



CLOUD COMPUTING LAB
BSE (V-B)

LAB 09

Submitted By: Urwa Zahra

Roll No: 2023-BSE-068

Submitted To: Sir Muhammad Shoaib

Task 1 — GitHub CLI, Codespace setup and authentication

Objective: Install GH CLI, authenticate for Codespaces, and create/connect to a Codespace.

- Install GitHub CLI

```
PS D:\> 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
```

- GH CLI authentication for Codespaces

```
PS C:\Users\TT> 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 Urwa012
PS C:\Users\TT>
```

- Connect to Codespace

```
BSE-068 (main) $ whoami
pwd
codespace
/workspaces/cc-urwazahra-2023-BSE-068
```

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

Objective: Install AWS CLI, configure it, and verify connectivity.

- AWS CLI install and version check

```
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/METADATA
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/WHEEL
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/licenses/AUTHORS.rst
inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/LICENSE
inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
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/REQUESTED
inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
u can now run: /usr/local/bin/aws --version
rwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $
```

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws --version
aws-cli/2.32.28 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $

```

- AWS configure credentials

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws configure
AWS Access Key ID [None]: [REDACTED]
AWS Secret Access Key [None]: [REDACTED]
Default region name [None]: me-central-1
Default output format [None]: json
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ cat ~/.aws/credentials
[default]
aws_access_key_id = [REDACTED]
aws_secret_access_key = [REDACTED]
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ cat ~/.aws/config
[default]
region = me-central-1
output = json

```

- Verify connectivity (aws sts get-caller-identity)

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws sts get-caller-identity
{
  "UserId": "AIDAWOW3ND6CPIGJXS7MS",
  "Account": "443915509636",
  "Arn": "arn:aws:iam::443915509636:user/urwa"
}
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $

```

Task 3 — Security group creation & ingress rules

Objective: Create and configure EC2 security group with SSH/HTTP rules.

- Create security group

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 create-security-group --group-name 'MySecurityGroup' --description 'My Security Group'
{
  "GroupId": "sg-0cb2571c3eef2e0a9",
  "SecurityGroupArn": "arn:aws:ec2:me-central-1:443915509636:security-group/sg-0cb2571c3eef2e0a9"
}

```

- Describe SG before ingress

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-security-groups --group-ids sg-0cb2571c3eef2e0a9
{
  "SecurityGroups": [
    {
      "GroupId": "sg-0cb2571c3eef2e0a9",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": []
        }
      ],
      "VpcId": "vpc-0b7740d2d5f400590",
      "SecurityGroupArn": "arn:aws:ec2:me-central-1:443915509636:security-group/sg-0cb2571c3eef2e0a9",
      "OwnerId": "443915509636",
      "GroupName": "MySecurityGroup",
      "Description": "My Security Group",
      "IpPermissions": []
    }
  ]
}
```

- Codespace public IP

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ curl icanhazip.com
4.240.18.226
```

- Authorize SSH & HTTP rules

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 authorize-security-group-ingress \
--group-id sg-0cb2571c3eef2e0a9 \
--protocol tcp \
--port 22 \
--cidr 123.45.67.89/32
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0e9291b2b870033e2",
      "GroupId": "sg-0cb2571c3eef2e0a9",
      "GroupOwnerId": "443915509636",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "CidrIpv4": "123.45.67.89/32",
      "SecurityGroupRuleArn": "arn:aws:ec2:me-central-1:443915509636:security-group-rule/sgr-0e9291b2b870033e2"
    }
  ]
}
```

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-security-groups --group-ids sg-0cb2571c3eef2e0a9
{
  "IpPermissions": [
    {
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "UserIdGroupPairs": [],
      "IpRanges": [
        {
          "CidrIp": "123.45.67.89/32"
        }
      ],
      "Ipv6Ranges": []
    }
  ],
  "SecurityGroups": [
    {
      "GroupId": "sg-0cb2571c3eef2e0a9",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": []
        }
      ],
      "VpcId": "vpc-0b7740d2d5f400590",
      "SecurityGroupArn": "arn:aws:ec2:me-central-1:443915509636:security-group/sg-0cb2571c3eef2e0a9",
      "OwnerId": "443915509636",
      "GroupName": "MySecurityGroup",
      "Description": "My Security Group",
      "IpPermissions": [
        {
          "IpProtocol": "tcp",
          "FromPort": 22,
          "ToPort": 22,
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "123.45.67.89/32"
            }
          ],
          "Ipv6Ranges": []
        }
      ]
    }
  ]
}
```

```
● @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 authorize-security-group-ingress --group-id sg-0cb2571c3eef2e0a9 --ip-permissions '{"FromPort":80,"ToPort":80,"IpProtocol":"tcp","IpRanges":[{"CidrIp":"123.45.67.89/32"}]}'
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-046fa20a4349f647a",
      "GroupId": "sg-0cb2571c3eef2e0a9",
      "GroupOwnerId": "443915509636",
      "IsEgress": false,
      "IpProtocol": "tcp",
      "FromPort": 80,
      "ToPort": 80,
      "CidrIpv4": "123.45.67.89/32",
      "SecurityGroupRuleArn": "arn:aws:ec2:me-central-1:443915509636:security-group-rule/sgr-046fa20a4349f647a"
    }
  ]
}
```

