Users\Waseem> gh codespace create --repo SeratFatima00/Lab9 --branch main --machine basicLinux32gb
  ⚠ Codespaces usage for this repository is paid for by SeratFatima00
verbose-system-wrq7q47rjx7vcgq46
PS C:\Users\Waseem> gh codespace list
NAME                                DISPLAY NAME    REPOSITORY          BRANCH  STATE      CREATED AT
verbose-system-wrq7q47rjx7vcgq46    verbose system  SeratFatima00/Lab9  main    Available  about 1 minute ago
PS C:\Users\Waseem>
```

```
PS C:\Users\Waseem> gh codespace ssh -c verbose-system-wrq7q47rjx7vcgq46
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.

@SeratFatima00 ➜ /workspaces/Lab9 (main) $ Connection to localhost closed by remote host.
Connection to localhost closed.
shell closed: exit status 0xffffffff
PS C:\Users\Waseem> gh codespace ssh -c verbose-system-wrq7q47rjx7vcgq46
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 Dec 27 06:00:29 2025 from ::1
@SeratFatima00 ➜ /workspaces/Lab9 (main) $ uname -a
Linux codespaces-cd4872 6.8.0-1030-azure #35~22.04.1-Ubuntu SMP Mon May 26 18:08:30 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
@SeratFatima00 ➜ /workspaces/Lab9 (main) $
```

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



```
@SeratFatima00 /workspaces/Lab9 (main) $ cat ~/.aws/credentials
[default]
aws_access_key_id = [redacted]
aws_secret_access_key = [redacted]
@SeratFatima00 /workspaces/Lab9 (main) $ cat ~/.aws/config
[default]
region = eu-north-1
output = json
@SeratFatima00 /workspaces/Lab9 (main) $
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws sts get-caller-identity
{
    "UserId": "AIDA3QUYFZVYDKQOBZV7R",
    "Account": "791666871664",
    "Arn": "arn:aws:iam::791666871664:user/Admin"
}
@SeratFatima00 /workspaces/Lab9 (main) $
```

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

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 create-security-group \
>   --group-name MySecurityGroup \
>   --description "My Security Group" \
>   --vpc-id vpc-0be30d1beb39c1848
{
    "GroupId": "sg-0ae675cd27298fd47",
    "SecurityGroupArn": "arn:aws:ec2:eu-north-1:791666871664:security-group/sg-0ae675cd27298fd47"
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-0ae675cd27298fd47
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0ae675cd27298fd47",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0be30d1beb39c1848",
            "SecurityGroupArn": "arn:aws:ec2:eu-north-1:791666871664:security-group/sg-0ae675cd27298fd47",
            "OwnerId": "791666871664",
            "GroupName": "MySecurityGroup",
            "Description": "My Security Group",
            "IpPermissions": []
        }
    ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ curl icanhazip.com
4.240.39.192
@SeratFatima00 /workspaces/Lab9 (main) $
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 authorize-security-group-ingress \
>   --group-id sg-0ae675cd27298fd47 \
>   --protocol tcp \
>   --port 22 \
>   --cidr 4.240.39.192/32
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0ed5f5ee801b6a72e",
            "GroupId": "sg-0ae675cd27298fd47",
            "GroupOwnerId": "791666871664",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "4.240.39.192/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:eu-north-1:791666871664:security-group-rule/sgr-0ed5f5ee801b6a72e"
        }
    ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-0ae675cd27298fd4
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0ae675cd27298fd47",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0be30d1beb39c1848",
            "SecurityGroupArn": "arn:aws:ec2:eu-north-1:791666871664:security-group/sg-0ae675cd27298fd47",
            "OwnerId": "791666871664",
            "GroupName": "MySecurityGroup",
            "Description": "My Security Group",
            "IpPermissions": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "4.240.39.192/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ]
        }
    ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 authorize-security-group-ingress \
>   --group-id sg-0ae675cd27298fd47 \
>   --ip-permissions '{"FromPort":80,"ToPort":80,"IpProtocol":"tcp","IpRanges":[{"CidrIp":"4.240.39.192/32"}]}'
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0272cad83c364d115",
            "GroupId": "sg-0ae675cd27298fd47",
            "GroupOwnerId": "791666871664",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "4.240.39.192/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:eu-north-1:791666871664:security-group-rule/sgr-0272cad83c364d115"
        }
    ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-0ae675cd27298fd47
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0ae675cd27298fd47",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0be30d1beb39c1848",
            "SecurityGroupArn": "arn:aws:ec2:eu-north-1:791666871664:security-group/sg-0ae675cd27298fd47",
            "OwnerId": "791666871664",
            "GroupName": "MySecurityGroup",
            "Description": "My Security Group",
            "IpPermissions": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "4.240.39.192/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "4.240.39.192/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
```

