

Name : Zuha Irfan

Roll No : 2023-BSE-073

Subject: Cloud Computing

Semester & Section : V-B

Lab : 09

Task 1 — GitHub CLI, Codespace setup and authentication

- task1_gh_install.png

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

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

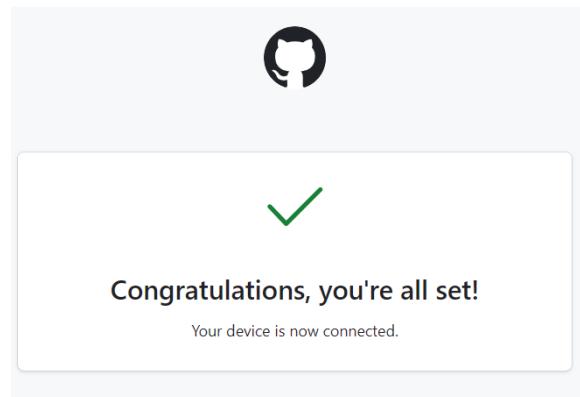
PS C:\Users\S Y S> winget install --id GitHub.cli
Found GitHub CLI [GitHub.cli] Version 2.83.1
The 'msstore' source requires that you view the following agreements before using.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
The source requires the current machine's 2-letter 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 2.83.1
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/releases/download/v2.83.1/gh_2.83.1_windows_amd64.msi
██████████████████ 17.6 MB / 17.6 MB
Successfully verified installer hash
Starting package install...
Successfully installed
PS C:\Users\S Y S> |
```

- task1_gh_auth_login.png

```
PS C:\Users\S Y S> 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: 5E25-8243
Press Enter to open https://github.com/login/device in your browser...
✓ Authentication complete.
- gh config set -h github.com git_protocol https
✓ Configured git protocol
|
```



- task1_codespace_list.png

```

PS C:\Lab9> gh codespace list
NAME          DISPLAY NAME      REPOSITORY    BRANCH STATE   CREATED AT
laughing-tribble-69xg6q7767rrh4g7  laughing tribble  Zuha-Irfan/Lab9  main Available about 2 minutes ago
PS C:\Lab9>

```

```

PS C:\Users\S Y S > & "C:\Program Files\GitHub CLI\gh.exe" auth status
github.com
  ✓ Logged in to github.com account Zuha-Irfan (keyring)
  - Active account: true
  - Git operations protocol: https
  - Token: gho_*****
  - Token scopes: 'codespace', 'gist', 'read:org', 'repo', 'workflow'

```

- task1_codespace_ssh_connected.png

```

PS C:\Lab9>
PS C:\Lab9> gh codespace ssh -c laughing-tribble-69xg6q7767rrh4g7
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.

@Zuha-Irfan → /workspaces/Lab9 (main) $

```

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

- task2_aws_install_and_version.png

```

@Zuha-Irfan → /workspaces/Lab9 (main) $ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
  % Total    % Received % Xferd  Average Speed   Time   Time     Time Current
               Dload  Upload Total Spent   Left Speed
100 60.0M 100 60.0M    0     0  52.6M    0:00:01  0:00:01 --:--:-- 52.6M
Archive: awscliv2.zip
  creating: aws/
  creating: aws/dist/
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/README.md
  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/_awsrt.abi3.so
  inflating: aws/dist/_ruamel_yaml.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib.so.1
  inflating: aws/dist/liblzma.so.5
  inflating: aws/dist/libbz2.so.1
  inflating: aws/dist/libffi.so.6
  inflating: aws/dist/libuuid.so.1
  inflating: aws/dist/libtinfo.so.5
  inflating: aws/dist/libreadline.so.6
  inflating: aws/dist/libsqlite3.so.0
  inflating: aws/dist/base_library.zip
  inflating: aws/dist/lib-dynload/_datetime.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_unicodedata.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_csv.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_statistics.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_contextvars.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_decimal.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_pickle.cpython-313-x86_64-linux-gnu.so

```

```

@Zuha-Irfan → /workspaces/Lab9 (main) $ aws --version
aws-cli/2.32.13 Python/3.13.9 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24

```

- task2_aws_configure_and_files.png

```
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws configure
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws configure
AWS Access Key ID [*****]: AKIAQYF75YQJRDCNZTVK
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
```

- task2_aws_get_caller_identity.png

