

**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.

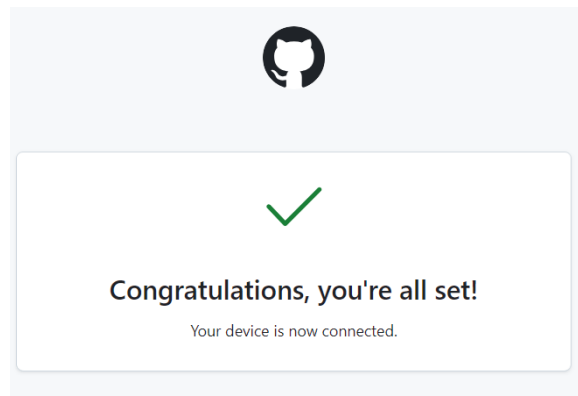
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\S Y S> winget install --id GitHub.cli
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/cli/releases/download/v2.83.1/gh_2.83.1_windows_amd64.msi
27.6 MB / 27.6 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> 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 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/libz.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/_datetimestr.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) $
@Zuha-Irfan → /workspaces/Lab9 (main) $ aws configure
AWS Access Key ID [*****dmin]: AKIAQYF75YQJRCNZTVK
AWS Secret Access Key [None]: 605sooIcZzJnekhPq07d1GwGamBuIBG2QHMgbAMj
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": "AIDAQYF75YQJZIJXFDQK",
  "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						
Vpcs						
CidrBlock	DhcpOptionsId	InstanceTenancy	IsDefault	OwnerId	State	VpcId
172.31.0.0/16	dopt-0016e1d34ba594b91	default	True	051942114323	available	vpc-03ed0c2085c712d1a

```

+-----+-----+
| InternetGatewayBlockMode |
+-----+-----+
| off |
+-----+-----+
| CidrBlockAssociationSet |
+-----+-----+
| AssociationId | CidrBlock |
+-----+-----+
| vpc-cidr-assoc-0770d6795df7edd89 | 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": "MySecurityGroup",
      "Description": "My Security Group",
      "IpPermissions": []
    }
  ]
}
```

```
@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
```

General purpose buckets All AWS Regions Directory buckets

General purpose buckets [2] info

Copy ARN Empty Delete Create bucket

Buckets are containers for data stored in S3.

Find buckets by name

Name	AWS Region	Creation date
<a href="#">zuha-lab9-bucket</a>	Asia Pacific (Mumbai) ap-south-1	December 10, 2025, 11:44:38 (UTC+05:00)
<a href="#">zuha-lab9-bucket-2025</a>	Asia Pacific (Mumbai) ap-south-1	December 10, 2025, 11:45:02 (UTC+05:00)

- 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-008b0dc335eb4af0 --protocol tcp --port 22 --cidr 172.31.0.0/16
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-012d8812aaa0b28d3",
      "GroupId": "sg-008b0dc335eb4af0",
      "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-012d8812aaa0b28d3"
    }
  ]
}
```

- task3\_authorize\_http\_and\_describe.png

```
@Zuha-Irfan → /workspaces/Lab9 (main) $
{
  "Return": true,
  "SecurityGroupRules": [
    {
      "SecurityGroupRuleId": "sgr-0cb65ef8ecf56b0bf",
      "GroupId": "sg-008b0dc335eb4af0",
      "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-0cb65ef8ecf56b0bf"
    }
  ]
}
```

- task3\_describe\_sg\_final.png