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

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 create-key-pair \
>   --key-name MyED25519Key \
>   --key-type ed25519 \
>   --key-format pem \
>   --query 'KeyMaterial' \
>   --output text > MyED25519Key.pem
@SeratFatima00 /workspaces/Lab9 (main) $ ls -l MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Dec 27 07:02 MyED25519Key.pem
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-key-pairs
{
    "KeyPairs": [
        {
            "KeyPairId": "key-0b3f199fe355e2816",
            "KeyType": "ed25519",
            "Tags": [],
            "CreateTime": "2025-12-27T07:02:07.931000+00:00",
            "KeyName": "MyED25519Key",
            "KeyFingerprint": "t2ubcAs0To3aG6X0lWkjVDW/lBucB0tHp42L7lrN9fM="
        }
    ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 delete-key-pair --key-name MyED25519Key
{
    "Return": true,
    "KeyPairId": "key-0b3f199fe355e2816"
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 run-instances \
>   --image-id ami-00c4bcf1a0fe68ee2 \
>   --count 1 \
>   --instance-type t3.micro \
>   --key-name MyED25519Key \
>   --security-group-ids sg-0ae675cd27298fd47 \
>   --subnet-id subnet-01bdd76844f20e481 \
>   --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=MyServer}]"
{
    "ReservationId": "r-00a0693c2f3a10503",
    "OwnerId": "791666871664",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "e2d76102-9429-46c4-89d2-ff3301685cee",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-12-27T07:28:50+00:00",
                        "AttachmentId": "eni-attach-08e3821e30b4aea36",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-0ae675cd27298fd47",
                            "GroupName": "MySecurityGroup"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "0e:85:24:4b:e2:3f",
                    "NetworkInterfaceId": "eni-02c7fbfa326c6b9db",
                    "OwnerId": "791666871664",
                    "PrivateDnsName": "ip-172-31-1-226.eu-north-1.compute.internal",
                    "PrivateIpAddress": "172.31.1.226",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-172-31-1-226.eu-north-1.compute.internal",
                            "PrivateIpAddress": "172.31.1.226"
                        }
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ ssh -i MyED25519Key.pem ec2-user@16.16.156.0
The authenticity of host '16.16.156.0 (16.16.156.0)' can't be established.
ED25519 key fingerprint is SHA256:iMkZWo3JgzNP0X0R1Ddk469+R7LGRGkK0iiUaXF4/dA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '16.16.156.0' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2026-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-1-226 ~]$
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 stop-instances --instance-ids i-0e94ca8f6177d20fb --region eu-north-1
{
    "StoppingInstances": [
        {
            "InstanceId": "i-0e94ca8f6177d20fb",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 start-instances --instance-ids i-0e94ca8f6177d20fb --region eu-north-1
{
    "StartingInstances": [
        {
            "InstanceId": "i-0e94ca8f6177d20fb",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