```
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws sts get-caller-identity
{
    "UserId": "AIDAQYF75YQJZIJXFDSQK",
    "Account": "051942114323",
    "Arn": "arn:aws:iam::051942114323:user/Admin"
```

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

- task3_create_security_group_output.png

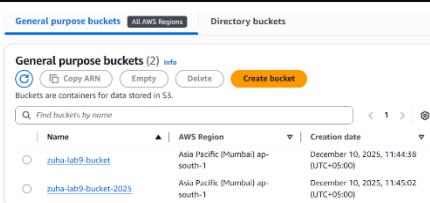
```
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws ec2 describe-vpcs --output table
  DescribeVpcs
+-----+
| VpcId |
+-----+
| vpc-03ed0c2085c712d1a |
+-----+
  BlockPublicAccessStates
+-----+
| State | off |
+-----+
  CidrBlockAssociationSet
+-----+
| AssociationId | CidrBlock |
+-----+
| vpc-cidi-assoc-07770d6795df7edd89 | 172.31.0.0/16 |
+-----+
  CidrBlockState
+-----+
| State | associated |
+-----+
```

- task3_describe_sg_before_ingress.png

```
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-008bdd0c335eb4af0
{
    "SecurityGroups": [
        {
            "GroupId": "sg-008bdd0c335eb4af0",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-03ed0c2085c712d1a",
            "SecurityGroupArn": "arn:aws:ec2:ap-south-1:051942114323:security-group/sg-008bdd0c335eb4af0",
            "OwnerId": "051942114323",
            "GroupName": "My Security Group",
            "Description": "My Security Group",
            "IpPermissions": []
        }
    ]
}
```

```
bash: syntax error near unexpected token `newline'
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws ec2 create-security-group --group-name MySecurityGroup --description "My Security Group" --vpc-id vpc-03ed0c2085c712d1a
{
    "GroupId": "sg-008bdd0c335eb4af0",
    "SecurityGroupArn": "arn:aws:ec2:ap-south-1:051942114323:security-group/sg-008bdd0c335eb4af0"
```

```
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws s3 mb s3://zuha-lab9-bucket
make_bucket: zuha-lab9-bucket
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws s3 mb s3://zuha-lab9-bucket-2025
make_bucket: zuha-lab9-bucket-2025
@Zuha-Irfan ~ /workspaces/Lab9 (main) $ aws s3 mb s3://your-bucket-name
make_bucket: your-bucket-name
make_bucket failed: s3://your-bucket-name An error occurred (BucketAlreadyExists) when calling the CreateBucket operation: The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.
-bash: make_bucket: command not found
```



- task3_codespace_public_ip.png

```
@Zuha-Irfan → /workspaces/Lab9 (main) $ curl icanhazip.com
20.192.21.49
```

- task3_authorize_ssh_and_describe.png

```
@Zuha-Irfan → /workspaces/Lab9 (main) $ aws ec2 authorize-security-group-ingress --group-id sg-088bdd0c335eb4af0 --protocol tcp --port 22 --cidr 172.31.0.0/16
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-812d8812eaa0b28d3",
            "GroupId": "sg-088bdd0c335eb4af0",
            "GroupOwnerId": "051942114323",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "172.31.0.0/16",
            "SecurityGroupRuleArn": "arn:aws:ec2:ap-south-1:051942114323:security-group-rule/sgr-812d8812eaa0b28d3"
        }
    ]
}
```

- task3_authorize_http_and_describe.png

```
@Zuha-Irfan → /workspaces/Lab9 (main) $ aws ec2 authorize-security-group-ingress --group-id sg-088bdd0c335eb4af0 --protocol tcp --port 80 --cidr 0.0.0.0/0
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-8cb65ef8ecf56b8bf",
            "GroupId": "sg-088bdd0c335eb4af0",
            "GroupOwnerId": "051942114323",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "172.31.0.0/16",
            "SecurityGroupRuleArn": "arn:aws:ec2:ap-south-1:051942114323:security-group-rule/sgr-8cb65ef8ecf56b8bf"
        }
    ]
}
```

- task3_describe_sg_final.png

```
@Zuha-Irfan → /workspaces/Lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-088bdd0c335eb4af0
{
    "SecurityGroups": [
        {
            "GroupId": "sg-088bdd0c335eb4af0",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "*0.0.0.0/*"
                        }
                    ],
                    "IpvRanges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-08edc208c712d1a",
            "SecurityGroupArn": "arn:aws:ec2:ap-south-1:051942114323:security-group/sg-088bdd0c335eb4af0",
            "OwnerId": "051942114323",
            "GroupName": "My Security Group",
            "Description": "My security group",
            "IpPermissions": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "172.31.0.0/16"
                        }
                    ],
                    "IpvRanges": [],
                    "PrefixListIds": []
                }
            ]
        }
    ],
    "skipping...
    {
        "SecurityGroups": [
            {
                "GroupId": "sg-088bdd0c335eb4af0",
                "IpPermissionsEgress": [
                    {
                        "IpProtocol": "-1",
                        "UserIdGroupPairs": [],
                        "IpRanges": [
                            {
                                "CidrIp": "*0.0.0.0/*"
                            }
                        ],
                        "IpvRanges": [],
                        "PrefixListIds": []
                    }
                ],
                "VpcId": "vpc-08edc208c712d1a",
                "SecurityGroupArn": "arn:aws:ec2:ap-south-1:051942114323:security-group/sg-088bdd0c335eb4af0",
                "OwnerId": "051942114323",
                "GroupName": "My Security Group",
                "Description": "My Security Group",
                "IpPermissions": [

```

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

- task4_create_keypair_output.png

