Name : Zuha Irfan

Roll No : 2023-BSE-073

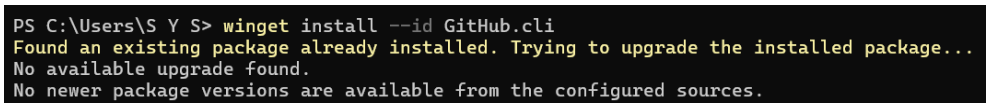
Subject: Cloud Computing

Semester & Section : V-B

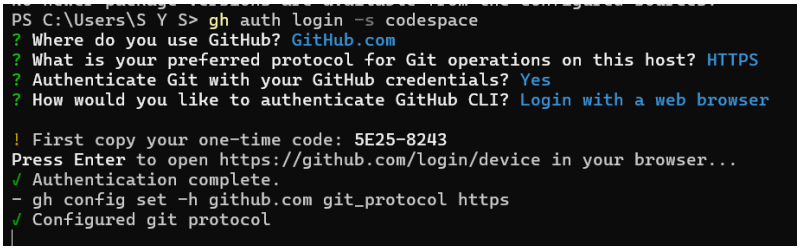
Lab :10

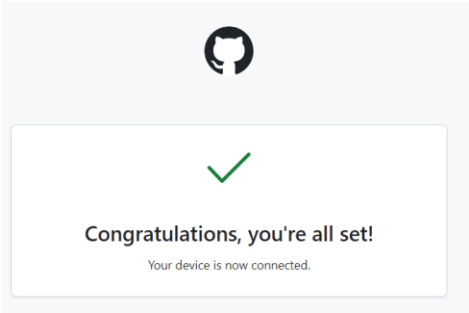
Task 1 — GitHub CLI Codespace Setup & Authentication

- task1_gh_install.png

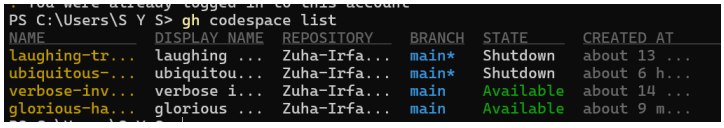


- task1_gh_auth_login.png





- task1_codespace_list.png



- task1_codespace_ssh_connected.png

```
PS C:\Users\S Y S\Lab10> gh codespace ssh -c glorious-halibut-q75qrjggr59w39gr7
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/LAB10 (main) $
```

Task 2 — Install AWS CLI, Terraform CLI, Provider Setup

- task2_aws_install_and_version.png

```
@Zuha-Irfan ➜ /workspaces/LAB10 (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 60.0M  100 60.0M    0     0   170M      0 --:--:-- --:--:-- --:--:--  171M
```

```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ unzip awscliv2.zip
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
  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
```

```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
@Zuha-Irfan ➜ /workspaces/LAB10 (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

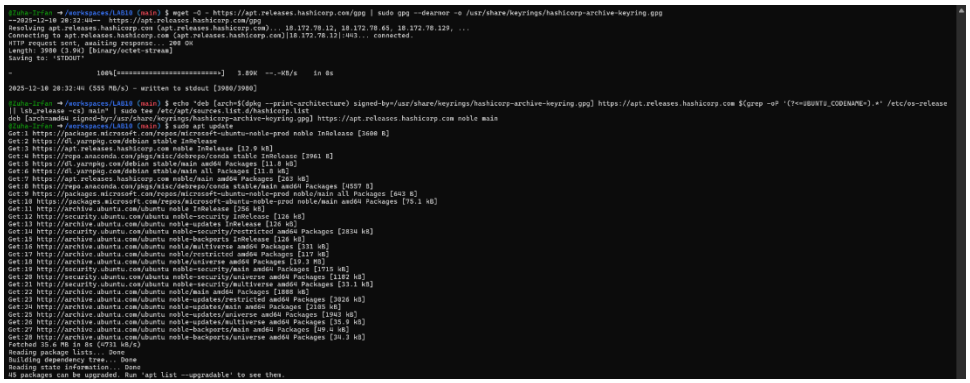
```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ aws configure
AWS Access Key ID [None]: AKIAQYF75YQJWLUAYQWO
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]: json
```

```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ cat ~/.aws/credentials
[default]
aws_access_key_id = AKIAQYF75YQJWLUAYQWO
aws_secret_access_key = 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
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ cat ~/.aws/config
[default]
region = ap-south-1
output = json
```

- task2_aws_get_caller_identity.png

```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ aws sts get-caller-identity
{
    "UserId": "AIDAQYF75YQJZIJXFDSQK",
    "Account": "051942114323",
    "Arn": "arn:aws:iam::051942114323:user/Admin"
}
```

- task2_terraform_install_and_version.png