Task 5 — Understand AWS describe-* commands

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-security-groups
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0fa54a21f7e5d9927",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0be30d1beb39c1848",
            "SecurityGroupArn": "arn:aws:ec2:eu-north-1:791666871664:security-group/sg-0fa54a21f7e5d9927",
            "OwnerId": "791666871664",
            "GroupName": "default",
            "Description": "default VPC security group",
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [
                        {
                            "UserId": "791666871664",
                            "GroupId": "sg-0fa54a21f7e5d9927"
                        }
                    ],
                    "IpRanges": [],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ]
        },
        {
            "GroupId": "sg-0ae675cd27298fd47",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
```


@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-vpcs
{
"Vpcs": [
{
"OwnerId": "791666871664",
"InstanceTenancy": "default",
"CidrBlockAssociationSet": [
{
"AssociationId": "vpc-cidr-assoc-04f111177fd2b5bce",
"CidrBlock": "172.31.0.0/16",
"CidrBlockState": {
"State": "associated"
}
}
],
"IsDefault": true,
"BlockPublicAccessStates": {
"InternetGatewayBlockMode": "off"
},
"VpcId": "vpc-0be30d1beb39c1848",
"State": "available",
"CidrBlock": "172.31.0.0/16",
"DhcpOptionsId": "dopt-0b0d23efc3a45451c"
}
]
}

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-subnets
{
    "Subnets": [
        {
            "AvailabilityZoneId": "eun1-az2",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "791666871664",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:eu-north-1:791666871664:subnet/subnet-0c5da435ab3e5b46b",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0c5da435ab3e5b46b",
            "State": "available",
            "VpcId": "vpc-0be30d1beb39c1848",
            "CidrBlock": "172.31.32.0/20",
            "AvailableIpAddressCount": 4091,
            "AvailabilityZone": "eu-north-1b",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true
        },
        {
            "AvailabilityZoneId": "eun1-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "791666871664",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:eu-north-1:791666871664:subnet/subnet-0bdbb334ba9665dba",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0bdbb334ba9665dba",
            "State": "available",
            "VpcId": "vpc-0be30d1beb39c1848",
            "CidrBlock": "172.31.16.0/20",
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-instances
{
    "Reservations": [
        {
            "ReservationId": "r-00a0693c2f3a10503",
            "OwnerId": "791666871664",
            "Groups": [],
            "Instances": [
                {
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2025-12-27T07:28:51+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-0e44b5588a7ea8f76"
                            }
                        }
                    ],
                    "ClientToken": "e2d76102-9429-46c4-89d2-ff3301685cee",
                    "EbsOptimized": false,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "NetworkInterfaces": [
                        {
                            "Association": {
                                "IpOwnerId": "amazon",
                                "PublicDnsName": "ec2-13-50-99-27.eu-north-1.compute.amazonaws.com",
                                "PublicIp": "13.50.99.27"
                            },
                            "Attachment": {
                                "AttachTime": "2025-12-27T07:28:50+00:00",
                                "AttachmentId": "eni-attach-08e3821e30b4aea36",
                                "DeleteOnTermination": true,
                                "DeviceIndex": 0,
                                "Status": "attached",
                                "NetworkCardIndex": 0
                            },
                            "Description": "",
                            "Groups": [
                                {
                                    "GroupId": "sg-0ae675cd27298fd47",
                                    "GroupName": "MySecurityGroup"
                                }
                            ],
                            "Ipv6Addresses": [],
                            "MacAddress": "0e:85:24:4b:e2:3f",
                            "NetworkInterfaceId": "eni-02c7fbfa326c6b9db",
```

```
@SeratFatima00 /workspaces/Lab9 (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"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "eu-west-3",
            "Endpoint": "ec2.eu-west-3.amazonaws.com"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "eu-west-2",
            "Endpoint": "ec2.eu-west-2.amazonaws.com"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "eu-west-1",
            "Endpoint": "ec2.eu-west-1.amazonaws.com"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "ap-northeast-3",
            "Endpoint": "ec2.ap-northeast-3.amazonaws.com"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "ap-northeast-2",
            "Endpoint": "ec2.ap-northeast-2.amazonaws.com"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "ap-northeast-1",
            "Endpoint": "ec2.ap-northeast-1.amazonaws.com"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "RegionName": "ca-central-1",
            "Endpoint": "ec2.ca-central-1.amazonaws.com"
        },
```

```
@SeratFatima00 /workspaces/Lab9 (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",
            "State": "available"
        },
        {
            "OptInStatus": "opt-in-not-required",
            "Messages": [],
            "RegionName": "eu-north-1",
            "ZoneName": "eu-north-1c",
            "ZoneId": "eun1-az3",
            "GroupName": "eu-north-1-zg-1",
            "NetworkBorderGroup": "eu-north-1",
            "ZoneType": "availability-zone",
            "GroupLongName": "Europe (Stockholm) 1",
            "State": "available"
        }
    ]
}
```