```
~ $ aws ec2 create-key-pair --key-name MyED25519Key --key-type ed25519 --key-format pem --query 'KeyMaterial' --output text > MyED25519Key.pem
-bash: MyED25519Key.pem: Permission denied
~ $ ls -l MyED25519Key.pem
-r-----. 1 cloudshell-user cloudshell-user 1675 Dec 10 08:13 MyED25519Key.pem
```

- task4_describe_keypairs.png

```
~ $ aws ec2 describe-key-pairs
{
    "KeyPairs": [
        {
            "KeyId": "key-0d82a077aed22c64",
            "KeyType": "rsa",
            "Tags": [],
            "CreateTime": "2025-12-10T08:13:51.446000+00:00",
            "KeyName": "MyED25519Key",
            "KeyFingerprint": "d1:83:49:42:b8:85:d1:ca:ee:01:f5:ae:37:0c:22:86:d1:b2:43:82"
        }
    ]
}
~ $ aws ec2 run-instances \
> --image-id ami-0b3c632b6b7289e44 \
> --count 1 \
> --instance-type t3.micro \
> --key-name MyED25519Key \
> --security-group-ids sg-0e87a857917e64df5 \
> --subnet-id subnet-068e847e705dcade9 \
> --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=MyServer}]"
{
    "ReservationId": "r-01a688fb9fc2fd380",
    "OwnerId": "051942114323",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "d30670c7-1f2d-415e-9676-82a4da816885",
            "EbsOptimized": false,
            "ImageId": "ami-0b3c632b6b7289e44",
            "InstanceId": "i-01a688fb9fc2fd380",
            "InstanceType": "t3.micro",
            "KeyName": "MyED25519Key",
            "LaunchTime": "2025-12-10T08:13:51.446000+00:00",
            "Monitoring": {
                "Enabled": false
            },
            "NetworkInterfaces": [
                {
                    "Association": {
                        "AllocationId": "eni-01a688fb9fc2fd380",
                        "AssociationId": "eni-01a688fb9fc2fd380",
                        "Primary": true
                    },
                    "MacAddress": "enp0s8",
                    "NetworkInterfaceId": "eni-01a688fb9fc2fd380",
                    "PrivateDns": "ip-172-31-1-12.ec2.internal",
                    "PrivateIpAddress": "172.31.1.12",
                    "Status": "in-use"
                }
            ],
            "Placement": {
                "AvailabilityZone": "us-east-1a",
                "GroupName": null
            },
            "State": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

- task4_delete_keypair_optional.png

```
@Zuha-Irfan → /workspaces/Lab9 (main) $  
{  
    "Return": true,  
    "KeyPairId": "key-0c813fb8285ade560"  
}
```

- task4_run_instances_output.png

```

$ az vm run-instances \
> --image-id amlt-002-532ab0-289e44 \
> --count 1 \
> --os-type Linux \
> --key-name MyID255Key \
> --security-group-ids sg-0e867ab5-7917-4d9f \
> --subnet-id subnet-686874c5-5fd5ac69 \
> --tag-specification "ResourceType=Instance,Tags=[{Key='Name',Value='MyServer'}]"
{
  "ReservationId": "r-01a3c5b7a98717562",
  "Location": "westus2",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientStates": "idle36-7656-4d74-9682-9760ce93f5f7",
      "Timestamp": false,
      "EnsSupport": true,
      "HyperV": "xen",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachmentId": "2005-12-07T14:25:36+00:00",
            "AttachmentLuid": "eni-attach-000e254d83898b6b",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "MacAddress": "00-0e-25-4d-83-89",
            "NetworkCardIndex": 0
          },
          "Connection": {}
        }
      ],
      "OsType": "Linux"
    }
  ]
}