```
@Zuha-Irfan → /workspaces/Lab9 (main) $ aws ec2 describe-security-groups --group-ids sg-008b0dc335eb4af0
{
  "SecurityGroups": [
    {
      "GroupId": "sg-008b0dc335eb4af0",
      "IpPermissionsEgress": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "IpRanges": [],
          "PrefixListIds": []
        }
      ],
      "IpId": "type-03dd8c2885c712d1a",
      "SecurityGroupArn": "arn:aws:ec2:ap-south-1:051942114323:security-group/sg-008b0dc335eb4af0",
      "OwnerId": "051942114323",
      "GroupName": "MySecurityGroup",
      "Description": "My Security Group",
      "IpPermissions": [
        {
          "IpProtocol": "tcp",
          "FromPort": 22,
          "ToPort": 22,
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "172.31.0.0/16"
            }
          ],
          "IpRanges": [],
          "PrefixListIds": []
        }
      ]
    }
  ],
  "Skipping": [
    {
      "SecurityGroups": [
        {
          "GroupId": "sg-008b0dc335eb4af0",
          "IpPermissionsEgress": [
            {
              "IpProtocol": "-1",
              "UserIdGroupPairs": [],
              "IpRanges": [
                {
                  "CidrIp": "0.0.0.0/0"
                }
              ],
              "IpRanges": [],
              "PrefixListIds": []
            }
          ],
          "IpId": "type-03dd8c2885c712d1a",
          "SecurityGroupArn": "arn:aws:ec2:ap-south-1:051942114323:security-group/sg-008b0dc335eb4af0",
          "OwnerId": "051942114323",
          "GroupName": "MySecurityGroup",
          "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": [
    {
      "KeyPairId": "key-0082a077aed22c64",
      "KeyType": "rsa",
      "Tags": [],
      "CreateTime": "2025-12-10T08:13:51.446000+00:00",
      "KeyName": "Hyd02519Key",
      "FingerPrint": "d1:83:4d:c2:b8:85:d1:ca:ee:01:f5:ae:37:0c:22:86:d1:b2:43:82"
    }
  ]
}

> $ aws ec2 run-instances \
> --image-id ami-0b3c8326b67289e44 \
> --count 1 \
> --instance-type t3.micro \
> --key-name Hyd02519Key \
> --security-group-ids sg-0e867a85791d4df5 \
> --subnet-id subnet-068e847e705dca6c9 \
> --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=MyServer}]"
{
  "ReservationId": "r-01a688fb9fc2fd388",
  "OwnerId": "051942114323",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "d38c70c7-1f2d-415e-9676-82a4da816885",
      "EbsOptimized": false,

```

- task4\_delete\_keypair\_optional.png

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

- task4 run instances output.png

```

- $ .\src\src\main\instance.ps1
> -image-id smi-0b3c8326b6728b6dd \
> -count 1 \
> -instance-type t3.micro \
> -key-name MyD25551KeyKey \
> -security-group idsg-0b07a057917e4df5 \
> -subnet-id subnet-0d6e657e7b654ac93 \
> -tag-specifications "ResourceType=Instance,Tags=[{Key=Name,Value=MyServer}]"
{
  "ReservationId": "r-01ae3f7b498f71562",
  "OwnerId": "951942114323",
  "Groups": [],
  "Instances": [
    {
      "Architecture": "x86_64",
      "BlockDeviceMappings": [],
      "ClientToken": "23a6b36-7456-4764-9682-976bec915f77",
      "EbsOptimized": false,
      "EnaSupport": true,
      "Hypervisor": "x86",
      "NetworkInterfaces": [
        {
          "Attachment": {
            "AttachTime": "2025-12-10T14:25:36+00:00",
            "AttachmentId": "eni-attach-00be254d889f886b6",
            "DeleteOnTermination": true,
            "DeviceIndex": 0,
            "Status": "attaching",
            "NetworkCardIndex": 0
          }
        }
      ],
      "PrivateIp": "10.0.1.1"
    }
  ]
}

```

- task4\_describe\_instances\_public\_ip.png

```
~ $ aws ec2 describe-instances \
> --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress]" \
> --output table
```

DescribeInstances	
i-065928c1ee36e8f8a	13.239.236.251
i-08e0f8e8ea460f597	13.211.77.180
i-0c075e918714c0f3b	15.134.145.153
i-0e012d769e0116a72	3.186.57.20
i-0cabb67e8ec54eed2	13.54.236.250

- task4 ssh permission error and fix.png

[illegible]

- 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": 80,
        "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-0e867a857917edd5",
      "IpPermissions": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": []
        }
      ],
      "VpcId": "vpc-05e25151426020c2",
      "SecurityGroups": "aws-aws-ec2/ap-southeast-2-051942114323:security-group/sg-0e867a857917edd5",
      "OwnerId": "051942114323",
      "Groupname": "default",
      "Description": "default VPC security group",
      "IpPermissions": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [
            {
              "UserId": "051942114323",
              "GroupId": "sg-0e867a857917edd5"
            }
          ]
        }
      ]
    }
  ]
}

