CLOUD COMPUTING LAB
BSE (V-B)

Submitted By:

Musfira Farooq

Roll No:

2023-BSE-045

Submitted To:

Sir Muhammad Shoaib

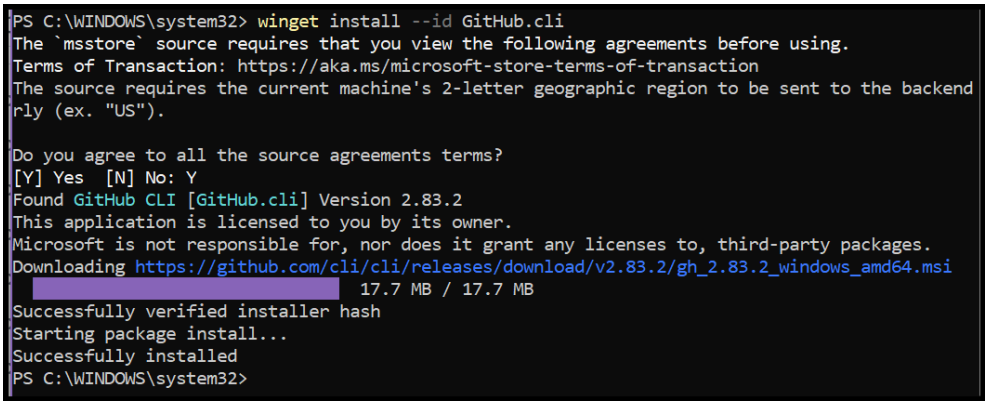
LAB 09

Codespaces + AWS: GH CLI, AWS CLI, EC2, IAM, Security Groups, Filters & Queries

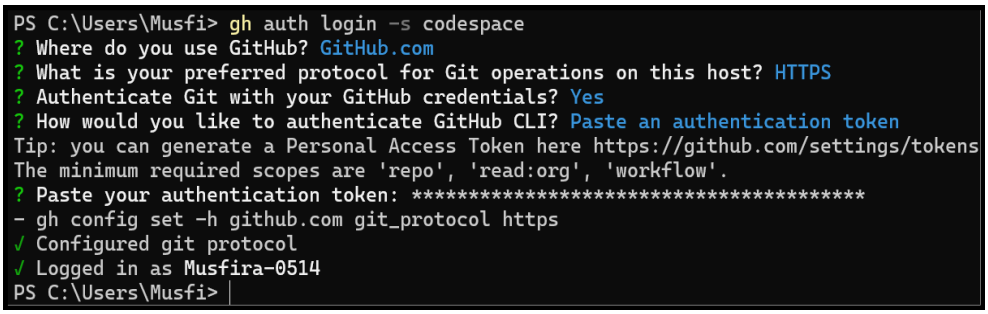
Task 1 — GitHub CLI, Codespace setup and authentication