```

- task4 describe instances public ip.png

```
"Ipv6Addresses": [],  
~ $ aws ec2 describe-instances \  
> --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress]" \  
> --output table  
  
+-----+-----+  
| DescribeInstances |  
+-----+-----+  
| i-06592c1e36e8f8a | 13.239.236.251 |  
| i-08e0f8eae460f597 | 13.211.77.180 |  
| i-0c075e18714c0f3b | 15.134.145.153 |  
| i-0e012d769e116aa72 | 3.106.57.20 |  
| i-0cabbe7e8ec54eed2 | 13.54.236.250 |
```

- task4 ssh permission error and fix.png

- task4 stop start terminate commands.png

```

~ $ aws ec2 stop-instances --instance-ids i-0c075e918714c0f3b
{
  "StoppingInstances": [
    {
      "InstanceId": "i-0c075e918714c0f3b",
      "CurrentState": {
        "Code": 64,
        "Name": "stopping"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}

~ $ aws ec2 start-instances --instance-ids i-0c075e918714c0f3b
{
  "StartingInstances": [
    {
      "InstanceId": "i-0c075e918714c0f3b",
      "CurrentState": {
        "Code": 0,
        "Name": "pending"
      },
      "PreviousState": {
        "Code": 80,
        "Name": "stopped"
      }
    }
  ]
}

~ $ aws ec2 terminate-instances --instance-ids i-0c075e918714c0f3b
{
  "TerminatingInstances": [
    {
      "InstanceId": "i-0c075e918714c0f3b",
      "CurrentState": {
        "Code": 32,
        "Name": "shutting-down"
      },
      "PreviousState": {
        "Code": 16,
        "Name": "running"
      }
    }
  ]
}

```

Task 5 — Understand AWS describe-* commands

- task5_describe_security_groups.png

```

~ $ aws ec2 describe-security-groups
{
  "SecurityGroups": [
    {
      "GroupId": "sg-0e867a857917eddf5",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": []
        }
      ],
      "VpcId": "vpc-05e2515142602cb2",
      "SecurityGroupName": "arn:aws:ec2:ap-southeast-2:051942114323:security group/sg-0e867a857917eddf5",
      "OwnerId": "051942114323",
      "GroupType": "VPC",
      "Description": "default VPC security group",
      "IpPermissions": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [
            {
              "UserId": "051942114323",
              "GroupId": "sg-0e867a857917eddf5"
            }
          ]
        }
      ]
    }
  ]
}

```

- task5_describe_vpcs.png

```

~ $ aws ec2 describe-vpcs
{
  "Vpcs": [
    {
      "OwnerId": "051942114323",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-038511c51649d258",
          "CidrBlock": "172.31.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ],
      "IsDefault": true,
      "BlockPublicAccessStates": {
        "InternetGatewayBlockMode": "off"
      },
      "VpcId": "vpc-05e2515142602cb2",
      "State": "available",
      "CidrBlock": "172.31.0.0/16",
      "DhcpOptionsId": "dopt-0abb70e699dc964bd"
    }
  ]
}

```

- task5_describe_subnets.png

```
~ $ aws ec2 describe-subnets
{
  "Subnets": [
    {
      "AvailabilityZoneId": "apse2-az1",
      "MapCustomerOwnedIp": false,
      "OwnerId": "951942114323",
      "AssignIpv6AddressOnCreation": false,
      "IpOwnerIdBlockAssociationSet": [],
      "SubnetCidrBlock": "10.0.0.0/24",
      "OwnerId": "951942114323",
      "EnableDns64": false,
      "Ip6Gateway": false,
      "PrivateIpAddressAssignmentOnLaunch": {
        "NameType": "ip-name",
        "EnableSourceNameOnAwsAccount": false,
        "EnableSourceNameOnAwsRegion": false
      },
      "BlockPublicAccessStates": {
        "InternetGatewayBlockMode": "off"
      },
      "SubnetId": "subnet-0b642a240b0bd22",
      "VpcId": "vpc-05e2515142620cb2",
      "CidrBlock": "172.31.12.0/28",
      "AssociateRouteTable": "951942114323",
      "AvailabilityZone": "ap-southeast-2b",
      "DefaultIaZ: true,
      "MapPublicIpOnLaunch": true
    },
    {
      "AvailabilityZoneId": "apse2-az1",
    }
  ]
}
```

