

# LAB:09

Arshia Jadoon

Bse-2023-014

## Task 1: GitHub CLI, Codespace Setup and Authentication

task1\_gh\_install.png

```
PS C:\Users\hp> 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: yes
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\hp> |
```

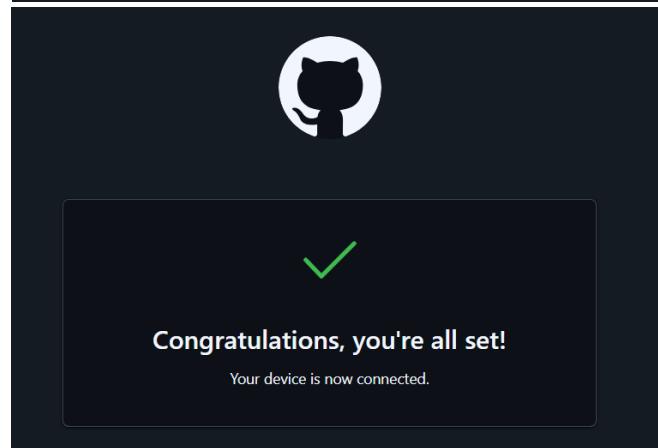
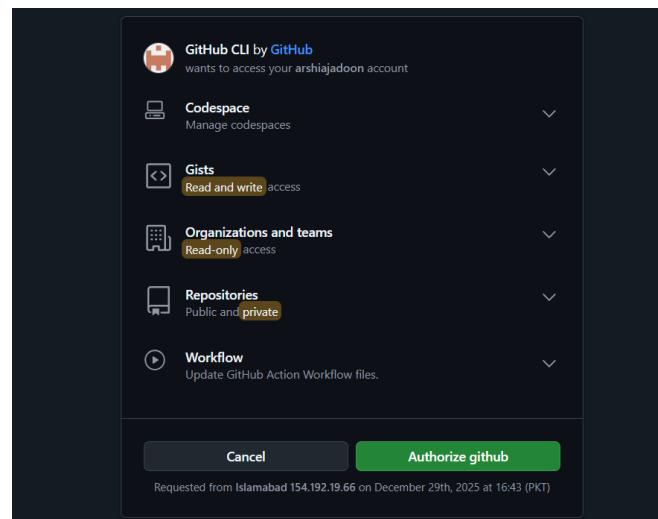
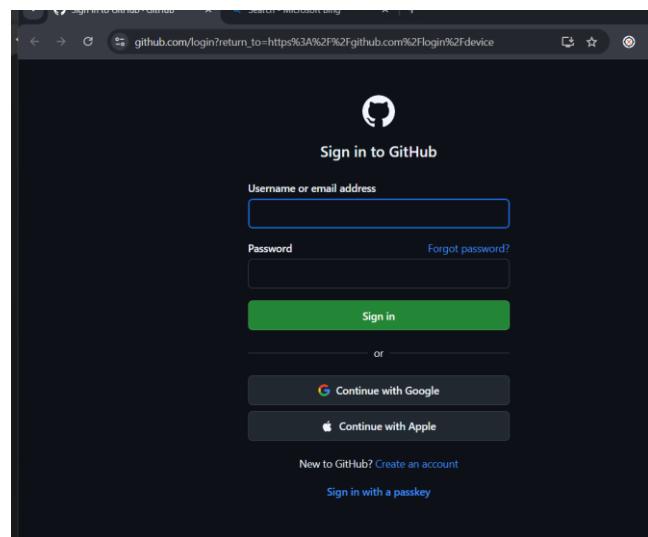
task1\_gh\_auth\_login.png

```
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hp> gh --version
gh version 2.83.2 (2025-12-10)
https://github.com/cli/cli/releases/tag/v2.83.2
PS C:\Users\hp> 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: A769-FA82
Press Enter to open https://github.com/login/device in your browser... |
```