- Final SG verification

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-security-groups --group-ids sg-0cb2571c3eef2e0a9
{
  "GroupId": "sg-0cb2571c3eef2e0a9",
  "IpPermissionsEgress": [
    {
      "IpProtocol": "-1",
      "UserIdGroupPairs": [],
      "IpRanges": [
        {
          "CidrIp": "0.0.0.0/0"
        }
      ],
      "Ipv6Ranges": [],
      "PrefixListIds": []
    }
  ],
  "VpcId": "vpc-0b7740d2d5f400590",
  "SecurityGroupArn": "arn:aws:ec2:me-central-1:443915509636:security-group/sg-0cb2571c3eef2e0a9",
  "OwnerId": "443915509636",
  "GroupName": "MySecurityGroup",
  "Description": "My Security Group",
  "IpPermissions": [
    {
      "IpProtocol": "tcp",
      "FromPort": 80,
      "ToPort": 80,
      "UserIdGroupPairs": [],
      "IpRanges": [
        {
          "CidrIp": "123.45.67.89/32"
        }
      ],
      "Ipv6Ranges": [],
      "PrefixListIds": []
    },
    {
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "UserIdGroupPairs": [],
      "IpRanges": [
        {
          "CidrIp": "123.45.67.89/32"
        }
      ],
      "Ipv6Ranges": [],
      "PrefixListIds": []
    }
  ]
}
```

Task 4 — Key pair creation, EC2 launch & SSH

Objective: Create key pair, launch EC2 instance, SSH access, and manage instance state.

- Create key pair

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 create-key-pair \
  --key-name MyED25519Key \
  --key-type ed25519 \
  --key-format pem \
  --query 'KeyMaterial' \
  --output text > MyED25519Key.pem
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ ls -l MyED25519Key.pem
ls -l MyED25519Key.pem
ls -l MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Jan  3 21:43 MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Jan  3 21:43 MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Jan  3 21:43 MyED25519Key.pem
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $
```

- Describe key pairs

```

    "PrefixListIds": []
  }
}

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 create-key-pair \
--key-name MyED25519Key \
--key-type ed25519 \
--key-format pem \
--query 'KeyMaterial' \
--output text > MyED25519Key.pem
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ ls -l MyED25519Key.pem
ls -l MyED25519Key.pem
ls -l MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Jan  3 21:43 MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Jan  3 21:43 MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Jan  3 21:43 MyED25519Key.pem
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $

```

- Delete key pair (optional)