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

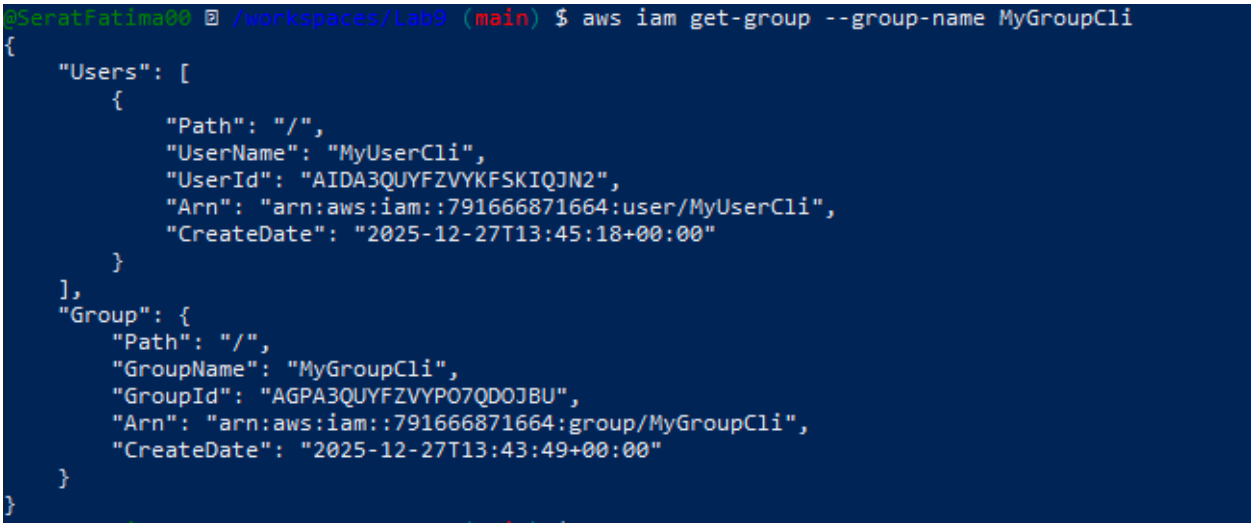
```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam create-group --group-name MyGroupCli
{
    "Group": {
        "Path": "/",
        "GroupName": "MyGroupCli",
        "GroupId": "AGPA3QUYFZVYPO7QDOJBU",
        "Arn": "arn:aws:iam::791666871664:group/MyGroupCli",
        "CreateDate": "2025-12-27T13:43:49+00:00"
    }
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam get-group --group-name MyGroupCli
{
    "Users": [],
    "Group": {
        "Path": "/",
        "GroupName": "MyGroupCli",
        "GroupId": "AGPA3QUYFZVYPO7QDOJBU",
        "Arn": "arn:aws:iam::791666871664:group/MyGroupCli",
        "CreateDate": "2025-12-27T13:43:49+00:00"
    }
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam create-user --user-name MyUserCli
{
    "User": {
        "Path": "/",
        "UserName": "MyUserCli",
        "UserId": "AIDA3QUYFZVYKFSKIQJN2",
        "Arn": "arn:aws:iam::791666871664:user/MyUserCli",
        "CreateDate": "2025-12-27T13:45:18+00:00"
    }
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam get-user --user-name MyUserCli
{
    "User": {
        "Path": "/",
        "UserName": "MyUserCli",
        "UserId": "AIDA3QUYFZVYKFSKIQJN2",
        "Arn": "arn:aws:iam::791666871664:user/MyUserCli",
        "CreateDate": "2025-12-27T13:45:18+00:00"
    }
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam add-user-to-group \
>   --user-name MyUserCli \
>   --group-name MyGroupCli
```