task1\_codespace\_list.png

```
? Choose Machine Type: 2 cores, 8 GB RAM, 32 GB storage  
super-duper-space-system-pj5wr9v4x96j2rv9v  
PS C:\Users\hp\LAB_9> gh codespace list  
NAME          DISPLAY NAME      REPOSITORY    BRANCH STATE   CREATED AT  
super-duper-space-system-pj5w... super-duper space sy... arshiajadoon/LAB_9 main Available less than a minute ago  
PS C:\Users\hp\LAB_9> |
```

task1\_codespace\_ssh\_connected.png

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

@arshiajadoon → /workspaces/LAB\_9 (main) \$ |

## Task 2: Install AWS CLI Inside Codespace

task2\_aws\_install\_and\_version.png

```
Installing awscli/2.32.24...done.  
You can now run: /usr/local/bin/aws --version  
@arshiajadoon → /workspaces/LAB_9 (main) $ aws --version  
aws-cli/2.32.24 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24  
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task2\_aws\_configure\_and\_files.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws configure  
AWS Access Key ID [None]: AKIAVEW074P2FM5KLY4V  
AWS Secret Access Key [None]: wQvxcbHfeuFS+iQwmyet216MR1lsjfAWmIsRZ8J2  
Default region name [None]: us-east-1  
Default output format [None]: json  
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task2\_aws\_configure\_and\_files.png

```
cat ~/.aws/config  
[default]  
aws_access_key_id = AKIAVEW074P2FM5KLY4V  
aws_secret_access_key = wQvxcbHfeuFS+iQwmyet216MR1lsjfAWmIsRZ8J2  
[default]  
region = us-east-1  
output = json  
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task2\_aws\_get\_caller\_identity.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws sts get-caller-identity  
{  
    "UserId": "353695163380",  
    "Account": "353695163380",  
    "Arn": "arn:aws:iam::353695163380:root"  
}  
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

## Task 3: Create Security Group and Add Ingress Rules

DescribeVpcs		
vpc-0399fce17f59406f7		True

task3\_describe\_sg\_before\_ingress.png

```
{
  "SecurityGroups": [
    {
      "GroupId": "sg-01fd8e6e7dc115fb6",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": []
        }
      ],
      "VpcId": "vpc-0399fce17f59406f7",
      "SecurityGroupArn": "arn:aws:ec2:us-east-1:353695163380:security-group/MySecurityGroup",
      "OwnerId": "353695163380",
      "GroupName": "MySecurityGroup",
      "Description": "My Security Group",
      "IpPermissions": [
        {
          "IpProtocol": "tcp",
          "FromPort": 22,
          "ToPort": 22,
          "UserIdGroupPairs": [],
          "IpRanges": [
            ...
          ]
        }
      ]
    }
  ]
}
```

task3\_codespace\_public\_ip.png

```
Last login: Mon Dec 29 12:30:50 2020 from ...
@arshiajadoon → /workspaces/LAB_9 (main) $ curl icanhazip.com
20.192.21.50
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task3\_authorize\_ssh\_and\_describe.png

```
--port 22 \
--cidr 0.0.0.0/0

An error occurred (InvalidPermission.Duplicate) when calling the AuthorizeSecurityGroupIngress operation: the specified rule "peer: 0.0.0.0/0, TCP, from port: 22, to port: 22, ALLOW" already exists
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task3\_authorize\_ssh\_and\_describe.png

```
Windows PowerShell x + 
{
    "SecurityGroups": [
        {
            "GroupId": "sg-01fd8e6e7dc115fb6",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0399fce17f59406f7",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:353695163380:security-group/sg-01fd8e6e7dc115fb6",
            "OwnerId": "353695163380",
            "GroupName": "MySecurityGroup",
            "Description": "My Security Group",
            "IpPermissions": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [

```

task3\_authorize\_http\_and\_describe.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 authorize-security-group-ingress \
--group-id sg-01fd8e6e7dc115fb6 \
--ip-permissions '[{"FromPort":80,"ToPort":80,"IpProtocol":"tcp","IpRanges":[{"CidrIp":"0.0.0.0/0"}]]'
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-041ee9e4e2441f8b3",
            "GroupId": "sg-01fd8e6e7dc115fb6",
            "GroupOwnerId": "353695163380",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "0.0.0.0/0",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:353695163380:security-group-rule/sgr-041ee9e4e2441f8b3"
        }
    ]
}
```

task3\_describe\_sg\_final.png