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

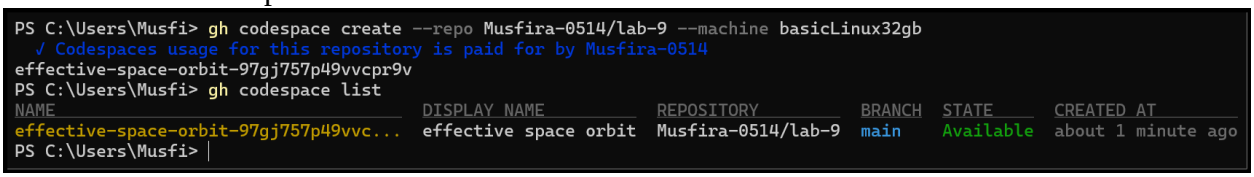
- Install GitHub CLI



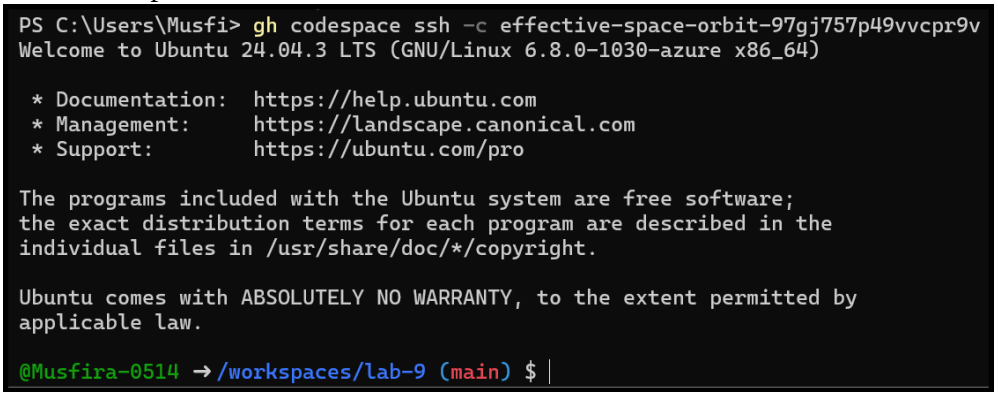
- GH CLI authentication for Codespaces



- List available Codespaces



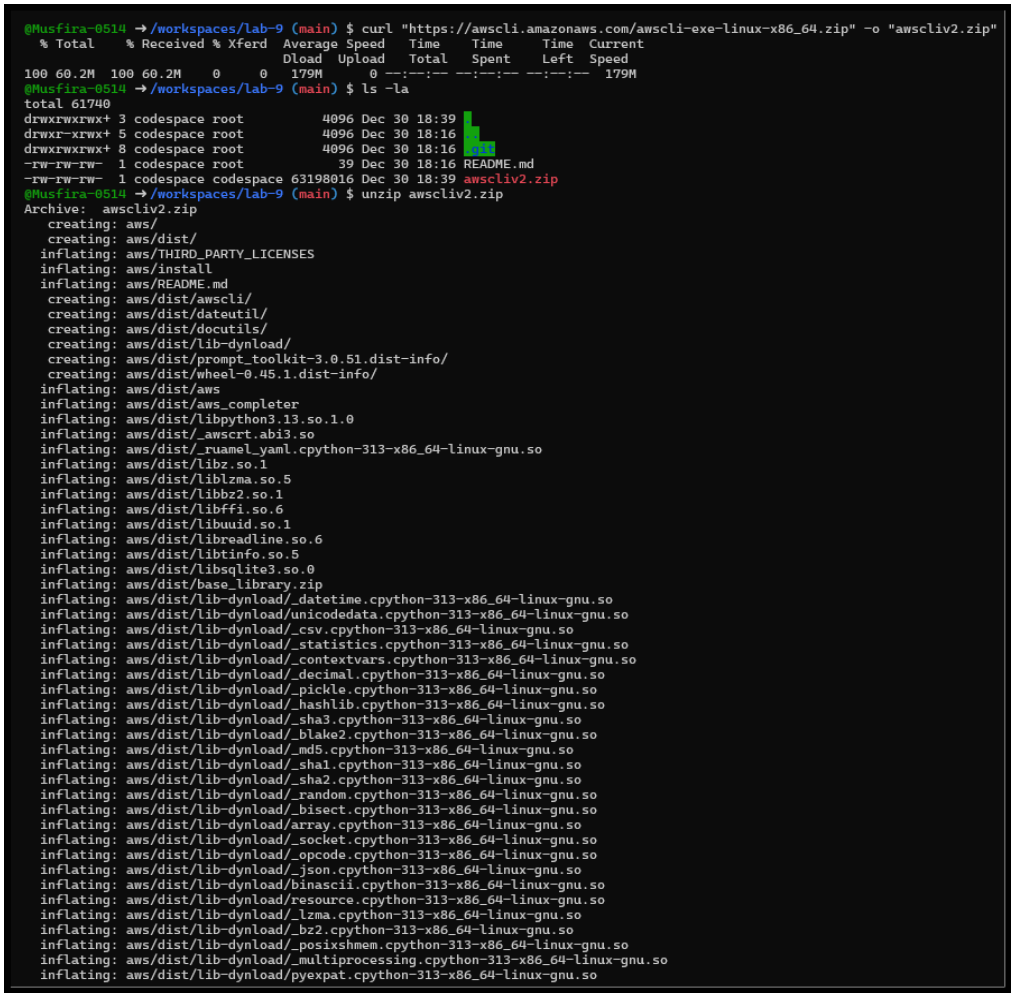
- Connect to Codespace



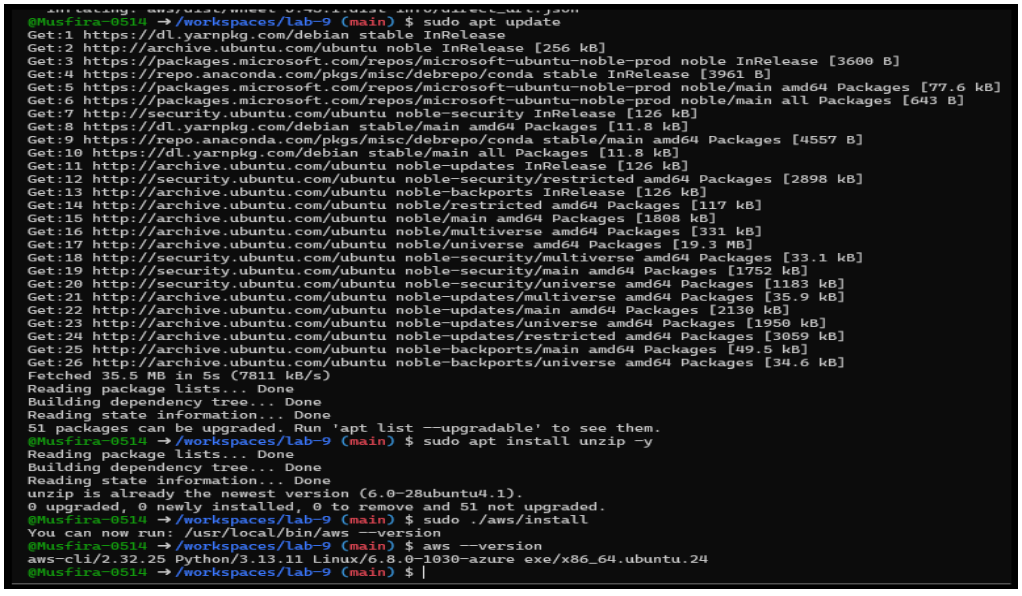
Task 2 — Install AWS CLI inside the Codespace and configure it

