Name : Zuha Irfan

Roll No : 2023-BSE-073

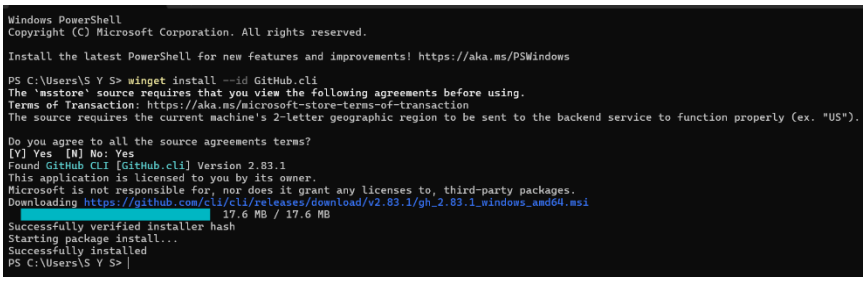
Subject: Cloud Computing

Semester & Section : V-B

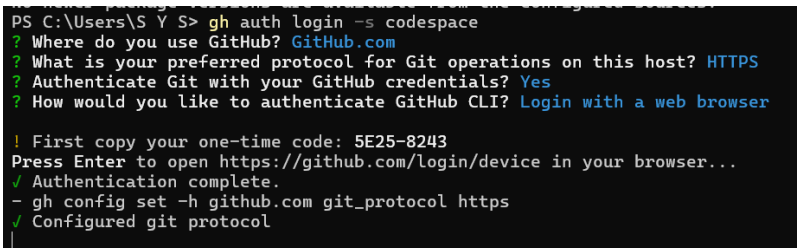
Lab : 09

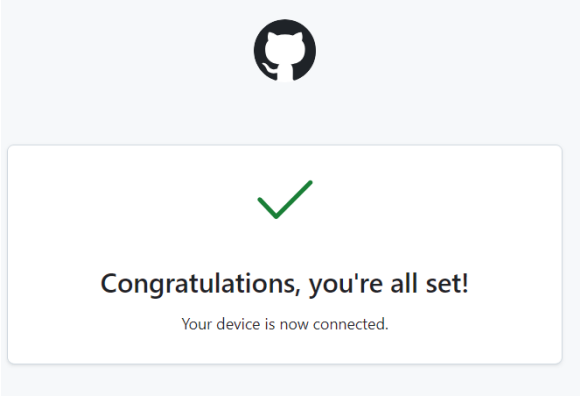
Task 1 — GitHub CLI, Codespace setup and authentication

- task1_gh_install.png



- task1_gh_auth_login.png





- 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:\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/_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/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
AWS Access Key ID [****************dmin]: 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

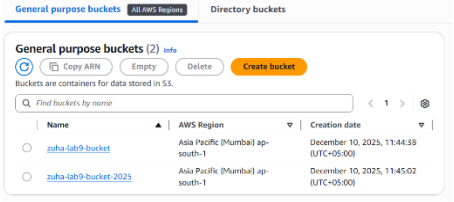
```
@Zuha-Irfan ➜ /workspaces/Lab9 (main) $ aws ec2 describe-vpcs --output table
-----------------------------------------------------------------------------------------------------------------------------------
|                                                          DescribeVpcs                                                           |
+---------------------------------------------------------------------------------------------------------------------------------+
|                                                            Vpcs                                                                 |
+---------------+-----------------------+-------------------+-------------+---------------+-------------+---------------------+
|   CidrBlock   |   DhcpOptionsId       | InstanceTenancy   | IsDefault   |   OwnerId     |   State     |       VpcId         |
+---------------+-----------------------+-------------------+-------------+---------------+-------------+---------------------+
|  172.31.0.0/16|  dopt-0016e1434ba594b91 |  default          |  True       |  051942114323 |  available  |  vpc-03ed0c2085c712d1a  |
+---------------+-----------------------+-------------------+-------------+---------------+-------------+---------------------+
||                                                  BlockPublicAccessStates                                                      ||
|+-----------------------------------------------------------------------------------------------+-------------------------------+|
||  InternetGatewayBlockMode                                                                     |  off                          ||
|+-----------------------------------------------------------------------------------------------+-------------------------------+|
||                                                  CidrBlockAssociationSet                                                      ||
|+-----------------------------------------------------------------------------------------------+-------------------------------+|
||                                  AssociationId                                                |             CidrBlock         ||
|+-----------------------------------------------------------------------------------------------+-------------------------------+|
||  vpc-cidr-assoc-0770d6795df7edd89                                                             |  172.31.0.0/16                ||
|+-----------------------------------------------------------------------------------------------+-------------------------------+|
|||                                                  CidrBlockState                                                             |||
||+-----------------------------------------------------------------------------------------------+-------------------------------+||
|||  State                                                                                        |  associated                   |||
||+-----------------------------------------------------------------------------------------------+-------------------------------+||
@Zuha-Irfan ➜ /workspaces/Lab9 (main) $
```