```
{  
  "SecurityGroups": [  
    {  
      "GroupId": "sg-01fd8e6e7dc115fb6",  
      "IpPermissionsEgress": [  
        {  
          "IpProtocol": "-1",  
          "UserIdGroupPairs": [],  
          "IpRanges": [  
            {  
              "CidrIp": "0.0.0.0/0"  
            }  
          ],  
          "Ipv6Ranges": [],  
          "PrefixListIds": []  
        }  
      ],  
      "VpcId": "vpc-0399fce17f59406f7",  
      "SecurityGroupArn": "arn:aws:ec2:us-east-1:353695163380:security-group/sg-01fd8e6e7dc115fb6",  
      "OwnerId": "353695163380",  
      "GroupName": "MySecurityGroup",  
      "Description": "My Security Group",  
      "IpPermissions": [  
        {  
          "IpProtocol": "tcp",  
          "FromPort": 80,  
          "ToPort": 80,  
          "UserIdGroupPairs": [],  
          "IpRanges": [  
            {  
              "CidrIp": "0.0.0.0/0"  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

task4\_create\_keypair\_output.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 create-key-pair \
--key-name MyED25519Key \
--key-type ed25519 \
--key-format pem \
--query 'KeyMaterial' \
--output text > MyED25519Key.pem
@arshiajadoon → /workspaces/LAB_9 (main) $ ls -l MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Dec 29 13:15 MyED25519Key.pem
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

### task4\_describe\_keypairs.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-key-pairs
{
  "KeyPairs": [
    {
      "KeyPairId": "key-09f44f47900193352",
      "KeyType": "ed25519",
      "Tags": [],
      "CreateTime": "2025-12-29T13:15:50.652000+00:00",
      "KeyName": "MyED25519Key",
      "KeyFingerprint": "/jSbwMygM5dJ5YYhf26NAz1qPIroxOyNnTRtMICmiw="
    }
  ]
}
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

### Subnet ID:

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-subnets --query "Subnets[*].[SubnetId,AvailabilityZone,VpcId]" --output table
|-----+-----+-----+
| SubnetId | AvailabilityZone | VpcId |
+-----+-----+-----+
| subnet-85c852b94657a123e | us-east-1c | vpc-0399fce17f59406f7 |
| subnet-0e92b2e93c4e8bc5f | us-east-1b | vpc-0399fce17f59406f7 |
| subnet-0bba87c011ac1c522 | us-east-1e | vpc-0399fce17f59406f7 |
| subnet-0ad8443b59e0d0d5a1 | us-east-1a | vpc-0399fce17f59406f7 |
| subnet-8416380fc1c1c79c2 | us-east-1d | vpc-0399fce17f59406f7 |
| subnet-0202cfeedd2946fdc | us-east-1f | vpc-0399fce17f59406f7 |
+-----+-----+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

### Finding ami id:

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-images \
--owners amazon \
--filters "Name=name,Values=amzn2-ami-hvm-*-*x86_64-gp2" \
--query "Images | sort_by(@, &CreationDate) | [-1].[ImageId,Name]" \
--output table
|-----+-----|
|       DescribeImages          |
+-----+-----+
| ami-03f9680ef0c07a3d1      |
| amzn2-ami-hvm-2.0.20251208.0-x86_64-gp2 |
+-----+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

### Instance id:

```
--output table
|-----+-----+-----|
|       DescribeInstances        |
+-----+-----+-----+
| i-0a06c9c646989cdf7 | 54.144.228.128 | running |
+-----+-----+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

#### task4\_ssh\_permission\_error\_and\_fix.png