- task5_describe_instances.png

```
~ $ aws ec2 describe-instances
{
  "Reservations": [
    {
      "ReservationId": "r-0131le67152923a89",
      "OwnerId": "951942114323",
      "Groups": [],
      "Instances": [
        {
          "Architecture": "x86_64",
          "BlockDeviceMappings": [
            {
              "DeviceName": "/dev/xvda",
              "Ebs": {
                "AttachTime": "2025-12-10T08:27:19+00:00",
                "DeleteOnTermination": true,
                "Status": "attached",
                "VolumeId": "vol-04e5af3491ac4d777"
              }
            }
          ],
          "ClientToken": "4b460835-8724-4448-bd76-fa3cb44a2548",
          "EbsOptimized": false,
          "EnaSupport": true,
          "Hypervisor": "xen",
          "NetworkInterfaces": [
            {
              "Association": {
                "IpAddress": "13.239.236.251",
                "PrivateIpAddress": "ec2-13-239-236-251.ap-southeast-2.compute.amazonaws.com",
                "PublicIp": "13.239.236.251"
              }
            }
          ]
        }
      ]
    }
  ]
}
```

- task5_describe_regions.png

```
~ $ 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"
    }
  ]
}
```

- task5_describe_availability_zones.png

```
~ $ aws ec2 describe-availability-zones
{
  "AvailabilityZones": [
    {
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "ap-southeast-2",
      "ZoneName": "ap-southeast-2a",
      "ZoneId": "apsse2-az1",
      "GroupName": "ap-southeast-2-zg-1",
      "NetworkBorderGroup": "ap-southeast-2",
      "ZoneType": "availability-zone",
      "GroupLongName": "Asia Pacific (Sydney) 1",
      "State": "available"
    },
    {
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "ap-southeast-2",
      "ZoneName": "ap-southeast-2b",
      "ZoneId": "apsse2-az2",
      "GroupName": "ap-southeast-2-zg-1",
      "NetworkBorderGroup": "ap-southeast-2",
      "ZoneType": "availability-zone",
      "GroupLongName": "Asia Pacific (Sydney) 1",
      "State": "available"
    },
    {
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "ap-southeast-2"
    }
  ]
}
```

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

- task6_create_group_and_user.png

```
RegionName : ap-southeast-2 ,  
~ $ aws iam create-group --group-name MyGroupCli  
{  
    "Group": {  
        "Path": "/",  
        "GroupName": "MyGroupCli",  
        "GroupId": "AGPAQYF75YQJWUZLKA6GQ",  
        "Arn": "arn:aws:iam::051942114323:group/MyGroupCli",  
        "CreateDate": "2025-12-10T14:50:36+00:00"  
    }  
}  
An error occurred (EntityAlreadyExists) when calling the CreateGroup operation:  
~ $ aws iam get-group --group-name MyGroupCli  
{  
    "Users": [],  
    "Group": {  
        "Path": "/",  
        "GroupName": "MyGroupCli",  
        "GroupId": "AGPAQYF75YQJWUZLKA6GQ",  
        "Arn": "arn:aws:iam::051942114323:group/MyGroupCli",  
        "CreateDate": "2025-12-10T14:50:36+00:00"  
    }  
}  
  
~ $ aws iam create-user --user-name MyUserCli  
{  
    "User": {  
        "Path": "/",  
        "UserName": "MyUserCli",  
        "UserId": "AIDAQYF75YQJ4MX4VHGCB",  
        "Arn": "arn:aws:iam::051942114323:user/MyUserCli",  
        "CreateDate": "2025-12-10T14:52:18+00:00"  
    }  
}  
~ $ aws iam get-user --user-name MyUserCli  
{  
    "User": {  
        "Path": "/",  
        "UserName": "MyUserCli",  
        "UserId": "AIDAQYF75YQJ4MX4VHGCB",  
        "Arn": "arn:aws:iam::051942114323:user/MyUserCli",  
        "CreateDate": "2025-12-10T14:52:18+00:00"  
    }  
}
```

- task6_add_user_to_group_and_verify.png

```
~ $ aws iam add-user-to-group --user-name MyUserCli --group-name MyGroupCli  
~ $ aws iam get-group --group-name MyGroupCli  
{  
    "Users": [  
        {  
            "Path": "/",  
            "UserName": "MyUserCli",  
            "UserId": "AIDAQYF75YQJ4MX4VHGCB",  
            "Arn": "arn:aws:iam::051942114323:user/MyUserCli",  
            "CreateDate": "2025-12-10T14:52:18+00:00"  
        }  
    ],  
    "Group": {  
        "Path": "/",  
        "GroupName": "MyGroupCli",  
        "GroupId": "AGPAQYF75YQJWUZLKA6GQ",  
        "Arn": "arn:aws:iam::051942114323:group/MyGroupCli",  
        "CreateDate": "2025-12-10T14:50:36+00:00"  
    }  
}
```