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

- AWS CLI install and version check

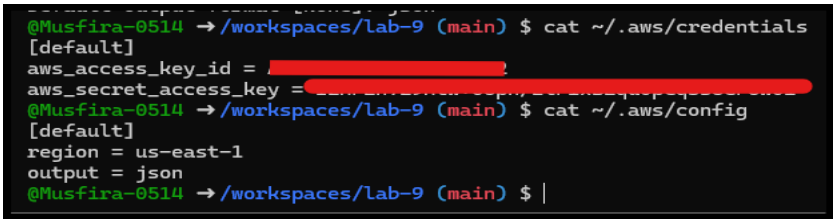


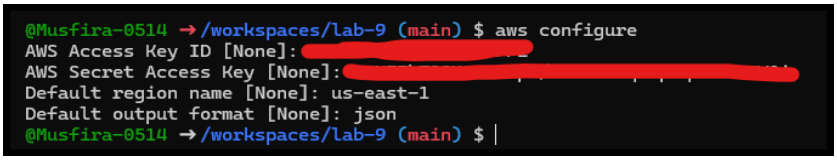
```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 60.2M  100 60.2M    0     0   179M      0 --:--:-- --:--:-- --:--:--  179M
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ ls -la
total 61740
drwxrwxrwx+ 3 codespace root          4096 Dec 30 18:39 .
drwxr-xrwx+ 5 codespace root          4096 Dec 30 18:16 ..
drwxrwxrwx+ 8 codespace root          4096 Dec 30 18:16 .git
-rw-rw-rw-  1 codespace root            39 Dec 30 18:16 README.md
-rw-rw-rw-  1 codespace codespace 63198016 Dec 30 18:39 awscliv2.zip
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ unzip awscliv2.zip
Archive:  awscliv2.zip
   creating: aws/
   creating: aws/dist/
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/install
  inflating: aws/README.md
   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/libuuid.so.1
  inflating: aws/dist/libreadline.so.6
  inflating: aws/dist/libtinfo.so.5
  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
  inflating: aws/dist/lib-dynload/_hashlib.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha3.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_blake2.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_md5.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha1.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_sha2.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_random.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_bisect.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/array.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_socket.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_opcode.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_json.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/binascii.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/resource.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_lzma.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_bz2.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_posixshmem.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/_multiprocessing.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/lib-dynload/pyexpat.cpython-313-x86_64-linux-gnu.so
```



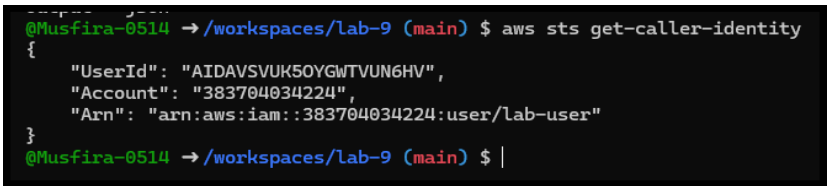
```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ sudo apt update
Get:1 https://dl.yarnpkg.com/debian stable InRelease
Get:2 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:3 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease [3600 B]
Get:4 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable InRelease [3961 B]
Get:5 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main amd64 Packages [77.6 kB]
Get:6 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main all Packages [643 B]
Get:7 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:8 https://dl.yarnpkg.com/debian stable/main amd64 Packages [11.8 kB]
Get:9 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable/main amd64 Packages [4557 B]
Get:10 https://dl.yarnpkg.com/debian stable/main all Packages [11.8 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2898 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [33.1 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1752 kB]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1183 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [35.9 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [2130 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1950 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [3059 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [49.5 kB]
Get:26 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [34.6 kB]
Fetched 35.5 MB in 5s (7811 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ sudo apt install unzip -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
unzip is already the newest version (6.0-28ubuntu4.1).
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws --version
aws-cli/2.32.25 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- AWS configure credentials





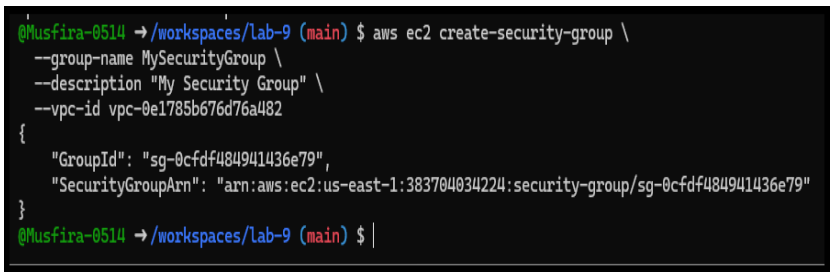
- Verify connectivity (aws sts get-caller-identity)



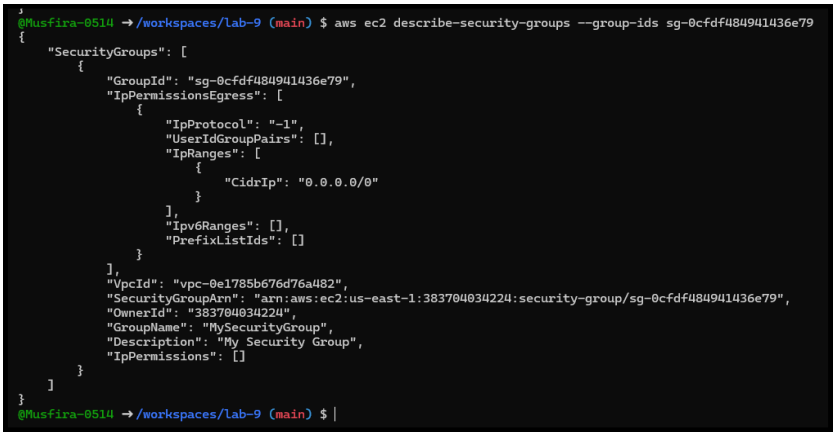
Task 3 — Security group creation & ingress rules

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

- Create security group



- Describe SG before ingress



- Codespace public IP

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ curl icanhazip.com
4.240.18.224
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Authorize SSH & HTTP rules

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 authorize-security-group-ingress   --group-id sg-0cfdf484941436e79
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0767fdee69aeb30a7",
            "GroupId": "sg-0cfdf484941436e79",
            "GroupOwnerId": "383704034224",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIpv4": "4.240.18.224/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:383704034224:security-group-rule/sgr-0767fdee69aeb30a7"
        }
    ]
}
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 authorize-security-group-ingress \
  --group-id sg-0cfdf484941436e79 \
  --protocol tcp \
  --port 22 \
  --cidr 4.240.18.224/32
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0cffb234a8a68ac66",
            "GroupId": "sg-0cfdf484941436e79",
            "GroupOwnerId": "383704034224",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv4": "4.240.18.224/32",
            "SecurityGroupRuleArn": "arn:aws:ec2:us-east-1:383704034224:security-group-rule/sgr-0cffb234a8a68ac66"
        }
    ]
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-security-groups --group-ids sg-0cfdf484941436e79
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0cfdf484941436e79",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0e1785b676d76a482",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:383704034224:security-group/sg-0cfdf484941436e79",
            "OwnerId": "383704034224",
            "GroupName": "MySecurityGroup",
            "Description": "My Security Group",
            "IpPermissions": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "4.240.18.224/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ]
        }
    ]
}
```