```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ sudo apt install terraform
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 45 not upgraded.
Need to get 30.6 MB of archives.
After this operation, 101 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com noble/main amd64 terraform amd64 1.14.1-1 [30.6 MB]
Fetched 30.6 MB in 0s (172 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 58631 files and directories currently installed.)
Preparing to unpack .../terraform_1.14.1-1_amd64.deb ...
Unpacking terraform (1.14.1-1) ...
Setting up terraform (1.14.1-1) ...
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ which terraform
/usr/bin/terraform
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ terraform --version
Terraform v1.14.1
on linux_amd64
```

- task2_provider_file_creation.png

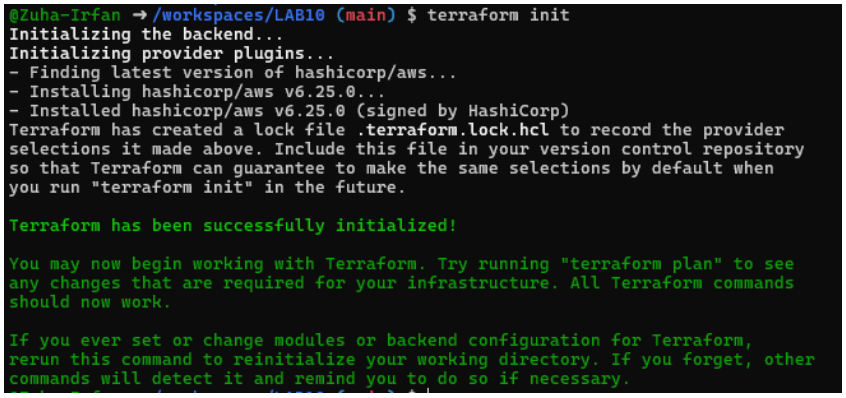
```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ vim main.tf
```

- task2_provider_block.png

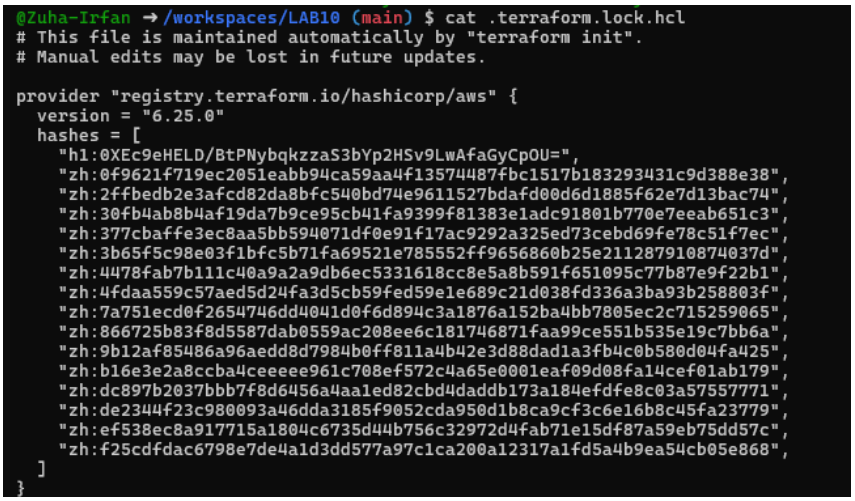
```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
i}
~
~
~
```

- task2_vim_save_main_tf.png

- task2_terraform_init_output.png



- task2_terraform_lock_hcl.png



- task2_terraform_dir_ls.png

Task 3 — VPC/Subnet Creation, Initialization, Verification

- task3_main_tf_resource_add.png

```
provider "aws" {
  region = "ap-south-1"
}

resource "aws_vpc" "development_vpc" {
  cidr_block = "10.0.0.0/16"
}

resource "aws_subnet" "dev_subnet_1" {
  vpc_id     = aws_vpc.development_vpc.id
  cidr_block = "10.0.10.0/24"
  availability_zone = "ap-south-1a"
}
```

```
~
:wq
```

- task3_terraform_apply_vpc_subnet.png

```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ terraform apply
aws_vpc.development_vpc: Refreshing state... [id=vpc-00cbeded203de2207]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_subnet.dev_subnet_1 will be created
  + resource "aws_subnet" "dev_subnet_1" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "ap-south-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "10.0.10.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "ap-south-1"
      + tags_all                                       = (known after apply)
      + vpc_id                                         = "vpc-00cbeded203de2207"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 0s [id=subnet-0cdb18650aba92d3c]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