- task6_policy_list_and_attach.png

```
~ $ aws iam list-policies \  
  --query "Policies[?contains(PolicyName, 'EC2')].{Name:PolicyName}" \  
  --output text  
AmazonEC2FullAccess  
AmazonEC2RunInstances  
AmazonEC2ImageBuilderForComputeRole  
AmazonEC2RoleforDataPipelineRole  
AmazonEC2ContainerServiceForEC2Role  
AmazonEC2RoleforLambda  
AmazonEC2RoleforAWSCodeDeploy  
AmazonEC2RoleforSSM  
CloudWatchMetricsForEC2Access  
AmazonEC2ContainerRegistryReadOnly  
AmazonEC2ContainerRegistryPowerUser  
AmazonEC2ContainerRegistryFullAccess  
AmazonEC2ContainerServiceAccessRole  
AmazonEC2ContainerServiceEventRole  
AWSLambdaContainerCustomPlatformForEC2Role  
AmazonEC2ContainerServiceEventsRole  
AmazonEC2SpotFleetTaggingRole  
AmazonEC2ContainerServicePolicy  
AWSLambdaRoleforEC2ScheduledInstances  
AWSFleetPolicyServiceRolePolicy  
AWSLambdaContainerServiceRolePolicy  
AWSAutoScalingPlanst2AutoScalingPolicy  
EC2InstanceConnect  
AmazonEC2ContainerServiceForAmazonMQizard  
EC2InstanceConnectFileImageBuilder  
EC2FleetTimeShiftableServiceRolePolicy  
AmazonEC2RoleforAWSCodeDeployLimited
```

```

~ $ aws iam list-policies \
>   --query 'Policies[?PolicyName==`AmazonEC2FullAccess`].{Name:PolicyName, Arn:Arn}' \
>   --output table
+-----+-----+
|          ListPolicies          |
+-----+-----+
|      Arn           |      Name       |
+-----+-----+
| arn:aws:iam::aws:policy/AmazonEC2FullAccess | AmazonEC2FullAccess |
+-----+-----+

```

- task6_create_login_profile_and_signin.png

```

~ $ aws iam create-login-profile \
>   --user-name MyUserCli \
>   --password "TempP@sw0rd!" \
>   --password-reset-required
{
  "LoginProfile": {
    "UserName": "MyUserCli",
    "CreatedDate": "2025-12-10T14:58:24+00:00",
    "PasswordResetRequired": true
  }
}
~ $ aws iam attach-group-policy \
>   --group-name MyGroupCli \
>   --policy-arm arn:aws:iam::aws:policy/IAMUserChangePassword
~ $ aws iam detach-group-policy \
>   --group-name MyGroupCli \
>   --policy-arm arn:aws:iam::aws:policy/IAMUserChangePassword

```

- task6_create_access_key_output.png

```

~ $ aws iam create-access-key --user-name MyUserCli
{
  "AccessKey": {
    "UserName": "MyUserCli",
    "AccessKeyId": "AKIAQYF75YQJ3OKL13U3",
    "Status": "Active",
    "SecretAccessKey": "CEZUPq/luA5o61JNsZH6Mebe9JN2RmhgYSoM821V",
    "CreateDate": "2025-12-10T14:59:58+00:00"
  }
}
~ $ aws iam list-access-keys --user-name MyUserCli
{
  "AccessKeyMetadata": [
    {
      "UserName": "MyUserCli",
      "AccessKeyId": "AKIAQYF75YQJ3OKL13U3",
      "Status": "Active",
      "CreateDate": "2025-12-10T14:59:58+00:00"
    }
  ]
}

```

```

~ $ aws iam attach-group-policy \
>   --group-name MyGroupCli \
>   --policy-arm arn:aws:iam::aws:policy/AmazonEC2FullAccess
~ $ aws iam list-attached-group-policies --group-name MyGroupCli
{
  "AttachedPolicies": [
    {
      "PolicyName": "AmazonEC2FullAccess",
      "PolicyArn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
    }
  ]
}