```
ED25519 key fingerprint is SHA256:0Y0JJ$JxcL5/xGzE2G5G7dZ2U6lhe2/Bxnz7486Qh5
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.144.228.128' (ED25519) to the list of known hosts.
      _#
      ~\_ #####_      Amazon Linux 2
      ~~ \#####\
      ~~ \###|      AL2 End of Life is 2026-06-30.
      ~~ \#/ ___
      ~~ \~' '-->
      ~~~ /      A newer version of Amazon Linux is available!
      ~~-. /_/
      _/_/_/      Amazon Linux 2023, GA and supported until 2028-03-15.
      _/m/          https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-17-133 ~]$ |
```

#### task4\_stop\_start\_terminate\_commands.png

```
Connection to localhost closed.
shell closed: exit status 254
PS C:\Users\hp> aws ec2 stop-instances --instance-ids i-0a06c9c646989cdf7
{
    "StoppingInstances": [
        {
            "InstanceId": "i-0a06c9c646989cdf7",
            "CurrentState": {
                "Code": 80,
                "Name": "stopped"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}

PS C:\Users\hp> aws ec2 start-instances --instance-ids i-0a06c9c646989cdf7
{
    "StartingInstances": [
        {
            "InstanceId": "i-0a06c9c646989cdf7",
            "CurrentState": {
```

```
PS C:\Users\hp> aws ec2 start-instances --instance-ids i-0a06c9c646989cdf7
{
    "StartingInstances": [
        {
            "InstanceId": "i-0a06c9c646989cdf7",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}

PS C:\Users\hp> |
```

task5\_describe\_security\_groups.png

```
{
    "SecurityGroups": [
        {
            "GroupId": "sg-03e7e8748d98464f7",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0399fce17f59406f7",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:353695163",
            "OwnerId": "353695163380",
            "GroupName": "default",
            "Description": "default VPC security group",
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [
                        {
                            "UserId": "353695163380",
                            "GroupId": "sg-03e7e8748d98464f7"
                        }
                    ]
                }
            ]
        }
    ],
    "More" -- |
```

task5\_describe\_vpcs.png

```
PS C:\Users\hp> aws ec2 describe-vpcs
{
    "Vpcs": [
        {
            "OwnerId": "353695163380",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0bd44b83db8727f98",
                    "CidrBlock": "172.31.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                },
                {
                    "IsDefault": true,
                    "BlockPublicAccessStates": {
                        "InternetGatewayBlockMode": "off"
                    },
                    "VpcId": "vpc-0399fce17f59406f7",
                    "State": "available",
                    "CidrBlock": "172.31.0.0/16",
                    "DhcpOptionsId": "dopt-0fb5d1a9bea1c4902"
                }
            ]
        }
    ]
}
```

task5\_describe\_subnets.png

```
{
    "Subnets": [
        {
            "AvailabilityZoneId": "use1-az4",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "353695163380",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:us-east-1:353695163380:subnet/subnet-05c052b94657a123e",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-05c052b94657a123e",
            "State": "available",
            "VpcId": "vpc-0399fce17f59406f7",
            "CidrBlock": "172.31.16.0/20",
            "AvailableIpAddressCount": 4096,
            "AvailabilityZone": "us-east-1c",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true
        }
    ],
    -- More -- |
}
```

task5\_describe\_instances.png

```
{  
    "Reservations": [  
        {  
            "ReservationId": "r-0e1eab825d84b0296",  
            "OwnerId": "353695163380",  
            "Groups": [],  
            "Instances": [  
                {  
                    "Architecture": "x86_64",  
                    "BlockDeviceMappings": [  
                        {  
                            "DeviceName": "/dev/xvda",  
                            "Ebs": {  
                                "AttachTime": "2025-12-29T13:24:44+00:00",  
                                "DeleteOnTermination": true,  
                                "Status": "attached",  
                                "VolumeId": "vol-0679b012e47bdaf5e"  
                            }  
                        }  
                    ],  
                    "ClientToken": "958c505f-bc07-4dda-9155-27b653c45ba9",  
                    "EbsOptimized": false,  
                    "EnaSupport": true,  
                    "Hypervisor": "xen",  
                    "NetworkInterfaces": [  
                        {  
                            "Association": {  
                                "IpOwnerId": "amazon",  
                                "PublicDnsName": "ec2-13-218-231-190.compute-1.amazonaws.com",  
                                "Primary": true  
                            }  
                        }  
                    ]  
                }  
            ]  
        }  
    ]  
}
```

task5\_describe\_regions.png

