Name: Maira Malik

Reg.no: 2023-BSE-040

Subject: Cloud Computing

LAB # 09

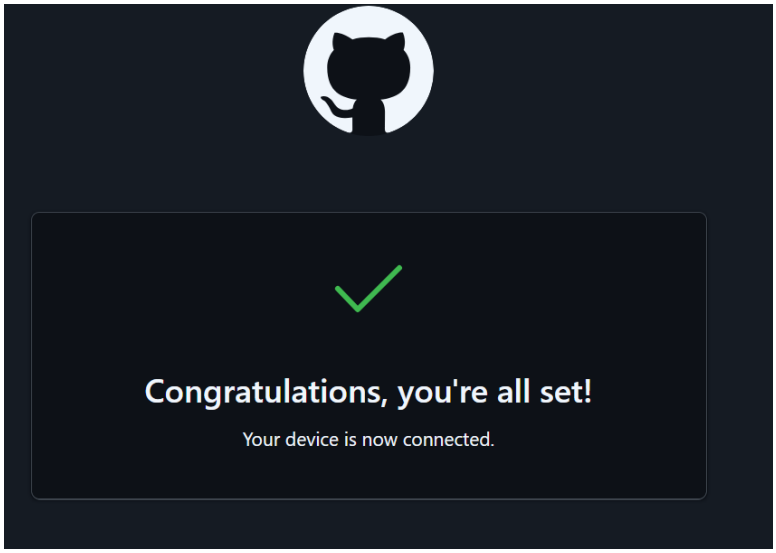
TASK 1:

1.

```
C:\Users\BOSS>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
```

2.

```
PS C:\Users\BOSS> 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
```



3.

```
Authentication complete.
- gh config set -h github.com git_protocol
Configured git protocol
Logged in as Maira222
PS C:\Users\BOSS> gh codespace list
no codespaces found
PS C:\Users\BOSS>
```

4.

```
PS C:\Users\BOSS> gh codespace create --repo maira222/lab9 --branch main --machine basicLinux32gb
  ⍰ Codespaces usage for this repository is paid for by Maira222
sturdy-sniffle-69xgxwq796q93rpp6
PS C:\Users\BOSS>
```

5.

```
PS C:\Users\BOSS> gh codespace ssh -c sturdy-sniffle-69xgxwq796q93rpp6
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

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

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

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

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

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

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

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

@Maira222 ⍰ /workspaces/lab9 (main) $
```

Task 2;

1.

```
@Maira222 ⍰ /workspaces/lab9 (main) $ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 60.3M  100 60.3M    0     0   165M      0 --:--:-- --:--:-- --:--:--  165M
@Maira222 ⍰ /workspaces/lab9 (main) $ unzip awscliv2.zip
Archive:  awscliv2.zip
   creating: aws/
   creating: aws/dist/
  inflating: aws/README.md
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/install
   creating: aws/dist/awscli/
   creating: aws/dist/dateutil/
   creating: aws/dist/docutils/
   creating: aws/dist/lib-dynload/
   creating: aws/dist/prompt_toolkit-3.0.51.dist-info/
   creating: aws/dist/wheel-0.45.1.dist-info/
  inflating: aws/dist/aws
  inflating: aws/dist/aws_completer
  inflating: aws/dist/libpython3.13.so.1.0
  inflating: aws/dist/_awscrt.abi3.so
  inflating: aws/dist/_ruamel_yaml.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/libz.so.1
  inflating: aws/dist/liblzma.so.5
  inflating: aws/dist/libbz2.so.1
  inflating: aws/dist/libffi.so.6
```

```
  inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licens
  inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
  inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
  inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
  inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.jso
  inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
  inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.t
  inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
  inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
@Maira222 ⍰ /workspaces/lab9 (main) $ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
@Maira222 ⍰ /workspaces/lab9 (main) $
```

2.

```
@Maira222 /workspaces/lab9 (main) $ aws --version
aws-cli/2.32.22 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
```

3.

```
@Maira222 /workspaces/lab9 (
AWS Access Key ID [**********
AWS Secret Access Key [None]:
Default region name [None]:
Default output format [None]:
@Maira222 /workspaces/lab9 (
```

4.

```
@Maira222 /wor
{
    "UserId": "A
    "Account": "
    "Arn": "arn:
}
@Maira222 /wor
```

Task 3:

1.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 create-security-group --group-name Lab9SecurityGroup --description "My Sec
urity Group" --vpc-id vpc-0549d3ac398b8e635
{
    "GroupId": "sg-053f227d71e45442e",
    "SecurityGroupArn": "arn:aws:ec2:me-central-1:737230811842:security-group/sg-053f227d71e45442e"
```

2.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-053f227d71e45442e
{
    "SecurityGroups": [
        {
            "GroupId": "sg-053f227d71e45442e",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0549d3ac398b8e635",
            "SecurityGroupArn": "arn:aws:ec2:me-central-1:737230811842:security-group/sg-053f227d71e45442e",
            "OwnerId": "737230811842",
            "GroupName": "Lab9SecurityGroup",
```

3.

```
@Maira222 /workspaces/lab9 (main) $ curl icanhazip.com
13.71.3.98
@Maira222 /workspaces/lab9 (main) $
```

4.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 authorize-security-group-ingress \
>   --group-id sg-053f227d71e45442e \
> --protocol tcp \
>   --port 22 \
>   --cidr 13.71.3.98/32
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-03b34d2059e72abb0",
            "GroupId": "sg-053f227d71e45442e",
            "GroupOwnerId": "737230811842",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "13.71.3.98/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:me-central-1:737230811842:security-gr
        }
    ]
}
@Maira222 /workspaces/lab9 (main) $
```

5.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-053f227d71e45442e
{
    "SecurityGroups": [
        {
            "GroupId": "sg-053f227d71e45442e",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0549d3ac398b8e635",
            "SecurityGroupArn": "arn:aws:ec2:me-central-1:737230811842:security-group/sg-053f227d71e45442e"
```