@SeratFatima00 /workspaces/Lab9 (main) $ aws iam get-group --group-name MyGroupCli
{
"Users": [
{
"Path": "/",
"UserName": "MyUserCli",
"UserId": "AIDA3QUYFZVYKFSKIQJN2",
"Arn": "arn:aws:iam::791666871664:user/MyUserCli",
"CreateDate": "2025-12-27T13:45:18+00:00"
}
],
"Group": {
"Path": "/",
"GroupName": "MyGroupCli",
"GroupId": "AGPA3QUYFZVYPO7QDOJBU",
"Arn": "arn:aws:iam::791666871664:group/MyGroupCli",
"CreateDate": "2025-12-27T13:43:49+00:00"
}
}

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam list-policies \
>   --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
AWSEC2SpotServiceRolePolicy
AWSServiceRoleForEC2ScheduledInstances
AWSEC2SpotFleetServiceRolePolicy
AWSApplicationAutoscalingEC2SpotFleetRequestPolicy
AWSEC2FleetServiceRolePolicy
AWSAutoScalingPlansEC2AutoScalingPolicy
EC2InstanceConnect
AmazonEC2RolePolicyForLaunchWizard
EC2InstanceProfileForImageBuilder
EC2FleetTimeShiftableServiceRolePolicy
AmazonEC2RoleforAWSCodeDeployLimited
EC2InstanceProfileForImageBuilderECRContainerBuilds
AWSApplicationMigrationEC2Access
AWSEC2CapacityReservationFleetRolePolicy
EC2FastLaunchServiceRolePolicy
AmazonSSMManagedEC2InstanceDefaultPolicy
AWSFaultInjectionSimulatorEC2Access
EC2ImageBuilderLifecycleExecutionPolicy
AWSEC2VssSnapshotPolicy
EC2FastLaunchFullAccess
AmazonEC2ContainerRegistryPullOnly
DeclarativePoliciesEC2Report
AmazonEC2ImageReferencesAccessPolicy
AWSEC2CapacityManagerServiceRolePolicy
AWSEC2SqlHaServiceRolePolicy
AWSEC2SqlHaInstancePolicy
AWSLambdaManagedEC2ResourceOperator
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam list-policies \
>   --query "Policies[?PolicyName=='AmazonEC2FullAccess'].{Name:PolicyName, ARN:Arn}" \
>   --output table
-------------------------------------------------------------------------
|                            ListPolicies                               |
+-----------------------------------------------+-----------------------+
|                      ARN                      |         Name          |
+-----------------------------------------------+-----------------------+
|  arn:aws:iam::aws:policy/AmazonEC2FullAccess  |  AmazonEC2FullAccess  |
+-----------------------------------------------+-----------------------+
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam attach-group-policy \
>   --group-name MyGroupCli \
>   --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam list-attached-group-policies --group-name MyGroupCli
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonEC2FullAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
        }
    ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam create-login-profile \
>    --user-name MyUserCli \
>    --password T[redacted]
>    --password-reset-required
{
    "LoginProfile": {
        "UserName": "MyUserCli",
        "CreateDate": "2025-12-27T13:57:06+00:00",
        "PasswordResetRequired": true
    }
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam attach-group-policy \
>    --group-name MyGroupCli \
>    --policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam detach-group-policy \
>    --group-name MyGroupCli \
>    --policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam create-access-key --user-name MyUserCli
{
    "AccessKey": {
        "UserName": "MyUserCli",
        "AccessKeyId": "A[redacted]
        "Status": "Active",
        "SecretAccessKey": "[redacted]",
        "CreateDate": "2025-12-27T14:14:35+00:00"
    }
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam list-access-keys --user-name MyUserCli
{
    "AccessKeyMetadata": [
        {
            "UserName": "MyUserCli",
            "AccessKeyId": "A[redacted]
            "Status": "Active",
            "CreateDate": "2025-12-27T14:14:35+00:00"
        }
    ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ export AWS_ACCESS_KEY_ID=AKIA3QUYFZVYPB354WM4
@SeratFatima00 /workspaces/Lab9 (main) $ export AWS_SECRET_ACCESS_KEY=xxxxxxxxxxxxxxx
@SeratFatima00 /workspaces/Lab9 (main) $ export AWS_SECRET_ACCESS_KEY=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
@SeratFatima00 /workspaces/Lab9 (main) $ printenv | grep AWS_
AWS_SECRET_ACCESS_KEY=ne3o1P4FOHbGCO4IxtzXWJxAieVbNYHy4ovSoa9B
AWS_ACCESS_KEY_ID=AKIA3QUYFZVYPB354WM4
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam get-user --user-name MyUserCli