```
{  
    "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": "us-east-1",  
            "Endpoint": "ec2.us-east-1.amazonaws.com"  
        }  
    ]  
}
```

ask5\_describe\_availability\_zones.png

```
{  
    "AvailabilityZones": [  
        {  
            "OptInStatus": "opt-in-not-required",  
            "Messages": [],  
            "RegionName": "us-east-1",  
            "ZoneName": "us-east-1a",  
            "ZoneId": "use1-az1",  
            "GroupName": "us-east-1-zg-1",  
            "NetworkBorderGroup": "us-east-1",  
            "ZoneType": "availability-zone",  
            "GroupLongName": "US East (N. Virginia) 1",  
            "State": "available"  
        },  
        {  
            "OptInStatus": "opt-in-not-required",  
            "Messages": [],  
            "RegionName": "us-east-1",  
            "ZoneName": "us-east-1b",  
            "ZoneId": "use1-az2",  
            "GroupName": "us-east-1-zg-1",  
            "NetworkBorderGroup": "us-east-1",  
            "ZoneType": "availability-zone",  
            "GroupLongName": "US East (N. Virginia) 1",  
            "State": "available"  
        },  
        {  
            "OptInStatus": "opt-in-not-required",  
            "Messages": []  
        }  
    ]  
-- More -- |
```

task6\_create\_group\_and\_user.png

```
PS C:\Users\hp> aws iam create-group --group-name MyGroupCli  
{  
    "Group": {  
        "Path": "/",  
        "GroupName": "MyGroupCli",  
        "GroupId": "AGPAVEW074P2HAJ6V7AVV",  
        "Arn": "arn:aws:iam::353695163380:group/MyGroupCli",  
        "CreateDate": "2025-12-29T13:51:38+00:00"  
    }  
}  
PS C:\Users\hp> |
```

```
PS C:\Users\hp> aws iam get-group --group-name MyGroupCli  
{  
    "Users": [],  
    "Group": {  
        "Path": "/",  
        "GroupName": "MyGroupCli",  
        "GroupId": "AGPAVEW074P2HAJ6V7AVV",  
        "Arn": "arn:aws:iam::353695163380:group/MyGroupCli",  
        "CreateDate": "2025-12-29T13:51:38+00:00"  
    }  
}
```

task6\_create\_group\_and\_user.png

```

@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam create-user --user-name MyUserCli
{
    "User": {
        "Path": "/",
        "UserName": "MyUserCli",
        "UserId": "AIDAVEW074P2MCMHQ2XRG",
        "Arn": "arn:aws:iam::353695163380:user/MyUserCli",
        "CreateDate": "2025-12-29T13:56:03+00:00"
    }
}
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam get-user --user-name MyUserCli
{
    "User": {
        "Path": "/",
        "UserName": "MyUserCli",
        "UserId": "AIDAVEW074P2MCMHQ2XRG",
        "Arn": "arn:aws:iam::353695163380:user/MyUserCli",
        "CreateDate": "2025-12-29T13:56:03+00:00"
    }
}
@arshiajadoon → /workspaces/LAB_9 (main) $ |

```

task6\_add\_user\_to\_group\_and\_verify.png

```

@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam add-user-to-group --user-name MyUserCli --group-name MyGroupCli
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam get-group --group-name MyGroupCli
{
    "Users": [
        {
            "Path": "/",
            "UserName": "MyUserCli",
            "UserId": "AIDAVEW074P2MCMHQ2XRG",
            "Arn": "arn:aws:iam::353695163380:user/MyUserCli",
            "CreateDate": "2025-12-29T13:56:03+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "MyGroupCli",
        "GroupId": "AGPAVEW074P2HAJ6V7AVV",
        "Arn": "arn:aws:iam::353695163380:group/MyGroupCli",
        "CreateDate": "2025-12-29T13:51:38+00:00"
    }
}

```

task6\_policy\_list\_and\_attach.png