```

- task6_env_exports_and_get_user_error.png

```

~ $ export AWS_ACCESS_KEY_ID=AKIAQYF75YQJ3OKL13U3
~ $ export AWS_SECRET_ACCESS_KEY=CEZUPq/luA5o61JNsZH6Mebe9JN2RmhgYSoM821V
~ $ export AWS_DEFAULT_REGION=ap-southeast-2
~ $ printenv | grep AWS_
AWS_CONTAINER_CREDENTIALS_FULL_URL=http://localhost:1338/latest/meta-data/container/security-credentials
AWS_DEFAULT_REGION=ap-southeast-2
AWS_ECR_METADATA_DISABLED=true
AWS_DEFAULT_REGION=ap-southeast-2
AWS_REGION=ap-southeast-2
SET_DNF_REGION_SCRIPT=$env | grep -m 1 AWS_REGION | grep -E '[a-z0-9-]*' | sudo tee /etc/dnf/vars/awsregion
AWS_PAGER=cat
AWS_SUPPRESS_MAINTENANCE_MODE_MESSAGE=true
AWS_SECRET_ACCESS_KEY=CEZUPq/luA5o61JNsZH6Mebe9JN2RmhgYSoM821V
AWS_CONTAINER_AUTHORIZATION_TOKEN=DOH7r1gBFCC4k7aTFw6Fab1KuBvnAEBgvXAg828-
AWS_ACCESS_KEY_ID=AKIAQYF75YQJ3OKL13U3
AWS_TOOL_USER_AWS_CLOUDSHIELD=2025-10-21
AWS_ACCESS_KEY_ID=AKIAQYF75YQJ3OKL13U3
$ AWS_SECRET_ACCESS_KEY=CEZUPq/luA5o61JNsZH6Mebe9JN2RmhgYSoM821V
$ AWS_DEFAULT_REGION=ap-southeast-2
$ aws sts get-caller-identity
{
  "UserId": "AIDAQYF75YQJ4MX4VHGCB",
  "Account": "051942114323",
  "Arn": "arn:aws:iam::051942114323:user/MyUserCli"
}
$ 

```

- task6_after_logout_and_get_user_success.png

```

~ $ unset AWS_ACCESS_KEY_ID
~ $ unset AWS_SECRET_ACCESS_KEY
~ $ unset AWS_DEFAULT_REGION
~ $ aws iam get-user --user-name MyUserCli
{
  "User": {
    "Path": "/",
    "UserName": "MyUserCli",
    "UserId": "AIDAQYF75YQJ4MX4VHGCB",
    "Arn": "arn:aws:iam::051942114323:user/MyUserCli",
    "CreateDate": "2025-12-10T14:52:18+00:00"
  }
}

```

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

- task7_filter_by_tag_public_ip.png

```
~ $ aws ec2 describe-instances \
>   --filters "Name=tag:Name,Values=MyServer" \
>   --query "Reservations[*].Instances[*].PublicIpAddress" \
>   --output text
13.239.236.251
13.211.77.180
3.106.57.20
13.54.236.250
```

- task7_filter_by_instance_type.png

```
~ $ aws ec2 describe-instances \
>   --filters "Name=instance-type,Values=t3.micro" \
>   --query "Reservations[].Instances[].[InstanceId]" \
>   --output table
+-----+
| DescribeInstances |
+-----+
| i-065928c1ee36e8f8a |
| i-08e0f8e8ea460f597 |
| i-0c075e918714c0f3b |
| i-0e012d769e116aa72 |
| i-0cabbe7e8ec54eed2 |
+-----+
```

- task7_filter_by_subnet.png

```
~ $ aws ec2 describe-instances \
>   --filters "Name=subnet_id,Values=subnet-068e847e705dca6c9" \
>   --query "Reservations[*].Instances[*].InstanceId" \
>   --output table
+-----+
| DescribeInstances |
+-----+
| i-065928c1ee36e8f8a |
| i-08e0f8e8ea460f597 |
| i-0e012d769e116aa72 |
| i-0cabbe7e8ec54eed2 |
+-----+
```

- task7_filter_by_vpc.png

```
~ $ aws ec2 describe-vpcs --query "Vpcs[*].VpcId" --output text
vpc-05e25151426020cb2
~ $ aws ec2 describe-instances \
>   --filters "Name=vpc-id,Values=vpc-05e25151426020cb2" \
>   --query "Reservations[*].Instances[*].InstanceId" \
>   --output table
+-----+
| DescribeInstances |
+-----+
| i-065928c1ee36e8f8a |
| i-08e0f8e8ea460f597 |
| i-0e012d769e116aa72 |
| i-0cabbe7e8ec54eed2 |
+-----+
```

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

- task8_query_table_instances_name_ip.png

```
~ $ aws ec2 describe-instances \
>   --filters "Name=tag:Name,Values=MyServer" \
>   --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress,Tags[?Key=='Name'].Value[0]]" \
>   --output table
+-----+
|             DescribeInstances           |
+-----+
| i-065928c1ee36e8f8a | 13.239.236.251 | MyServer |
| i-08e0f8e8ea460f597 | 13.211.77.180 | MyServer |
| i-0c075e918714c0f3b | None          | MyServer |
| i-0e012d769e116aa72 | 3.106.57.20  | MyServer |
| i-0cabbe7e8ec54eed2 | 13.54.236.250 | MyServer |
+-----+
```