- 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-03ed0c2085
c712d1a
{
    "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
```

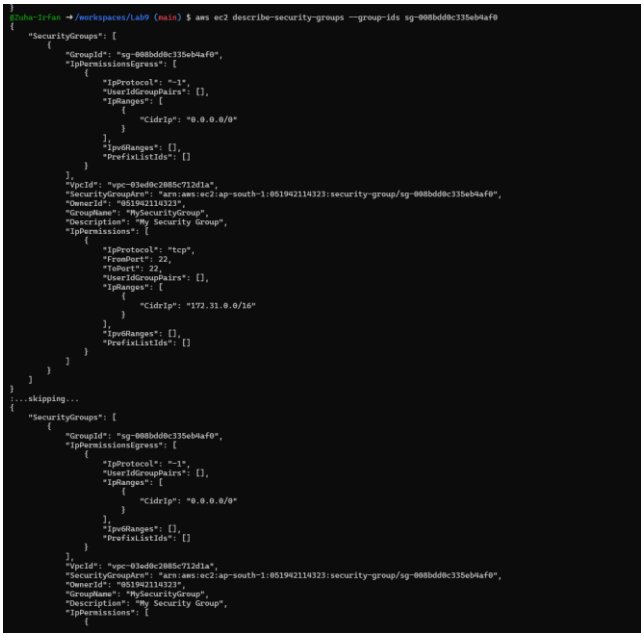


- task3_codespace_public_ip.png

- task3_authorize_ssh_and_describe.png

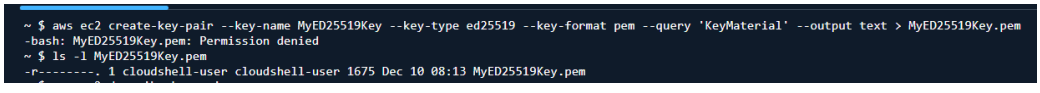
- task3_authorize_http_and_describe.png

- task3_describe_sg_final.png



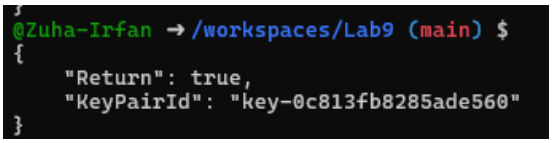
Task 4 — Create a key pair, describe key pairs, and launch EC2 instanc

- task4_create_keypair_output.png



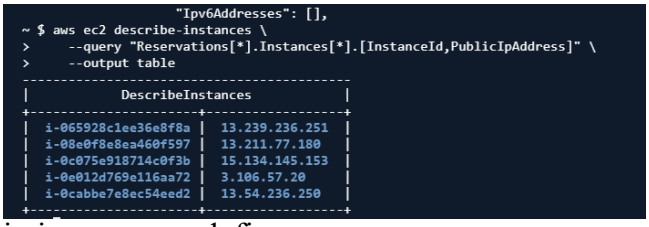
- task4_describe_keypairs.png

- task4_delete_keypair_optional.png

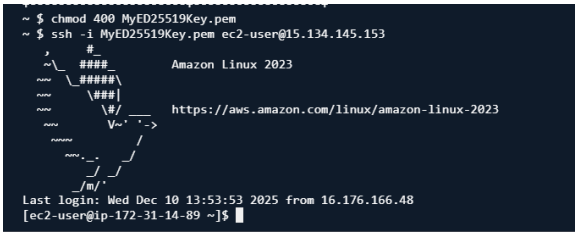


- task4_run_instances_output.png

- task4_describe_instances_public_ip.png



- 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

- task5_describe_vpcs.png

- task5_describe_subnets.png

- task5_describe_instances.png

- task5_describe_regions.png

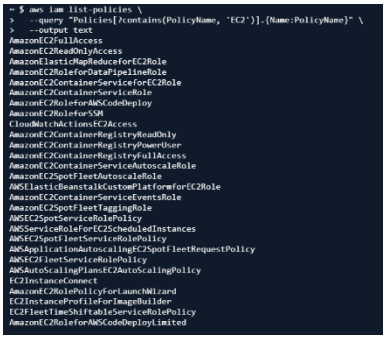
- task5_describe_availability_zones.png

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

- task6_create_group_and_user.png

- task6_add_user_to_group_and_verify.png

- task6_policy_list_and_attach.png