```

- task5\_describe\_vpcs.png

```

~ $ aws ec2 describe-vpcs
{
  "Vpcs": [
    {
      "OwnerId": "051942114323",
      "InstanceTenancy": "default",
      "CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-038511c5164942587",
          "CidrBlock": "172.31.0.0/16",
          "CidrBlockState": {
            "State": "associated"
          }
        }
      ],
      "IsDefault": true,
      "BlockPublicAccessStates": {
        "InternetGatewayBlockMode": "off"
      },
      "VpcId": "vpc-05e25151426020c2",
      "State": "available",
      "CidrBlock": "172.31.0.0/16",
      "DhcpOptionsId": "dopt-0abb70e699dc964bd"
    }
  ]
}

```

- task5\_describe\_subnets.png

```
~ $ aws ec2 describe-subnets
{
  "Subnets": [
    {
      "AvailabilityZoneId": "apse2-az3",
      "MapCustomerOwnedIpv4Address": false,
      "OwnerId": "951942114323",
      "AssignIpv6AddressesOnCreation": false,
      "Ipv6CidrBlockAssociationSet": [],
      "SubnetId": "subnet-e2-ap-south-east-2-051942114323",
      "Subnet": "subnet-e2-ap-south-east-2-051942114323",
      "EnabledForIpv6": false,
      "IpPermissions": false,
      "PrivateDnsEntryOptions": {
        "Hostnametype": "ip_name",
        "EnableResourceRecordCaching": false,
        "EnableResourceRecordAAAARecord": false
      },
      "BlockPublicAccess": {
        "InternetGatewayBlock": "off"
      },
      "SubnetId": "subnet-0b642a24b0c7bd22",
      "State": "available",
      "VpcId": "vpc-05e2515126080c3d",
      "CidrBlock": "172.31.32.0/20",
      "AvailabilityZone": "ap-southeast-2b",
      "DefaultAz": true,
      "MapPublicIpOnLaunch": true
    },
    {
      "AvailabilityZoneId": "apse2-az1",
```

- task5\_describe\_instances.png

```
~ $ aws ec2 describe-instances
{
  "Reservations": [
    {
      "ReservationId": "r-0131e67152923a89",
      "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-0d6ef3491acdd777"
              }
            }
          ],
          "ClientToken": "4b460835-8724-4448-bd76-fa3cb4a2548",
          "EbsOptimized": false,
          "EnaSupport": true,
          "Hypervisor": "amzn",
          "NetworkInterfaces": [
            {
              "Association": {
                "IpOwnerId": "amazon",
                "PublicIpSubnet": "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": "apse2-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": "apse2-az3",
      "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

```

~ $ aws iam create-group --group-name MyGroupCli
{
  "Group": {
    "Path": "/",
    "GroupName": "MyGroupCli",
    "GroupId": "AGPAQYF75YQJMUZLKA6GQ",
    "Arn": "arn:aws:iam:051942114323:group/MyGroupCli",
    "CreateDate": "2025-12-10T14:50:36+00:00"
  }
}
An error occurred (EntityAlreadyExists) when calling the CreateGroup operation: The entity already exists.
~ $ aws iam get-group --group-name MyGroupCli
{
  "Users": [],
  "Group": {
    "Path": "/",
    "GroupName": "MyGroupCli",
    "GroupId": "AGPAQYF75YQJMUZLKA6GQ",
    "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

```

- $ az srm add-user-to-group --user-name MyUserC11 --group-name MyGroupC11
- $ az srm get-group --group-name MyGroupC11
{
  "Users": [
    {
      "Path": "/",
      "UserName": "MyUserC11",
      "UserId": "A1DAQVF75YQJ4HX4VHGBG",
      "Arn": "arn:aws:iam:051942114323:user/MyUserC11",
      "CreateDate": "2025-12-10T14:52:18+00:00"
    }
  ],
  "Group": {
    "Path": "/",
    "GroupName": "MyGroupC11",
    "GroupId": "AGPAQVF75YQJWJLKA66G",
    "Arn": "arn:aws:iam:051942114323:group/MyGroupC11",
    "CreateDate": "2025-12-10T14:50:36+00:00"
  }
}