- Final SG verification

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-security-groups --group-ids sg-0cfdf484941436e79
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0cfdf484941436e79",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-0e1785b676d76a482",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:383704034224:security-group/sg-0cfdf484941436e79",
            "OwnerId": "383704034224",
            "GroupName": "MySecurityGroup",
            "Description": "My Security Group",
            "IpPermissions": [
                {
                    "IpProtocol": "tcp",
                    "FromPort": 80,
                    "ToPort": 80,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "4.240.18.224/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                },
                {
                    "IpProtocol": "tcp",
                    "FromPort": 22,
                    "ToPort": 22,
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "4.240.18.224/32"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ]
        }
    ]
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

Task 4 — Key pair creation, EC2 launch & SSH

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

- Create key pair

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 create-key-pair \
  --key-name MyED25519Key \
  --key-type ed25519 \
  --key-format pem \
  --query 'KeyMaterial' \
  --output text > MyED25519Key.pem
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ ls -l MyED25519Key.pem
-rw-rw-rw- 1 codespace codespace 388 Dec 30 21:25 MyED25519Key.pem
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Describe key pairs

```
-rw-rw-rw- 1 codespace codespace 388 Dec 30 21:25 MyED25519Key.pem
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-key-pairs
{
    "KeyPairs": [
        {
            "KeyPairId": "key-0b0ff9c2a18189f40",
            "KeyType": "ed25519",
            "Tags": [],
            "CreateTime": "2025-12-27T21:20:54.492000+00:00",
            "KeyName": "prod-nginx-key",
            "KeyFingerprint": "91HvOvmllZnyyH/jpSzBpc/j0hhYTQ7VwLgOmvXX9Bs="
        },
        {
            "KeyPairId": "key-01a056590739976e6",
            "KeyType": "ed25519",
            "Tags": [],
            "CreateTime": "2025-12-30T21:25:38.412000+00:00",
            "KeyName": "MyED25519Key",
            "KeyFingerprint": "jKNM3oxHIDuGLIcHUiccZC9r8Vdl4JDWUAj+9FtgpwQ="
        }
    ]
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Delete key pair (optional)

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 delete-key-pair --key-name MyED25519Key
{
    "Return": true,
    "KeyPairId": "key-01a056590739976e6"
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Launch EC2 instance

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 run-instances \
  --image-id ami-0c02fb55956c7d316 \
  --count 1 \
  --instance-type t3.micro \
  --key-name MyED25519Key \
  --security-group-ids sg-0cfdf4849411436e79 \
  --subnet-id subnet-0460444671b4c7475 \
  --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=MyServer}]"
{
    "ReservationId": "r-021c095eb1b7d7244",
    "OwnerId": "383704034224",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "0a92d476-08ce-4a83-8ccf-056998afcba1",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-12-30T21:35:11+00:00",
                        "AttachmentId": "eni-attach-02d851a1fbac091be",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-0cfdf4849411436e79",
                            "GroupName": "MySecurityGroup"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "02:e1:57:86:24:15",
                    "NetworkInterfaceId": "eni-0042bda3c2917b735",
                    "OwnerId": "383704034224",
                    "PrivateDnsName": "ip-172-31-0-151.ec2.internal",
                    "PrivateIpAddress": "172.31.0.151",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-172-31-0-151.ec2.internal",
                            "PrivateIpAddress": "172.31.0.151"
                        }
                    ],
                    "SourceDestCheck": true,
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Describe instance & public IP

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress]" \
  --output table
-----------------------------------------
|           DescribeInstances           |
+----------------------+----------------+
|  i-035e8154fbc024b89 |  44.200.155.93 |
+----------------------+----------------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- SSH permission error & fix

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ chmod 400 MyED25519Key.pem
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ ssh -i MyED25519Key.pem ec2-user@44.200.155.93
The authenticity of host '44.200.155.93 (44.200.155.93)' can't be established.
ED25519 key fingerprint is SHA256:h64BXj3DQJCYwEbmUjQMBanS75cbh2/ytDMpUfbGlnM.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.200.155.93' (ED25519) to the list of known hosts.

       Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
55 package(s) needed for security, out of 102 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-0-151 ~]$
```