```
~ $ 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": "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",
        "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

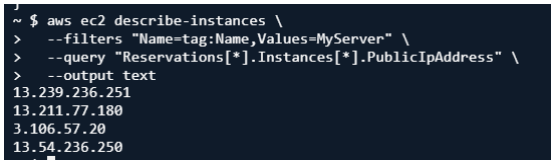
```
~ $ export AWS_ACCESS_KEY_ID=AKIAQYF75YQJ3OKLI3U3
~ $ export AWS_SECRET_ACCESS_KEY=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
~ $ export AWS_DEFAULT_REGION=ap-southeast-2
~ $ printenv | grep AWS_
AWS_CONTAINER_CREDENTIALS_FULL_URI=http://localhost:1338/latest/meta-data/container/security-credentials
AWS_EXECUTION_ENV=CloudShell
AWS_EC2_METADATA_DISABLED=true
AWS_DEFAULT_REGION=ap-southeast-2
AWS_REGION=ap-southeast-2
SET_DNF_REGION_SCRIPT=env | grep -m 1 AWS_REGION | grep -Eo '[a-z0-9-]*' | sudo tee /etc/dnf/vars/awsregion
AWS_PAGER=less -K
AWS_SDK_JS_SUPPRESS_MAINTENANCE_MODE_MESSAGE=true
AWS_SECRET_ACCESS_KEY=CEZUPq/1uA5o61JNsZH6Mebe9JN2RmhgYSoM821V
AWS_CONTAINER_AUTHORIZATION_TOKEN=DObZIr6BfCC+4k7aPIFvW6fab1KuBvnAEbgVvXAq828=
AWS_ACCESS_KEY_ID=AKIAQYF75YQJ3OKLI3U3
AWS_TOOLING_USER_AGENT=AWS-CloudShell/2025.10.21
~ $ AWS_ACCESS_KEY_ID=AKIAQYF75YQJ3OKLI3U3
~ $ AWS_SECRET_ACCESS_KEY=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
~ $ 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

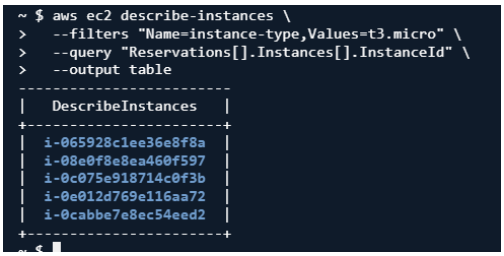
```
~ $ unset AWS_ACCESS_KEY_ID
~ $ unset AWS_SECRET_ACCESS_KEY
~ $ unset AWS_DEFAULT_REGION
~ $ aws iam get-user --user-name MyUserCli
{
    "User": {
        "Path": "/",
        "UserName": "MyUserCli",
        "UserId": "AIDAQYF75YQJ4MX4VHGCB",
        "Arn": "arn:aws:iam::051942114323:user/MyUserCli",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