```

- task6\_policy\_list\_and\_attach.png

```

5 select distinct policy_name
6   query "Policies[?contains(PolicyName, '{C2}').(Name-PolicyName)]\"
7   AmazonC2FullAccess
8   AmazonC2FullAccess
9   AmazonLastClickBehaviorForC2Role
10  AmazonC2RoleForDataPipelineRole
11  AmazonC2RoleForDataPipelineRole
12  AmazonC2RoleForDataPipelineRole
13  AmazonC2RoleForDataPipelineRole
14  AmazonC2RoleForDataPipelineRole
15  AmazonC2RoleForDataPipelineRole
16  AmazonC2RoleForDataPipelineRole
17  AmazonC2RoleForDataPipelineRole
18  AmazonC2RoleForDataPipelineRole
19  AmazonC2RoleForDataPipelineRole
20  AmazonC2RoleForDataPipelineRole
21  AmazonC2RoleForDataPipelineRole
22  AmazonC2RoleForDataPipelineRole
23  AmazonC2RoleForDataPipelineRole
24  AmazonC2RoleForDataPipelineRole
25  AmazonC2RoleForDataPipelineRole
26  AmazonC2RoleForDataPipelineRole
27  AmazonC2RoleForDataPipelineRole
28  AmazonC2RoleForDataPipelineRole
29  AmazonC2RoleForDataPipelineRole
30  AmazonC2RoleForDataPipelineRole
31  AmazonC2RoleForDataPipelineRole
32  AmazonC2RoleForDataPipelineRole
33  AmazonC2RoleForDataPipelineRole
34  AmazonC2RoleForDataPipelineRole
35  AmazonC2RoleForDataPipelineRole
36  AmazonC2RoleForDataPipelineRole
37  AmazonC2RoleForDataPipelineRole
38  AmazonC2RoleForDataPipelineRole
39  AmazonC2RoleForDataPipelineRole
40  AmazonC2RoleForDataPipelineRole
41  AmazonC2RoleForDataPipelineRole
42  AmazonC2RoleForDataPipelineRole
43  AmazonC2RoleForDataPipelineRole
44  AmazonC2RoleForDataPipelineRole
45  AmazonC2RoleForDataPipelineRole
46  AmazonC2RoleForDataPipelineRole
47  AmazonC2RoleForDataPipelineRole
48  AmazonC2RoleForDataPipelineRole
49  AmazonC2RoleForDataPipelineRole
50  AmazonC2RoleForDataPipelineRole
51  AmazonC2RoleForDataPipelineRole
52  AmazonC2RoleForDataPipelineRole
53  AmazonC2RoleForDataPipelineRole
54  AmazonC2RoleForDataPipelineRole
55  AmazonC2RoleForDataPipelineRole
56  AmazonC2RoleForDataPipelineRole
57  AmazonC2RoleForDataPipelineRole
58  AmazonC2RoleForDataPipelineRole
59  AmazonC2RoleForDataPipelineRole
60  AmazonC2RoleForDataPipelineRole
61  AmazonC2RoleForDataPipelineRole
62  AmazonC2RoleForDataPipelineRole
63  AmazonC2RoleForDataPipelineRole
64  AmazonC2RoleForDataPipelineRole
65  AmazonC2RoleForDataPipelineRole
66  AmazonC2RoleForDataPipelineRole
67  AmazonC2RoleForDataPipelineRole
68  AmazonC2RoleForDataPipelineRole
69  AmazonC2RoleForDataPipelineRole
70  AmazonC2RoleForDataPipelineRole
71  AmazonC2RoleForDataPipelineRole
72  AmazonC2RoleForDataPipelineRole
73  AmazonC2RoleForDataPipelineRole
74  AmazonC2RoleForDataPipelineRole
75  AmazonC2RoleForDataPipelineRole
76  AmazonC2RoleForDataPipelineRole
77  AmazonC2RoleForDataPipelineRole
78  AmazonC2RoleForDataPipelineRole
79  AmazonC2RoleForDataPipelineRole
80  AmazonC2RoleForDataPipelineRole
81  AmazonC2RoleForDataPipelineRole
82  AmazonC2RoleForDataPipelineRole
83  AmazonC2RoleForDataPipelineRole
84  AmazonC2RoleForDataPipelineRole
85  AmazonC2RoleForDataPipelineRole
86  AmazonC2RoleForDataPipelineRole
87  AmazonC2RoleForDataPipelineRole
88  AmazonC2RoleForDataPipelineRole
89  AmazonC2RoleForDataPipelineRole
90  AmazonC2RoleForDataPipelineRole
91  AmazonC2RoleForDataPipelineRole
92  AmazonC2RoleForDataPipelineRole
93  AmazonC2RoleForDataPipelineRole
94  AmazonC2RoleForDataPipelineRole
95  AmazonC2RoleForDataPipelineRole
96  AmazonC2RoleForDataPipelineRole
97  AmazonC2RoleForDataPipelineRole
98  AmazonC2RoleForDataPipelineRole
99  AmazonC2RoleForDataPipelineRole
100 AmazonC2RoleForDataPipelineRole