- Stop/Start/Terminate instance

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 stop-instances --instance-ids i-035e8154fbc024b89
{
    "StoppingInstances": [
        {
            "InstanceId": "i-035e8154fbc024b89",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --instance-ids i-035e8154fbc024b89 \
  --query "Reservations[*].Instances[*].State.Name" \
  --output text
stopped
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 start-instances --instance-ids i-035e8154fbc024b89
{
    "StartingInstances": [
        {
            "InstanceId": "i-035e8154fbc024b89",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

Task 5 — AWS describe- commands

Objective: Inspect AWS resources.

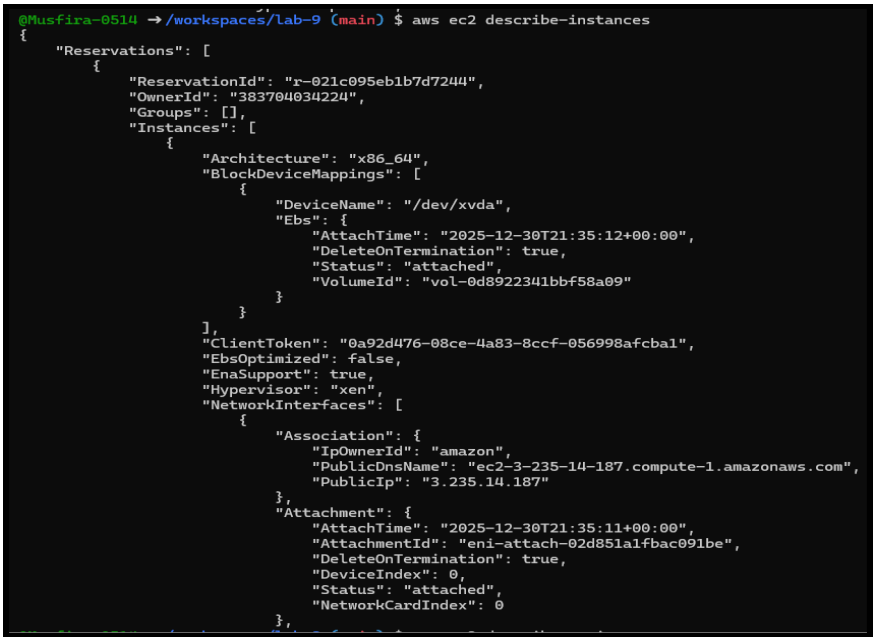
- aws ec2 describe-security-groups

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-security-groups
{
{
    "SecurityGroups": [
        {
            "GroupId": "sg-02d78977f60bf347d",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-09f6731d9869ccde1",
            "SecurityGroupArn": "arn:aws:ec2:us-east-1:383704034224:security-group/sg-02d78977f60bf347d",
            "OwnerId": "383704034224",
            "GroupName": "default",
            "Description": "default VPC security group",
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [
                        {
                            "UserId": "383704034224",
                            "GroupId": "sg-02d78977f60bf347d"
                        }
                    ],