- task3_aws_cli_verify_subnet.png

```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ aws ec2 describe-subnets --filter "Name=subnet-id,Values=subnet-0cdb18650aba92d3c"
{
    "Subnets": [
        {
            "AvailabilityZoneId": "aps1-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "051942114323",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:ap-south-1:051942114323:subnet/subnet-0cdb18650aba92d3c",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0cdb18650aba92d3c",
            "State": "available",
            "VpcId": "vpc-00cbeded203de2207",
            "CidrBlock": "10.0.10.0/24",
            "AvailableIpAddressCount": 251,
            "AvailabilityZone": "ap-south-1a",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": false
        }
    ]
}
```

- task3_aws_cli_verify_vpc.png

```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ aws ec2 describe-vpcs --filter "Name=vpc-id,Values=vpc-00cbeded203de2207"
{
    "Vpcs": [
        {
            "OwnerId": "051942114323",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0ea402f379704f5dc",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-00cbeded203de2207",
            "State": "available",
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-0016e1434ba594b91"
        }
    ]
}
```

Task 4 — Data Source, Targeted Destroy, Tags

- task4_main_tf_datasource_resource_add.png

```
}
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ vim main.tf
```

```
provider "aws" {
  region = "ap-south-1"
}

# VPC created by Terraform
resource "aws_vpc" "development_vpc" {
  cidr_block = "10.0.0.0/16"
}

# Subnet inside the Terraform VPC
resource "aws_subnet" "dev_subnet_1" {
  vpc_id            = aws_vpc.development_vpc.id
  cidr_block        = "10.0.10.0/24"
  availability_zone = "ap-south-1a"
}

# Data source: get the default VPC
data "aws_vpc" "existing_vpc" {
  default = true
}

# Subnet inside the existing default VPC
resource "aws_subnet" "dev_subnet_1_existing" {
  vpc_id            = data.aws_vpc.existing_vpc.id
  cidr_block        = "172.31.48.0/24"
  availability_zone = "ap-south-1a"   # FIXED
}
~
```