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 delete-key-pair --key-name MyED25519Key
{
  "Return": true,
  "KeyPairId": "key-0eeb900804707e4d9"
}
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $

```

- Launch EC2 instance

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 run-instances \
--image-id ami-05e66df2bafcb7dea \
--count 1 \
--instance-type t3.micro \
--key-name MyED25519Key \
--security-group-ids sg-0cb2571c3eef2e0a9 \
--subnet-id subnet-0465fd933750965e \
--tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=MyServer}]"
{
  "ReservationId": "r-092ae90704e1866d8",
  "OwnerId": "443915509636",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "adb409b9-8294-4ec7-b3fa-3cd8c3d338b4",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2026-01-03T22:05:42+00:00",
            "AttachmentId": "eni-attach-0689d0b15e78ce3c3",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "Status": "attaching",
            "NetworkCardIndex": 0
          },
          "Description": "",
          "Groups": [
            {
              "GroupId": "sg-0cb2571c3eef2e0a9",
              "GroupName": "MySecurityGroup"
            }
          ],
          "Ipv6Addresses": [],
          "MacAddress": "0a:d7:1c:c0:93:5f",
          "NetworkInterfaceId": "eni-0597da10f1b3987e4",
          "OwnerId": "443915509636",
          "PrivateDnsName": "ip-172-31-16-201.me-central-1.compute.internal",
          "PrivateIpAddress": "172.31.16.201",
          "PrivateIpAddresses": [
            {
              "Primary": true,
              "PrivateDnsName": "ip-172-31-16-201.me-central-1.compute.internal",
              "PrivateIpAddress": "172.31.16.201"
            }
          ]
        }
      ]
    }
  ]
}

```

- Describe instance & public IP

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances \
--query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress]" \
--output table
-----
| DescribeInstances |
+-----+-----+
| i-04a7037e582a6e914 | 3.28.47.243 |
+-----+-----+
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ chmod 400 MyED25519Key.pem

```

- SSH permission error & fix

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ ssh -i MyED25519Key.pem ec2-user@3.28.47.243
The authenticity of host '3.28.47.243 (3.28.47.243)' can't be established.
ED25519 key fingerprint is SHA256:zGRGN7+QqdFwyv+g6RrhbPH2TjT2sP6vmOxnpgGrfd8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '3.28.47.243' (ED25519) to the list of known hosts.
```

- Stop/Start/Terminate instance

```
Connection to 3.28.47.243 closed.
```

```
[root@bz012 ~]# aws ecs stop-instances --instance-ids i-04a7037e582ade914
```

```
{
  "StoppingInstances": [
    {
      "InstanceId": "i-04a7037e582ade914",
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

```
[root@bz012 ~]# aws ecs start-instances --instance-ids i-04a7037e582ade914
```

```
An error occurred (IncorrectInstanceState) when calling the StartInstances operation: The instance 'i-04a7037e582ade914' is in a state from which it cannot be started.
```

```
[root@bz012 ~]# aws ecs terminate-instances --instance-ids i-04a7037e582ade914
```

```
{
  "terminatingInstances": [
    {
      "InstanceId": "i-04a7037e582ade914",
      "CurrentState": {
        "Code": 48,
        "Name": "terminated"
      },
      "PreviousState": {
        "Code": 64,
        "Name": "stopped"
      }
    }
  ]
}
```

Task 5 — AWS describe- commands

Objective: Inspect AWS resources.

- `aws ec2 describe-security-groups`

```

@Urwa012 → /workspaces/cc-urwazhara-2023-BSE-068 (main) $ aws ec2 describe-security-groups
{
  "SecurityGroups": [
    {
      "GroupId": "sg-0cb2571c3eef2e0a9",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": []
        }
      ],
      "VpcId": "vpc-0b7740d2d5f400590",
      "SecurityGroupArn": "arn:aws:ec2:me-central-1:443915509636:security-group/sg-0cb2571c3eef2e0a9"
    }
  ]
}

```

- `aws ec2 describe-vpcs:`


```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-vpcs
{
  "Vpcs": [
    {
      "OwnerId": "443915509636",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-044f7e148a3d80b3d",
          "CidrBlock": "172.31.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ],
      "IsDefault": true,
      "BlockPublicAccessStates": {
        "InternetGatewayBlockMode": "off"
      },
      "VpcId": "vpc-0b7740d2d5f400590",
    }
  ]
}
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $

```

- aws ec2 describe-subnets

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-subnets
{
  "Subnets": [
    {
      "AvailabilityZoneId": "mec1-az2",
      "MapCustomerOwnedIpOnLaunch": false,
      "OwnerId": "443915509636",
      "AssignIpv6AddressOnCreation": false,
      "Ipv6CidrBlockAssociationSet": [],
      "SubnetArn": "arn:aws:ec2:me-central-1:443915509636:subnet/subnet-0465fd9337550965e",
      "EnableDns64": false,
      "Ipv6Native": false,
      "PrivateDnsNameOptionsOnLaunch": {
        "HostnameType": "ip-name",
        "EnableResourceNameDnsARecord": false,
        "EnableResourceNameDnsAAAARecord": false
      },
      "BlockPublicAccessStates": {
        "InternetGatewayBlockMode": "off"
      }
    }
  ],
}