                    "IpRanges": [],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ]
        },
        {
            "GroupId": "sg-0a745d4d5f53f30c9",
```

- aws ec2 describe-subnets

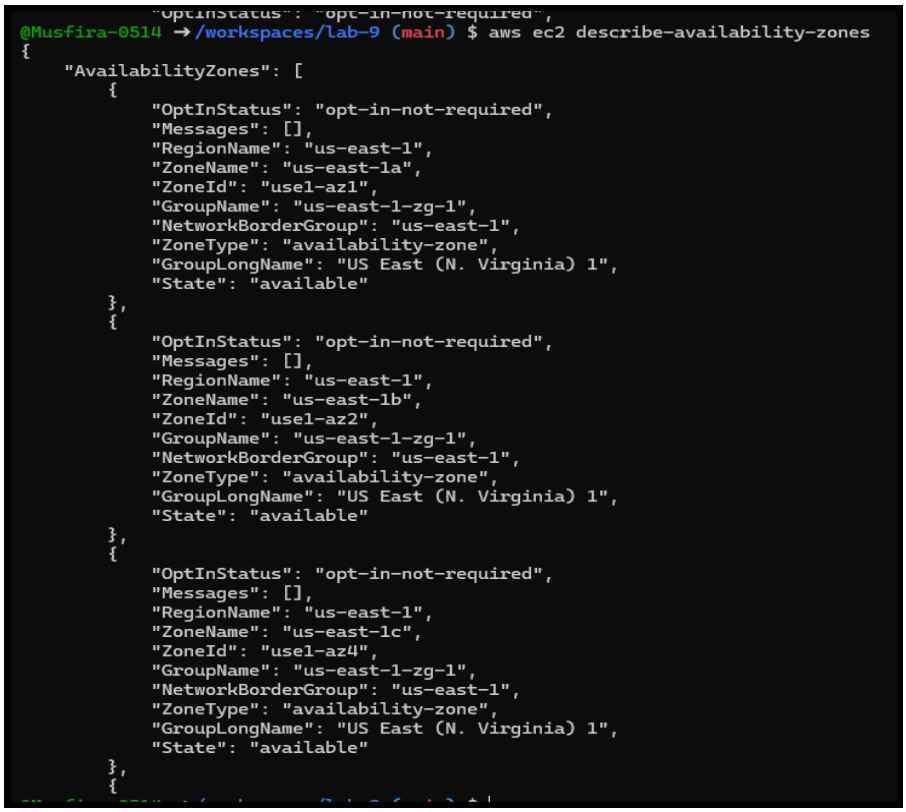
```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-subnets
{
    "Subnets": [
        {
            "AvailabilityZoneId": "use1-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "383704034224",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:us-east-1:383704034224:subnet/subnet-0460444671b4c7475",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0460444671b4c7475",
            "State": "available",
            "VpcId": "vpc-0e1785b676d76a482",
            "CidrBlock": "172.31.0.0/20",
            "AvailableIpAddressCount": 4090,
            "AvailabilityZone": "us-east-1a",
            "DefaultForAz": true,
            "MapPublicIpOnLaunch": true
        },
        {
            "AvailabilityZoneId": "use1-az5",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "383704034224",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:us-east-1:383704034224:subnet/subnet-06b91165ee5aa919a",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
```

- aws ec2 describe-instances



- aws ec2 describe-regions

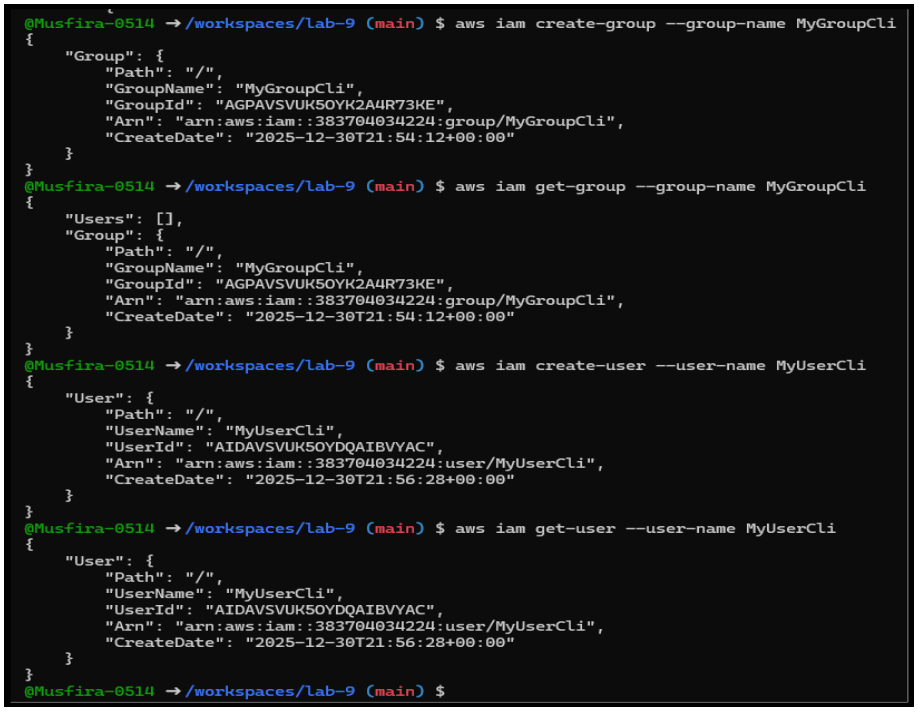
- aws ec2 describe-availability-zones



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

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

- Create group & user