```

@arshiajadoon → /workspaces/LAB_9 (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
AWSElasticBeanstalkCustomPlatformforEC2Role
AmazonEC2ContainerServiceEventsRole
AmazonEC2SpotFleetTaggingRole
AWSEC2SpotServiceRolePolicy
AWSServiceRoleForEC2ScheduledInstances
AWSEC2SpotFleetServiceRolePolicy
AWSApplicationAutoScalingEC2SpotFleetRequestPolicy
AWSEC2FleetServiceRolePolicy
AWSAutoScalingPlansEC2AutoScalingPolicy
EC2InstanceConnect

```

```
@arshiajadoon → /workspaces/LAB_9 (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 |
+-----+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task6\_policy\_list\_and\_attach.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam attach-group-policy \
--group-name MyGroupCli \
--policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam list-attached-group-policies --group-name MyGroupCli
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonEC2FullAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
        }
    ]
}
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task6\_create\_login\_profile\_and\_signin.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam create-login-profile \
--user-name MyUserCli \
--password 'MySecurePassword123!' \
--password-reset-required
{
    "LoginProfile": {
        "UserName": "MyUserCli",
        "CreateDate": "2025-12-29T14:11:03+00:00",
        "PasswordResetRequired": true
    }
}
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

```
}
```

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam attach-group-policy \
--group-name MyGroupCli \
--policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam detach-group-policy \
--group-name MyGroupCli \
--policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task6\_create\_access\_key\_output.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam create-access-key --user-name MyUserCli
{
    "AccessKey": {
        "UserName": "MyUserCli",
        "AccessKeyId": "AKIAVEW074P2GCX4JH4K",
        "Status": "Active",
        "SecretAccessKey": "qzFHW/qrdQ+qbBIo9qjC79MhsPYg50rKrMDDrmlB",
        "CreateDate": "2025-12-29T14:12:31+00:00"
    }
}
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam list-access-keys --user-name MyUserCli
{
    "AccessKeyMetadata": [
        {
            "UserName": "MyUserCli",
            "AccessKeyId": "AKIAVEW074P2GCX4JH4K",
            "Status": "Active",
            "CreateDate": "2025-12-29T14:12:31+00:00"
        }
    ]
}
```

task6\_env\_exports\_and\_get\_user\_error.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ export AWS_ACCESS_KEY_ID=AKIAVEW074P2GCX4JH4K
export AWS_SECRET_ACCESS_KEY=qzfhw/qrdQ+qbbIo9qjC79MhsPYg50rKrMDDrmLB
printenv | grep AWS_
AWS_SECRET_ACCESS_KEY=qzfhw/qrdQ+qbbIo9qjC79MhsPYg50rKrMDDrmLB
AWS_ACCESS_KEY_ID=AKIAVEW074P2GCX4JH4K
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task6\_after\_logout\_and\_get\_user\_success.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam get-user --user-name MyUserCli
{
    "User": {
        "Path": "/",
        "UserName": "MyUserCli",
        "UserId": "AIDAVEW074P2MCMHQ2XRG",
        "Arn": "arn:aws:iam::353695163380:user/MyUserCli",
        "CreateDate": "2025-12-29T13:56:03+00:00"
    }
}
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task7\_filter\_by\_tag\_public\_ip.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-instances \
--filters "Name=tag:Name,Values=MyServer" \
--query "Reservations[ * ].Instances[ * ].PublicIpAddress" \
--output text
13.218.231.190
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task7\_filter\_by\_instance\_type.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-instances \
--filters "Name=instance-type,Values=t3.micro" \
--query "Reservations[ * ].Instances[ * ].InstanceId" \
--output table
-----
|  DescribeInstances   |
+-----+
|  i-0a06c9c646989cdf7  |
+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task7\_filter\_by\_subnet.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-instances \
--filters "Name=vpc-id,Values=vpc-0399fce17f59406f7" \
--query "Reservations[ * ].Instances[ * ].InstanceId" \
--output table
-----
|  DescribeInstances   |
+-----+
|  i-0a06c9c646989cdf7  |
+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task7\_filter\_by\_vpc.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-instances \
--filters "Name=vpc-id,Values=vpc-0399fce17f59406f7" \
--query "Reservations[*].Instances[*].InstanceId" \
--output table
-----
|   DescribeInstances   |
+-----+
| i-0a06c9c646989cdf7 |
+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

### Task 8: Query Outputs for Reporting

task8\_query\_table\_instances\_name\_ip.png