6.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 authorize-security-group-ingress   --group-id 'sg-053f227d71e4
-permissions '{"FromPort":80,"ToPort":80,"IpProtocol":"tcp","IpRanges":[{"CidrIp":"13.71.3.98/32"}]}'
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0501e6938a950d2bf",
            "GroupId": "sg-053f227d71e45442e",
            "GroupOwnerId": "737230811842",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "13.71.3.98/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:me-central-1:737230811842:security-group-rule/sgr-0501e6938
        }
    ]
}
@Maira222 /workspaces/lab9 (main) $
```

7.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-053f227d71e45442e
{
    "SecurityGroups": [
        {
            "GroupId": "sg-053f227d71e45442e",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0549d3ac398b8e635",
            "SecurityGroupArn": "arn:aws:ec2:me-central-1:737230811842:security-group/sg-053f227d71e45442e"
            "OwnerId": "737230811842",
            "GroupName": "Lab9SecurityGroup",
```

Task 4:

1.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 create-key-pair \
>   --key-name Lab9Key \
>  --key-type ed25519 \
>   --key-format pem \
>   --query 'KeyMaterial' \
>   --output text > Lab9Key.pem
@Maira222 /workspaces/lab9 (main) $ ls -l Lab9Key.pem
-rw-rw-rw- 1 codespace codespace 388 Dec 22 21:43 Lab9Key.pem
@Maira222 /workspaces/lab9 (main) $
```

2.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-key-pairs
{
    "KeyPairs": [
        {
            "KeyPairId": "key-0afa0f57fe6a89b64",
            "KeyType": "ed25519",
            "Tags": [],
            "CreateTime": "2025-12-22T21:43:47.763000+00:00",
            "KeyName": "Lab9Key",
            "KeyFingerprint": "IVrjH9wDCl1uJu7DUmTmX84CDmWfrmP56ZJA0KREtJ4="
        }
    ]
}
```

3.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 run-instances \
>   --image-id ami-05e66df2bafcb7dea \
>   --count 1 \
>   --instance-type t3.micro \
>   --key-name Lab9Key \
>  --security-group-ids sg-053f227d71e45442e \
> --subnet-id subnet-0761224bffed3cb12 \
>  --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=MyServer}]"
{
    "ReservationId": "r-011aacbcc65e5b157",
    "OwnerId": "737230811842",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "ab666cd3-c029-4288-b3cd-03f04f242cf6",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
```

4.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-instances \
>   --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress]" \
>   --output table
-----------------------------------------
|          DescribeInstances            |
+----------------------+----------------+
|  i-058222684cf47fe36 |  51.112.47.28  |
+----------------------+----------------+
@Maira222 /workspaces/lab9 (main) $
```

5.

```
@Maira222 /workspaces/lab9 (main) $ ssh -i Lab9Key.pem ec2-user@51.112.47.28
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-46-247 ~]$
```

6.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 stop-instances --instance-ids i-058222684cf47fe36
{
    "StoppingInstances": [
        {
            "InstanceId": "i-058222684cf47fe36",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
@Maira222 /workspaces/lab9 (main) $
```

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 start-instances --instance-ids i-058222684cf47fe36
{
    "StartingInstances": [
        {
            "InstanceId": "i-058222684cf47fe36",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
@Maira222 /workspaces/lab9 (main) $
```

Task 5:

1.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-security-groups
{
    "SecurityGroups": [
        {
            "GroupId": "sg-053f227d71e45442e",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0549d3ac398b8e635",
            "SecurityGroupArn": "arn:aws:ec2:me-central-1:737230811842:security-group/sg-053f227d71e45442e",
            "OwnerId": "737230811842",
            "GroupName": "Lab9SecurityGroup",
```

2.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-vpcs
{
    "Vpcs": [
        {
            "OwnerId": "737230811842",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0a71c3f7210aa4452",
                    "CidrBlock": "172.31.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": true,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-0549d3ac398b8e635",
            "State": "available",
            "CidrBlock": "172.31.0.0/16",
```

3.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-subnets
{
    "Subnets": [
        {
            "AvailabilityZoneId": "mec1-az2",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "737230811842",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:me-central-1:737230811842:subnet/subnet-0e52440a341f9f0dc",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0e52440a341f9f0dc",
            "State": "available",
```

4.

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 describe-instances
{
    "Reservations": [
        {
            "ReservationId": "r-011aacbcc65e5b157",
            "OwnerId": "737230811842",
            "Groups": [],
            "Instances": [
                {
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2025-12-22T21:48:59+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-0fede087eeb9ac3cd"
                            }
                        }
                    ],
                    "ClientToken": "ab666cd3-c029-4288-b3cd-03f04f242cf6",
```

5.

```
@Maira222 /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"
```

6.

```
@Maira222 /workspaces/lab9 (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",
            "ZoneId": "mec1-az2",
```

Cleanup:

```
@Maira222 /workspaces/lab9 (main) $ aws ec2 delete-security-group --group-id sg-053f227d71e45442e
{
    "Return": true,
    "GroupId": "sg-053f227d71e45442e"
}
@Maira222 /workspaces/lab9 (main) $ aws ec2 delete-key-pair --key-name Lab9Key
{
    "Return": true,
    "KeyPairId": "key-0afa0f57fe6a89b64"
}
@Maira222 /workspaces/lab9 (main) $
```