```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam get-group --group-name MyGroupCli
{
    "Users": [
        {
            "Path": "/",
            "UserName": "MyUserCli",
            "UserId": "AIDAVSVUK5OYDQAIBVYAC",
            "Arn": "arn:aws:iam::383704034224:user/MyUserCli",
            "CreateDate": "2025-12-30T21:56:28+00:00"
        }
    ],
    "Group": {
        "Path": "/",
        "GroupName": "MyGroupCli",
        "GroupId": "AGPAVSVUK5OYK2A4R73KE",
        "Arn": "arn:aws:iam::383704034224:group/MyGroupCli",
        "CreateDate": "2025-12-30T21:54:12+00:00"
    }
}
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam add-user-to-group \
  --user-name MyUserCli \
  --group-name MyGroupCli
```

- Attach policy to group

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam list-policies \
  --query "Policies[?contains(PolicyName, 'EC2')].{Name:PolicyName}" \
  --output text
AmazonEC2FullAccess
AmazonEC2ReadOnlyAccess
AmazonElasticMapReduceforEC2Role
AmazonEC2RoleforDataPipelineRole
AmazonEC2ContainerServiceforEC2Role
AmazonEC2ContainerServiceRole
AmazonEC2RoleforAWSCodeDeploy
AmazonEC2RoleforSSM
CloudWatchActionsEC2Access
AmazonEC2ContainerRegistryReadOnly
AmazonEC2ContainerRegistryPowerUser
AmazonEC2ContainerRegistryFullAccess
AmazonEC2ContainerServiceAutoscaleRole
AmazonEC2SpotFleetAutoscaleRole
AWSElasticBeanstalkCustomPlatformforEC2Role
AmazonEC2ContainerServiceEventsRole
AmazonEC2SpotFleetTaggingRole
AWSEC2SpotServiceRolePolicy
AWSServiceRoleForEC2ScheduledInstances
AWSEC2SpotFleetServiceRolePolicy
AWSApplicationAutoscalingEC2SpotFleetRequestPolicy
AWSEC2FleetServiceRolePolicy
AWSAutoScalingPlansEC2AutoScalingPolicy
EC2InstanceConnect
AmazonEC2RolePolicyForLaunchWizard
EC2InstanceProfileForImageBuilder
EC2FleetTimeShiftableServiceRolePolicy
AmazonEC2RoleforAWSCodeDeployLimited
EC2InstanceProfileForImageBuilderECRContainerBuilds
AWSApplicationMigrationEC2Access
AWSEC2CapacityReservationFleetRolePolicy
EC2FastLaunchServiceRolePolicy
AmazonSSMManagedEC2InstanceDefaultPolicy
AWSFaultInjectionSimulatorEC2Access
EC2ImageBuilderLifecycleExecutionPolicy
AWSEC2VssSnapshotPolicy
EC2FastLaunchFullAccess
AmazonEC2ContainerRegistryPullOnly
DeclarativePoliciesEC2Report
AmazonEC2ImageReferencesAccessPolicy
AWSEC2CapacityManagerServiceRolePolicy
AWSEC2SqlHaServiceRolePolicy
AWSEC2SqlHaInstancePolicy
AWSLambdaManagedEC2ResourceOperator
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam list-policies \
  --query 'Policies[?PolicyName==`AmazonEC2FullAccess`].{Name:PolicyName, ARN:Arn}' \
  --output table
-------------------------------------------------------------------------
|                           ListPolicies                                |
+-----------------------------------------+-----------------------------+
|                   ARN                   |          Name               |
+-----------------------------------------+-----------------------------+
|  arn:aws:iam::aws:policy/AmazonEC2FullAccess |  AmazonEC2FullAccess  |
+-----------------------------------------+-----------------------------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam attach-group-policy \
  --group-name MyGroupCli \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam list-attached-group-policies --group-name MyGroupCli
{
    "AttachedPolicies": [
        {
            "PolicyName": "AmazonEC2FullAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AmazonEC2FullAccess"
        }
    ]
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Create login profile

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam create-login-profile \
  --user-name MyUserCli \
  --password [redacted] \
  --password-reset-required
{
    "LoginProfile": {
        "UserName": "MyUserCli",
        "CreateDate": "2025-12-30T22:04:46+00:00",
        "PasswordResetRequired": true
    }
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam attach-group-policy \
  --group-name MyGroupCli \
  --policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam detach-group-policy \
  --group-name MyGroupCli \
  --policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Create access key

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam create-access-key --user-name MyUserCli
{
    "AccessKey": {
        "UserName": "MyUserCli",
        "AccessKeyId": [redacted],
        "Status": "Active",
        "SecretAccessKey": [redacted],
        "CreateDate": "2025-12-30T22:07:47+00:00"
    }
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam list-access-keys --user-name MyUserCli
{
    "AccessKeyMetadata": [
        {
            "UserName": "MyUserCli",
            "AccessKeyId": [redacted],
            "Status": "Active",
            "CreateDate": "2025-12-30T22:07:47+00:00"
        }
    ]
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Test environment variable authentication

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ export AWS_ACCESS_KEY_ID=[redacted]
export AWS_SECRET_ACCESS_KEY=[redacted]
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ printenv | grep AWS_
AWS_SECRET_ACCESS_KEY=[redacted]
AWS_ACCESS_KEY_ID=[redacted]
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws iam get-user --user-name MyUserCli

An error occurred (AccessDenied) when calling the GetUser operation: User: arn:aws:iam::383704034224:
sed policy allows the iam:GetUser action
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

```
PS C:\Users\Musfi> aws sts get-caller-identity
{
    "UserId": "AIDAVSVUK5OYC4REAISW5",
    "Account": "383704034224",
    "Arn": "arn:aws:iam::383704034224:user/terraform_user"
}

PS C:\Users\Musfi>
```

Task 7 — Filters: Describe-instances with filters

- Filter by Tag

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --filters "Name=tag:Name,Values=MyServer" \
  --query "Reservations[*].Instances[*].PublicIpAddress" \
  --output text
3.235.14.187
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Filter by Instance type

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --filters "Name=instance-type,Values=t3.micro" \
  --query "Reservations[].Instances[].InstanceId" \
  --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-035e8154fbc024b89  |
+-----------------------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Filter by Subnet

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --filters "Name=subnet-id,Values=subnet-0460444671b4c7475" \
  --query "Reservations[*].Instances[*].InstanceId" \
  --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-035e8154fbc024b89  |
+-----------------------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Filter by VPC

```
+-----------------------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --query "Reservations[*].Instances[*].VpcId" \
  --output text
vpc-0e1785b676d76a482
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --filters "Name=vpc-id,Values=vpc-0e1785b676d76a482" \
  --query "Reservations[*].Instances[*].InstanceId" \
  --output table
-------------------------
|   DescribeInstances   |
+-----------------------+
|  i-035e8154fbc024b89  |
+-----------------------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