```
@arshiajadoon → /workspaces/LAB_9 (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-0a06c9c646989cdf7 | 13.218.231.190 | MyServer |
+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task8\_query\_table\_instance\_state.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-instances \
--query "Reservations[*].Instances[*].[InstanceId,State.Name]" \
--output table
-----
|       DescribeInstances       |
+-----+-----+
| i-0a06c9c646989cdf7 | running |
+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

task8\_query\_table\_instance\_type\_az.png

```
Output table
-----
|           DescribeInstances           |
+-----+-----+
| i-0a06c9c646989cdf7 | t3.micro | us-east-1c |
+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

### Cleanup: Remove All Resources

cleanup\_terminate\_instance.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 terminate-instances --instance-ids i-0a06c9c646989cdf7
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-0a06c9c646989cdf7",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

cleanup\_delete\_volumes\_snapshots.png

```
}
```

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-volumes --query "Volumes[*].[VolumeId,State]"
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-volumes --query "Volumes[*].[VolumeId,State]"
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-volumes --query "Volumes[*].[VolumeId,State]"
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

### cleanup\_delete\_security\_group\_and\_keypair.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-volumes --query "Volumes[*].[VolumeId,State]"
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-volumes --query "Volumes[*].[VolumeId,State]"
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-volumes --query "Volumes[*].[VolumeId,State]"
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 delete-security-group --group-id "sg-01fd8e6e7dc115fb6"
{
    "Return": true,
    "GroupId": "sg-01fd8e6e7dc115fb6"
}
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 delete-key-pair --key-name MyED25519Key
{
    "Return": true,
    "KeyPairId": "key-09f44f47900193352"
}
@arshiajadoon → /workspaces/LAB_9 (main) $ rm MyED25519Key.pem
rm: remove write-protected regular file 'MyED25519Key.pem'? Y
@arshiajadoon → /workspaces/LAB_9 (main) $ rm MyED25519Key.pem
rm: cannot remove 'MyED25519Key.pem': No such file or directory
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

### cleanup\_iam\_users\_deleted.png

```
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam list-access-keys --user-name MyUserCli
{
    "AccessKeyMetadata": [
        {
            "UserName": "MyUserCli",
            "AccessKeyId": "AKIAVEW074P2GCX4JH4K",
            "Status": "Active",
            "CreateDate": "2025-12-29T14:12:31+00:00"
        }
    ]
}
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam delete-access-key --user-name MyUserCli --access-key-id AKIAVEW074P2GCX4JH4K
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam delete-login-profile --user-name MyUserCli
An error occurred (NoSuchEntity) when calling the DeleteLoginProfile operation: Login Profile for User MyUserCli was not found.
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam remove-user-from-group --user-name MyUserCli --group-name MyGroupCli
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam delete-user --user-name MyUserCli
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam detach-group-policy --group-name MyGroupCli --policy-arm aws:policy/AmazonEC2FullAccess
An error occurred (NoSuchEntity) when calling the DetachGroupPolicy operation: Policy arn:aws:iam::aws:policy/AmazonEC2FullAccess was not found.
@arshiajadoon → /workspaces/LAB_9 (main) $ aws iam delete-group --group-name MyGroupCli
@arshiajadoon → /workspaces/LAB_9 (main) $ aws ec2 describe-instances --query "Reservations[*].Instances[*].State.Name" --output table
```

### cleanup\_summary.png#

```
State.Name" --output table
-----
|           DescribeInstances          |
+-----+-----+
| i-0a06c9c646989cdf7 | terminated |
+-----+-----+
@arshiajadoon → /workspaces/LAB_9 (main) $ |
```