```



```
~ $ 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@ssw0rd!" \
> --password-reset-required
{
  "LoginProfile": {
    "UserName": "MyUserCli",
    "CreateDate": "2025-12-10T14:58:24+00:00",
    "PasswordResetRequired": true
  }
}
~ $ aws iam attach-group-policy \
> --group-name MyGroupCli \
> --policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword
~ $ aws iam detach-group-policy \
> --group-name MyGroupCli \
> --policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword
```

- task6\_create\_access\_key\_output.png

```
~ $ aws iam create-access-key --user-name MyUserCli
{
  "AccessKey": {
    "UserName": "MyUserCli",
    "AccessKeyId": "AKIAQYF75YQJ3OKLI3U3",
    "Status": "Active",
    "SecretAccessKey": "CEZUPq/1uA5o61JNsZH6Mebe9JN2RhgYS0M821V",
    "CreateDate": "2025-12-10T14:59:58+00:00"
  }
}
~ $ aws iam list-access-keys --user-name MyUserCli
{
  "AccessKeyMetadata": [
    {
      "UserName": "MyUserCli",
      "AccessKeyId": "AKIAQYF75YQJ3OKLI3U3",
      "Status": "Active",
      "CreateDate": "2025-12-10T14:59:58+00:00"
    }
  ]
}
```

```
~ $ aws iam attach-group-policy \
> --group-name MyGroupCli \
> --policy-arn 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=AKIAQYF75YQJ3OKLI3U3
~ $ export AWS_SECRET_ACCESS_KEY=CEZUPq/1uA5o61JNsZH6Mebe9JN2RhgYS0M821V
~ $ export AWS_DEFAULT_REGION=ap-southeast-2
~ $ printenv | grep AWS
AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost:1338/latest/meta-data/container/security-credentials
AWS_EXECUTION_ENV=CloudShell
AWS_EC2_METADATA_DISABLED=true
AWS_DEFAULT_REGION=ap-southeast-2
AWS_REGION=ap-southeast-2
SET_OMF_REGION_SCRIPT=env | grep -m 1 AWS_REGION | grep -Eo '[a-z0-9-]*' | sudo tee /etc/dnf/vars/awregion
AWS_PAGER=less -K
AWS_S3_SUPPRESS_MAINTEINANCE_MODE_MESSAGE=true
AWS_SECRET_ACCESS_KEY=CEZUPq/1uA5o61JNsZH6Mebe9JN2RhgYS0M821V
AWS_CONTAINER_AUTHORIZATION_TOKEN=00b21r6BfCC+4k7aP1FvM6Fab1KuBvNAEbgVvXAgB28-
AWS_ACCESS_KEY_ID=AKIAQYF75YQJ3OKLI3U3
AWS_TOOLING_USER_AGENT=aws-CloudShell/2025.10.21
~ $ AWS_ACCESS_KEY_ID=AKIAQYF75YQJ3OKLI3U3
~ $ AWS_SECRET_ACCESS_KEY=CEZUPq/1uA5o61JNsZH6Mebe9JN2RhgYS0M821V
~ $ AWS_DEFAULT_REGION=ap-southeast-2
~ $ aws sts get-caller-identity
{
  "UserId": "AIDAQYF75YQJ4MX4VHGC8",
  "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": "AIDAQYF75YQJ4MX4VHGC8",
    "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-065928c1ee36e8f8a | running |
i-08e0f8e8ea460f597 | running |
i-0c075e918714c0f3b | terminated |
i-0e012d769e116aa72 | 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
-----
i-065928c1ee36e8f8a | t3.micro | ap-southeast-2a |
i-08e0f8e8ea460f597 | t3.micro | ap-southeast-2a |
i-0c075e918714c0f3b | t3.micro | ap-southeast-2a |
i-0e012d769e116aa72 | t3.micro | ap-southeast-2a |
i-0cabbe7e8ec54eed2 | 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-ids $(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,
  "KeyPairId": "key-0d82a0777aed22c64"
}
```