```

- aws ec2 describe-instances

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances
{
  "Reservations": []
}

```

- aws ec2 describe-regions

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (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",

```

- aws ec2 describe-availability-zones

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-availability-zones
{
  "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",

```

Task 6 — IAM: Group, User, Policies, Keys

Objective: Create IAM users/groups, attach policies, and configure keys.

- Create group & user

```

○ @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $
● @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam create-group --group-name MyGroupCli
{
  "Group": {
    "Path": "/",
    "GroupName": "MyGroupCli",
    "GroupId": "AGPAWOW3ND6CP3M37JL3S",
    "Arn": "arn:aws:iam::443915509636:group/MyGroupCli",
    "CreateDate": "2026-01-04T10:19:28+00:00"
  }
}
● @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam get-group --group-name MyGroupCli
{
  "Users": [],
  "Group": {
    "Path": "/",
    "GroupName": "MyGroupCli",
    "GroupId": "AGPAWOW3ND6CP3M37JL3S",
    "Arn": "arn:aws:iam::443915509636:group/MyGroupCli",
    "CreateDate": "2026-01-04T10:19:28+00:00"
  }
}
● @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam create-user --user-name MyUserCli
{
  "User": {
    "Path": "/",
    "UserName": "MyUserCli",
    "UserId": "AIDAWOW3ND6CBCVDTRITW",
    "Arn": "arn:aws:iam::443915509636:user/MyUserCli",
    "CreateDate": "2026-01-04T10:19:49+00:00"
  }
}
● @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam get-user --user-name MyUserCli
{
  "User": {
    "Path": "/",
    "UserName": "MyUserCli",
    "UserId": "AIDAWOW3ND6CBCVDTRITW",
    "Arn": "arn:aws:iam::443915509636:user/MyUserCli",
    "CreateDate": "2026-01-04T10:19:49+00:00"
  }
}
● @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam add-user-to-group --user-name MyUserCli --group-name MyGroupCli
● @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam get-group --group-name MyGroupCli
{
  "Users": [
    {
      "Path": "/",
      "UserName": "MyUserCli",
      "UserId": "AIDAWOW3ND6CBCVDTRITW",
      "Arn": "arn:aws:iam::443915509636:user/MyUserCli",
      "CreateDate": "2026-01-04T10:19:49+00:00"
    }
  ],
  "Group": {
    "Path": "/",

```

- Attach policy to group

```

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam list-policies \
--query "Policies[?contains(PolicyName, 'EC2')].{Name:PolicyName}" \
--output text
AmazonEC2ContainerServiceRole
AmazonEC2RoleforAWSCodeDeploy
AmazonEC2RoleforSSM
CloudWatchActionsEC2Access
AmazonEC2ContainerRegistryReadOnly
AmazonEC2ContainerRegistryPowerUser
AmazonEC2ContainerRegistryFullAccess
AmazonEC2ContainerServiceAutoscaleRole
AmazonEC2SpotFleetAutoscaleRole
AWSElasticBeanstalkCustomPlatformForEC2Role
AmazonEC2ContainerServiceEventsRole
AmazonEC2SpotFleetTaggingRole
AWSEC2SpotServiceRolePolicy
AWSServiceRoleForEC2ScheduledInstances
AWSEC2SpotFleetServiceRolePolicy
AWSApplicationAutoscalingEC2SpotFleetRequestPolicy
AWSEC2FleetServiceRolePolicy
AWSAutoScalingPlansEC2AutoScalingPolicy
EC2InstanceConnect
AmazonEC2RolePolicyForLaunchWizard
EC2InstanceProfileForImageBuilder
EC2FleetTimeShiftTableServiceRolePolicy
AmazonEC2RoleforAWSCodeDeployLimited
EC2InstanceProfileForImageBuilderECRContainerBuilds
AWSApplicationMigrationEC2Access
AWSEC2CapacityReservationFleetRolePolicy
EC2FastLaunchServiceRolePolicy
AmazonSSMManagedEC2InstanceDefaultPolicy
AWSFaultInjectionSimulatorEC2Access
EC2ImageBuilderLifecycleExecutionPolicy
AWSEC2VssSnapshotPolicy
EC2FastLaunchFullAccess
AmazonEC2ContainerRegistryPullOnly
DeclarativePoliciesEC2Report
AmazonEC2ImageReferencesAccessPolicy
AWSEC2CapacityManagerServiceRolePolicy
AWSEC2SqlHaServiceRolePolicy
AWSEC2SqlHaInstancePolicy
AWSLambdaManagedEC2ResourceOperator
• @Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam list-policies --query 'Policies[?PolicyName==`AmazonEC2FullAccess`].{Name:PolicyName}' --output table
-----
|                               ListPolicies                               |
+-----+-----+-----+-----+
|                               ARN                               | Name |
+-----+-----+-----+-----+
| arn:aws:iam::aws:policy/AmazonEC2FullAccess | AmazonEC2FullAccess |
+-----+-----+-----+-----+