An error occurred (AccessDenied) when calling the GetUser operation: User: arn:aws:iam::791666871664:user/MyUserCli is not authorized to perform: iam:GetUser on resource: user MyUserCli becaus
e no identity-based policy allows the iam:GetUser action
@SeratFatima00 /workspaces/Lab9 (main) $ exit
logout
Connection to localhost closed.
shell closed: exit status 254
```

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

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-instances \
>   --filters "Name=tag:Name,Values=MyServer" \
>   --query "Reservations[*].Instances[*].PublicIpAddress" \
>   --output text
13.50.99.27
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-instances \
>   --filters "Name=instance-type,Values=t3.micro" \
>   --query "Reservations[].Instances[].InstanceId" \
>   --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-0e94ca8f6177d20fb  |
+-----------------------+
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-instances \
>   --filters "Name=subnet-id,Values=subnet-01bdd76844f20e481" \
>   --query "Reservations[*].Instances[*].InstanceId" \
>   --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-0e94ca8f6177d20fb  |
+-----------------------+
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-instances \
>   --filters "Name=vpc-id,Values=vpc-0be30d1beb39c1848" \
>   --query "Reservations[*].Instances[*].InstanceId" \
>   --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-0e94ca8f6177d20fb  |
+-----------------------+
```

Task 8 — Use --query to format outputs for reporting

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-instances \
>   --filters "Name=tag:Name,Values=MyServer" \
>   --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress,Tags[?Key=='Name'].Value|[0]]" \
>   --output table
-----------------------------------------------------
|                DescribeInstances                  |
+----------------------+---------------+------------+
|  i-0e94ca8f6177d20fb |  13.50.99.27  |  MyServer  |
+----------------------+---------------+------------+
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-instances \
>   --query "Reservations[*].Instances[*].[InstanceId,State.Name]" \
>   --output table
-----------------------------------
|         DescribeInstances       |
+----------------------+----------+
|  i-0e94ca8f6177d20fb |  running |
+----------------------+----------+
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-instances \
>   --query "Reservations[*].Instances[*].[InstanceId,InstanceType,Placement.AvailabilityZone]" \
>   --output table
-----------------------------------------------------
|                DescribeInstances                  |
+----------------------+-----------+---------------+
|  i-0e94ca8f6177d20fb |  t3.micro |  eu-north-1c  |
+----------------------+-----------+---------------+
```

Cleanup — Remove resources to avoid charges

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 terminate-instances --instance-ids i-0e94ca8f6177d20fb
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-0e94ca8f6177d20fb",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-volumes --output table
----------------
|DescribeVolumes|
+---------------+
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-snapshots --owner-ids self --output table
------------------
|DescribeSnapshots|
+-----------------+
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 delete-security-group --group-id sg-0ae675cd27298fd47
aws ec2 delete-security-group --group-id sg-0ae675cd27298fd47
{
    "Return": true,
    "GroupId": "sg-0ae675cd27298fd47"
}
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 delete-security-group --group-id sg-0ae675cd27298fd47

An error occurred (InvalidGroup.NotFound) when calling the DeleteSecurityGroup operation: The security group 'sg-0ae675cd27298fd47' does not exist
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 delete-key-pair --key-name MyED25519Key
{
    "Return": true,
    "KeyPairId": "key-0138827f692289477"
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam delete-access-key \
>   --user-name MyUserCli \
>   --access-key-id AKIA3QUYFZVYPB354WM4
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam delete-login-profile --user-name MyUserCli
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam remove-user-from-group \
>   --user-name MyUserCli \
>   --group-name MyGroupCli
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam detach-group-policy \
>   --group-name MyGroupCli \
>   --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam delete-user --user-name MyUserCli
@SeratFatima00 /workspaces/Lab9 (main) $ aws iam delete-group --group-name MyGroupCli
```