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

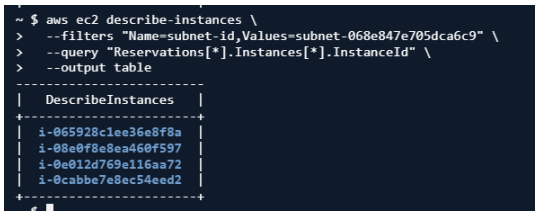
- task7_filter_by_tag_public_ip.png



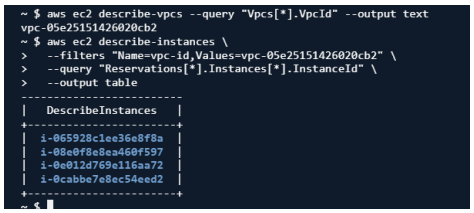
- task7_filter_by_instance_type.png



- task7_filter_by_subnet.png



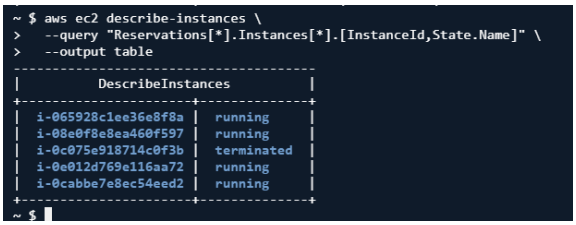
- task7_filter_by_vpc.png



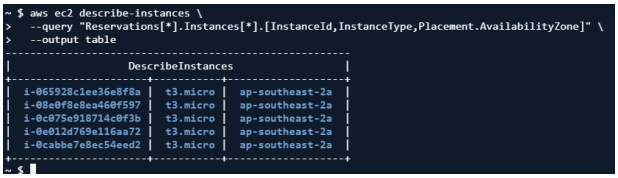
Task 8 — Use --query to format outputs for reporting

- task8_query_table_instances_name_ip.png

- task8_query_table_instance_state.png



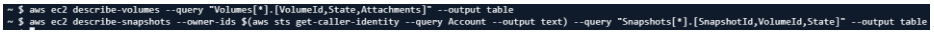
- task8_query_table_instance_type_az.png



Cleanup — Remove resources to avoid charges

- cleanup_terminate_instance.png

- cleanup_delete_volumes_snapshots.png



- cleanup_delete_security_group_and_keypair.png

- cleanup_iam_users_deleted.png

- cleanup_summary.png

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

```
~ $ aws ec2 describe-security-groups
{
    "SecurityGroups": [
        {
            "GroupId": "sg-0e867a857917e4df5",
            "IpPermissionsEgress": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [],
                    "IpRanges": [
                        {
                            "CidrIp": "0.0.0.0/0"
                        }
                    ],
                    "Ipv6Ranges": [],
                    "PrefixListIds": []
                }
            ],
            "VpcId": "vpc-05e25151426020cb2",
            "SecurityGroupArn": "arn:aws:ec2:ap-southeast-2:051942114323:security-group/sg-0e867a85791?e4df5",
            "OwnerId": "051942114323",
            "GroupName": "default",
            "Description": "default VPC security group",
            "IpPermissions": [
                {
                    "IpProtocol": "-1",
                    "UserIdGroupPairs": [
                        {
                            "UserId": "051942114323",
                            "GroupId": "sg-0e867a857917e4df5"
                        }
```

```
~ $ aws iam list-users
{
    "Users": [
        {
            "Path": "/",
            "UserName": "Admin",
            "UserId": "AIDAQYF75YQJZIJXFDSQK",
            "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": []
}
```