@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam attach-group-policy --group-name MyGroupCli --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam list-attached-group-policies --group-name MyGroupCli
{
  "AttachedPolicies": [
    {
      "PolicyName": "AmazonEC2FullAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
    }
  ]
}
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $

```

- Create login profile

```
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam create-login-profile \
--user-name MyUserCli \
--password '...' \
--password-reset-required
{
  "LoginProfile": {
    "UserName": "MyUserCli",
    "CreateDate": "2026-01-04T10:46:38+00:00",
    "PasswordResetRequired": true
  }
}
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam attach-group-policy --group-name MyGroupCli --policy-arn arn:aws:iam::aws:policy/IAMUse
d
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam detach-group-policy --group-name MyGroupCli --policy-arn arn:aws:iam::aws:policy/IAMUse
d
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ []
```

- Create access key

```
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam create-access-key --user-name MyUserCli
{
  "AccessKey": {
    "UserName": "MyUserCli",
    "AccessKeyId": "...",
    "Status": "Active",
    "SecretAccessKey": "...",
    "CreateDate": "2026-01-04T10:48:58+00:00"
  }
}
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam list-access-keys --user-name MyUserCli
{
  "AccessKeyMetadata": [
    {
      "UserName": "MyUserCli",
      "AccessKeyId": "...",
      "Status": "Active",
      "CreateDate": "2026-01-04T10:48:58+00:00"
    }
  ]
}
```

- Test environment variable authentication

```
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ export AWS_ACCESS_KEY_ID=AI...
export AWS_SECRET_ACCESS_KEY=yT...
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ printenv | grep AWS_
AWS_SECRET_ACCESS_KEY=yT...
AWS_ACCESS_KEY_ID=AI...
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws iam get-user --user-name MyUserCli

An error occurred (AccessDenied) when calling the GetUser operation: User: arn:aws:iam::443915509636:user/MyUserCli is not authorized to perform: iam:GetUser
ce: user MyUserCli because no identity-based policy allows the iam:GetUser action
@Urwa012 →/workspaces/cc-urwazahra-2023-BSE-068 (main) $ []
```

```
PS C:\Users\TT> aws sts get-caller-identity
{
  "UserId": "AIDAWOW3ND6CMDWMCP3CQ",
  "Account": "443915509636",
  "Arn": "arn:aws:iam::443915509636:user/assignment"
}
PS C:\Users\TT>
```

Task 7 — Filters: Describe-instances with filters

- Filter by Tag

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances \
--filters "Name=tag:Name,Values=MyServer" \
--query "Reservations[*].Instances[*].PublicIpAddress" \
--output text
3.28.125.18
3.29.7.59
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances \
```

- Filter by Instance type

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances \
--filters "Name=instance-type,Values=t3.micro" \
--query "Reservations[].Instances[].InstanceId" \
--output table
-----
| DescribeInstances |
+-----+
| i-0506b35c74df29bb3 |
| i-02fcb3aaad45c3762 |
+-----+
```

- Filter by Subnet