Task 8 — Query outputs formatted for reporting

- Instances with Name & IP

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --filters "Name=tag:Name,Values=MyServer" \
  --query "Reservations[*].Instances[*].[InstanceId,PublicIpAddress,Tags[?Key=='Name'].Value|[0]]" \
  --output table
-----------------------------------------------------
|                DescribeInstances                  |
+---------------------+---------------+-----------+
|  i-035e8154fbc024b89 |  3.235.14.187  |  MyServer |
+---------------------+---------------+-----------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Instance ID & State

```
+-----------------------+------------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --query "Reservations[*].Instances[*].[InstanceId,State.Name]" \
  --output table
-----------------------------------
|         DescribeInstances       |
+-----------------------+---------+
|  i-035e8154fbc024b89  |  running  |
+-----------------------+---------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Instance Type & AZ

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --query "Reservations[*].Instances[*].[InstanceId,InstanceType,Placement.AvailabilityZone]" \
  --output table
-----------------------------------------------------
|                DescribeInstances                  |
+---------------------+-----------+-------------+
|  i-035e8154fbc024b89 |  t3.micro |  us-east-1a  |
+---------------------+-----------+-------------+
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

Cleanup — Remove AWS resources

- Terminate instances

```
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 terminate-instances --instance-ids i-035e8154fbc024b89
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-035e8154fbc024b89",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

- Delete volumes/snapshots

```
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --query "Reservations[*].Instances[*].[InstanceId,State.Name]" \
  --output table
-----------------------------------------
|           DescribeInstances           |
+----------------------+----------------+
|  i-035e8154fbc024b89 |  shutting-down |
+----------------------+----------------+
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 describe-volumes --output table
----------------
|DescribeVolumes|
+---------------+
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 describe-snapshots --owner-ids self --output table
------------------
|DescribeSnapshots|
+-----------------+
@Musfira-0514 → /workspaces/lab-9 (main) $
```

- Delete SG & key pair

```
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 delete-security-group --group-id sg-0cfdf484941436e79
{
    "Return": true,
    "GroupId": "sg-0cfdf484941436e79"
}
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 delete-key-pair --key-name MyED25519Key
{
    "Return": true,
    "KeyPairId": "key-005a5248599ca763a"
}
@Musfira-0514 → /workspaces/lab-9 (main) $
```

- Delete IAM users & groups

```
@Musfira-0514 → /workspaces/lab-9 (main) $ aws iam delete-access-key \
  --user-name MyUserCli \
  --access-key-id ...
@Musfira-0514 → /workspaces/lab-9 (main) $ aws iam delete-login-profile --user-name MyUserCli
@Musfira-0514 → /workspaces/lab-9 (main) $ aws iam remove-user-from-group \
  --user-name MyUserCli \
  --group-name MyGroupCli
@Musfira-0514 → /workspaces/lab-9 (main) $ aws iam detach-group-policy \
  --group-name MyGroupCli \
  --policy-arn arn:aws:iam::aws:policy/AmazonEC2FullAccess
@Musfira-0514 → /workspaces/lab-9 (main) $ aws iam detach-group-policy \
  --group-name MyGroupCli \
  --policy-arn arn:aws:iam::aws:policy/IAMUserChangePassword

An error occurred (NoSuchEntity) when calling the DetachGroupPolicy operation: Policy arn:aws:
@Musfira-0514 → /workspaces/lab-9 (main) $ aws iam delete-user --user-name MyUserCli
@Musfira-0514 → /workspaces/lab-9 (main) $
```

- Final verification

```
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --query "Reservations[*].Instances[*].[InstanceId,State.Name]" \
  --output table
-----------------------------------------
|           DescribeInstances           |
+----------------------+----------------+
|  i-035e8154fbc024b89 |  terminated    |
+----------------------+----------------+
```

```
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 describe-instances \
  --filters Name=instance-state-name,Values=pending,running,stopping,stopped \
  --query "Reservations[*].Instances[*].[InstanceId,State.Name]" \
  --output table
```


@Musfira-0514 ➜ /workspaces/lab-9 (main) $ aws ec2 describe-instances --output tabl
DescribeInstances
Reservations
OwnerId 383704034224
ReservationId r-021c095eb1b7d7244
Instances
AmiLaunchIndex 0
Architecture x86_64
ClientToken 0a92d476-08ce-4a83-8ccf-056998afcba1
CurrentInstanceBootMode legacy-bios
EbsOptimized False
EnaSupport True
Hypervisor xen
ImageId ami-0c02fb55956c7d316
InstanceId i-035e8154fbc024b89
InstanceType t3.micro
KeyName MyED25519Key
LaunchTime 2025-12-30T21:44:18+00:00
PlatformDetails Linux/UNIX
PrivateDnsName
PublicDnsName
RootDeviceName /dev/xvda
RootDeviceType ebs
StateTransitionReason User initiated (2025-12-30 22:25:06 GMT)
UsageOperation RunInstances
UsageOperationUpdateTime 2025-12-30T21:35:11+00:00
VirtualizationType hvm
CapacityReservationSpecification
CapacityReservationPreference open
CpuOptions
CoreCount 1
ThreadsPerCore 2
EnclaveOptions
Enabled False
HibernationOptions
Configured False
MaintenanceOptions
AutoRecovery default
RebootMigration default
MetadataOptions
HttpEndpoint enabled
HttpProtocolIpv6 disabled
HttpPutResponseHopLimit 1
HttpTokens optional
InstanceMetadataTags disabled
State pending