- task8_query_table_instance_state.png

```
~ $ aws ec2 describe-instances \
> --query "Reservations[*].Instances[*.InstanceId,State.Name]" \
> --output table
+-----+-----+
| DescribeInstances | |
+-----+-----+
| i-065928c1e36e8f8a | running |
| i-08efb8eae460f597 | running |
| i-0c075ea918714c0f3b | terminated |
| i-0012d769e116aa72 | running |
| i-0cabbe7e8ec54eed2 | running |
+-----+-----+
~ $
```

- task8_query_table_instance_type_az.png

```
$ aws ec2 describe-instances \
--query "Reservations[*].Instances[*].[InstanceId,InstanceType,Placement.AvailabilityZone]" \
--output table
```

DescribeInstances			
InstanceId	InstanceType	Placement AvailabilityZone	
i-065928c1ee36e8f8a	t3.micro	ap-southeast-2a	
i-08ef0f8e8a460f597	t3.micro	ap-southeast-2a	
i-0c975e918714cd5b	t3.micro	ap-southeast-2a	
i-0e812d769e16aa72	t3.micro	ap-southeast-2a	
i-0cabbef88ec54eed2	t3.micro	ap-southeast-2a	

Cleanup — Remove resources to avoid charges

- cleanup terminate instance.png

```
$ aws ec2 terminate-instances --instance-ids i-0e012d769e116aa72 i-0cabbe7e8ec54eed2
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-0e012d769e116aa72",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        },
        {
            "InstanceId": "i-0cabbe7e8ec54eed2",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

- cleanup_delete_volumes_snapshots.png

```
- $ aws ec2 describe-volumes --query "Volumes[*].[VolumeId,State,Attachments]" --output table  
- $ aws ec2 describe-snapshots --owner-id $(aws sts get-caller-identity --query "Account" --output text) --query "Snapshots[*].[SnapshotId,VolumeId,State]" --output table
```

- cleanup delete security group and keypair.png

```
~ $ aws ec2 delete-key-pair --key-name MyED25519Key
{
    "Return": true,
    "KeyId": "key-0d82a0777aed22c64"
}
```

- cleanup iam users deleted.png

```
-~ $ aws iam delete-access-key --user-name MyUserCli --access-key-id AKIAQYF75YQJ3KL13U3
-~ $ aws iam delete-login-profile --user-name MyUserCli
-~ $ aws iam detach-user --user-name MyUserCli --group-name MyGroupCli
-~ $ aws iam delete-group --group-name MyGroupCli
-~ $ aws iam detach-group-policy --group-name MyGroupCli --policy-arm arn:aws:iam::aws:policy/AmazonEC2FullAccess
-~ $ aws iam delete-group --group-name MyGroupCli
```

- cleanup summary.png

```

$ aws ec2 describe instances
{
    "Reservations": [
        {
            "ReservationId": "r-8121e67515293d89",
            "OwnerId": "953942114123",
            "Groups": [],
            "Instances": [
                {
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [],
                    "ClientToken": "4d46835 8724 4448 bd76 f3cb4d4e548",
                    "EbsOptimized": false,
                    "ImageId": "ami-00000000",
                    "IngressPermissions": [],
                    "NetworkInterfaceCount": 1,
                    "RootDeviceName": "/dev/xvda",
                    "RootDeviceType": "Amazon EBS",
                    "SecurityGroups": [],
                    "StateReason": {
                        "Code": "ClientUserInitiatedShutdown",
                        "Message": "Client user initiated shutdown"
                    },
                    "Tags": [
                        {
                            "Key": "Name",
                            "Value": "MyServer"
                        }
                    ],
                    "VirtualizationType": "hvm",
                    "VmOptions": {}
                }
            ]
        }
    ]
}

```

```
~ $ aws ec2 describe-volumes
{
    "Volumes": []
}
```

```
~ $ aws ec2 describe-security-groups
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0e867a857917e4df5",
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-08c291532d620cb7",
            "SecurityGroupRules": "arn:aws:ec2:ap-southeast-2:051942114323:security-group/sg-0e867a857917e4df5",
            "OwnerId": "051942114323",
            "GroupName": "default",
            "Description": "default VPC security group",
            "Tags": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [
                        {
                            "UserId": "051942114323",
                            "GroupId": "sg-0e867a857917e4df5"
                        }
                    ]
                }
            ]
        }
    ]
}
```

```
~ $ aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "Admin",
            "UserId": "AIDAQYF75YQZIJXFD5QK",
            "Arn": "arn:aws:iam::051942114323:user/Admin",
            "CreateDate": "2025-12-09T13:20:01+00:00",
            "PasswordLastUsed": "2025-12-10T06:14:32+00:00"
        }
    ]
}
~ $ aws iam list-groups
{
    "Groups": []
}
```