```
--output table
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances --filter
s "Name=subnet-id,Values=subnet-0465fd9337550965e" --query "Reservations[*].Instances[*].Inst
anceId" --output table
-----
| DescribeInstances |
+-----+
| i-0506b35c74df29bb3 |
| i-02fcb3aaad45c3762 |
+-----+
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances \
```

- Filter by VPC

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances --filter
s "Name=vpc-id,Values=vpc-0b7740d2d5f400590" --query "Reservations[*].Instances[*].[InstanceI
d,SubnetId,State.Name]" --output table
-----
| DescribeInstances |
+-----+
| i-036adb1c913aff670 | subnet-0465fd9337550965e | running |
| i-0506b35c74df29bb3 | subnet-0465fd9337550965e | running |
| i-02fcb3aaad45c3762 | subnet-0465fd9337550965e | running |
```

Task 8 — Query outputs formatted for reporting

- Instances with Name & IP

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (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-036adb1c913aff670	51.112.215.37	MyServer
i-0506b35c74df29bb3	3.28.125.18	MyServer
i-02fcb3aaad45c3762	3.29.7.59	MyServer

- Instance ID & State

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances \
--query "Reservations[*].Instances[*].[InstanceId,State.Name]" \
--output table
```

DescribeInstances	
i-036adb1c913aff670	running
i-0506b35c74df29bb3	running
i-02fcb3aaad45c3762	running

- Instance Type & AZ

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances \
--query "Reservations[*].Instances[*].[InstanceId,InstanceType,Placement.AvailabilityZone]" \
--output table
```

DescribeInstances		
i-036adb1c913aff670	t3.micro	me-central-1b
i-0506b35c74df29bb3	t3.micro	me-central-1b
i-02fcb3aaad45c3762	t3.micro	me-central-1b

Cleanup — Remove AWS resources

- Terminate instances

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 terminate-instances \
--instance-ids i-036adb1c913aff670 i-0506b35c74df29bb3 i-02fcb3aaad45c3762
```

```
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-036adb1c913aff670",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    },
    {
      "InstanceId": "i-0506b35c74df29bb3",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    },
    {
      "InstanceId": "i-02fcb3aaad45c3762",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}
```

- Delete volumes/snapshots

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-volumes --output tab
le
+-----+
| DescribeVolumes |
+-----+
| Volumes |
+-----+
| AvailabilityZone | me-central-1b |
| AvailabilityZoneId | mecl-az2 |
| CreationTime | 2026-01-04T11:06:54.769000+00:00 |
| Encrypted | False |
| Iops | 3000 |
| MultiAttachEnabled | False |
| Size | 8 |
| SnapshotId | snap-041b493405a2d89fb |
| State | in-use |
| Throughput | 125 |
| VolumeId | vol-04c07c0b17d46803b |
| VolumeType | gp3 |
+-----+
| Attachments |
+-----+
| AttachmentId | 2026-01-04T11:06:54+00:00 |
| DeletionProtection | True |
| Device | /dev/xvda |
| InstanceId | i-0506b35c74df29bb3 |
| State | attached |
| VolumeId | vol-04c07c0b17d46803b |
+-----+
| Operator |
+-----+
| Managed | False |
+-----+
```

- Delete SG & key pair

```
+-----+
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 delete-security-group --group
-id sg-0cb2571c3eef2e0a9
{
  "Return": true,
  "GroupId": "sg-0cb2571c3eef2e0a9"
}
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 delete-key-pair --key-name My
ED25519Key
{
  "Return": true,
  "KeyId": "key-0cdd8082f756959e"
}
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $
```

- Final verification

```
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $ aws ec2 describe-instances \
--query "Reservations[*].Instances[*].[InstanceId,State.Name]" \
--output table
+-----+
| DescribeInstances |
+-----+
+-----+
| i-036adb1c913aff670 | terminated |
| i-0506b35c74df29bb3 | terminated |
| i-02fcb3aad45c3762 | terminated |
+-----+
@Urwa012 → /workspaces/cc-urwazahra-2023-BSE-068 (main) $
```