- cleanup\_iam\_users\_deleted.png

```
~ $ aws iam delete-access-key --user-name MyUserCli --access-key-id AKIAQYF75VQJ30KL1303
~ $ aws iam delete-login-profile --user-name MyUserCli
~ $ aws iam remove-user-from-group --user-name MyUserCli --group-name MyGroupCli
~ $ aws iam delete-user --user-name MyUserCli
~ $ aws iam detach-group-policy --group-name MyGroupCli --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
~ $ aws iam delete-group --group-name MyGroupCli
```

- cleanup\_summary.png

```
~ $ aws ec2 describe-instances
{
  "Reservations": [
    {
      "ReservationId": "r-011e07b15292a00",
      "OwnerId": "9194214432",
      "Groups": [],
      "Instances": [
        {
          "Architecture": "x86_64",
          "BlockDeviceMappings": [],
          "ClientToken": "4b4d0835-8724-4448-bd76-fa3b4da3548",
          "EbsOptimized": false,
          "EnclaveSupport": "mvp",
          "Hypervisor": "x86",
          "NetworkInterfaces": [],
          "RootDeviceName": "/dev/xvda",
          "RootDeviceType": "ebs",
          "SecurityGroups": [],
          "StateReasons": [
            {
              "Code": "Client.UserInitiatedShutdown",
              "Message": "Client.UserInitiatedShutdown: User initiated shutdown"
            }
          ],
          "Tags": [
            {
              "Key": "Name",
              "Value": "MyServer"
            }
          ],
          "VirtualizationType": "hvm",
          "CpuOptions": {

```

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

```
~ $ aws ec2 describe-security-groups
{
  "SecurityGroups": [
    {
      "GroupId": "sg-0e857a57917e4d4f5",
      "IpPermissions": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [],
          "IpRanges": [
            {
              "CidrIp": "0.0.0.0/0"
            }
          ],
          "Ipv6Ranges": [],
          "PrefixListIds": []
        }
      ],
      "VpcId": "vpc-0e25515142682b32",
      "SecurityGroupName": "arn:aws:ec2:ap-southeast-2:051942114323::security-group/sg-0e857a57917e4d4f5",
      "OwnerId": "051942114323",
      "GroupName": "default",
      "Description": "default VPC security group",
      "IpPermissions": [
        {
          "IpProtocol": "-1",
          "UserIdGroupPairs": [
            {
              "UserId": "051942114323",
              "GroupId": "sg-0e857a57917e4d4f5"
            }
          ]
        }
      ]
    }
  ]
}
```

```
~ $ aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "UserName": "Admin",
      "UserId": "AIDAQVF75YQJZIJXFD5QK",
      "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": []
}
```