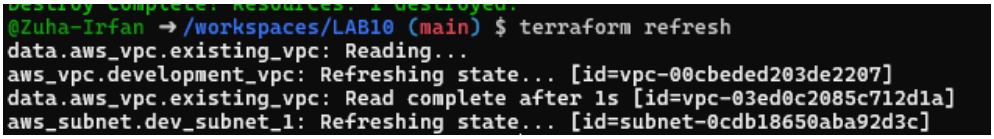
```
~
:wq
```

- task4_terraform_apply_datasource_resource.png

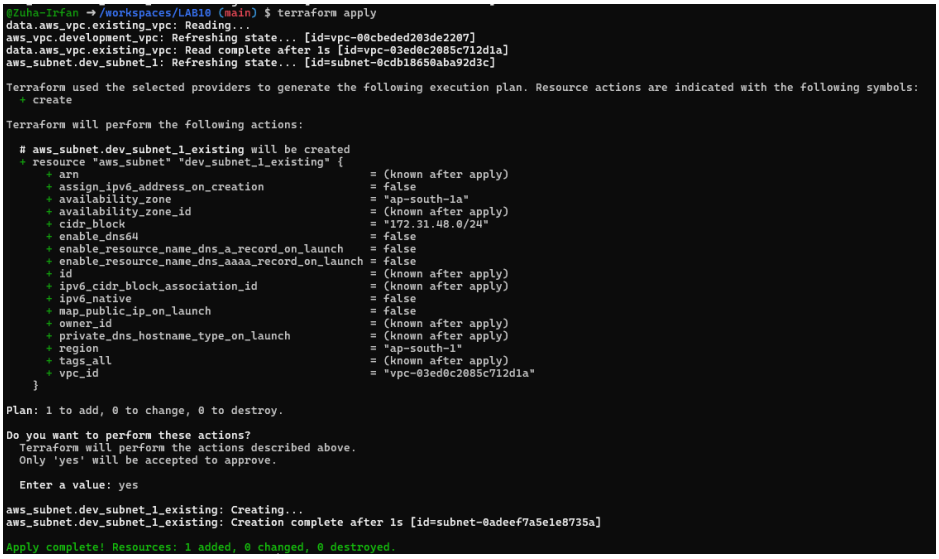
```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00cbeded203de2207]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-03ed0c2085c712d1a]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cdb18650aba92d3c]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_subnet.dev_subnet_1_existing will be created
  + resource "aws_subnet" "dev_subnet_1_existing" {
      + arn                                            = (known after apply)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "ap-south-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "172.31.48.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "ap-south-1"
      + tags_all                                       = (known after apply)
      + vpc_id                                         = "vpc-03ed0c2085c712d1a"
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_subnet.dev_subnet_1_existing: Creating...
aws_subnet.dev_subnet_1_existing: Creation complete after 1s [id=subnet-060d922ba78eb99aa]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

- task4_terraform_destroy_targeted.png

- task4_terraform_refresh_state.png

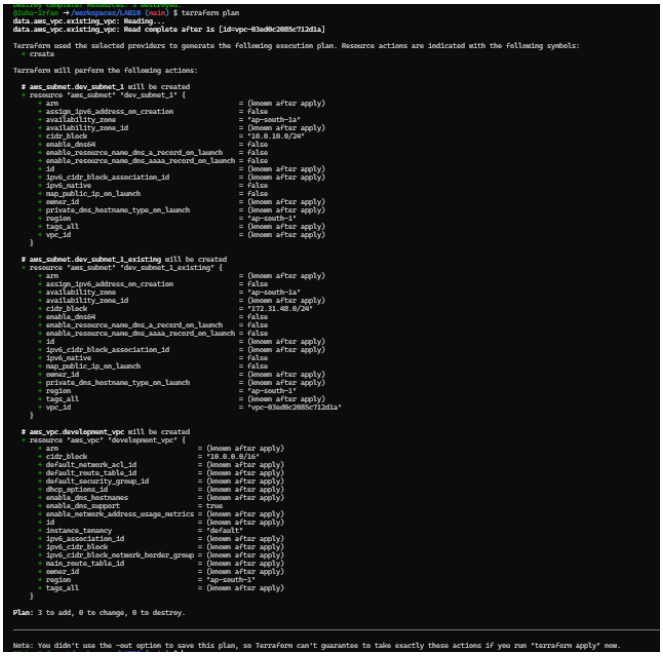


- task4_terraform_apply_after_refresh.png

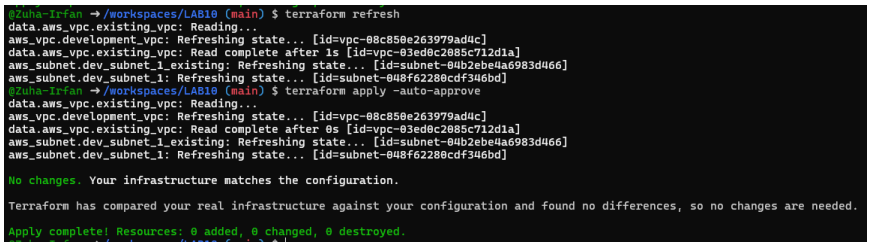


- task4_terraform_destroy_all.png

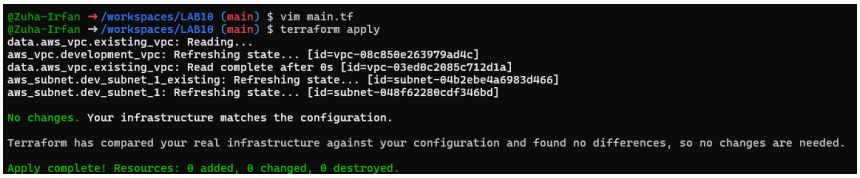
- task4_terraform_plan_output.png



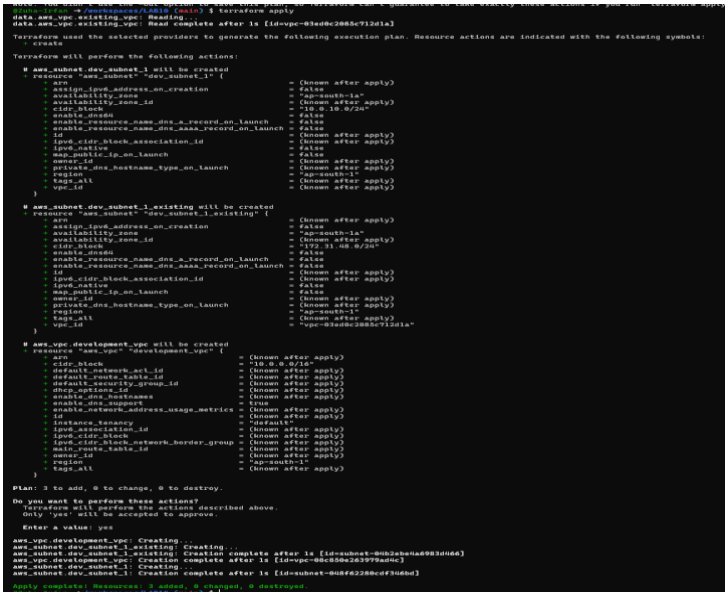
- task4_terraform_apply_after_destroy.png



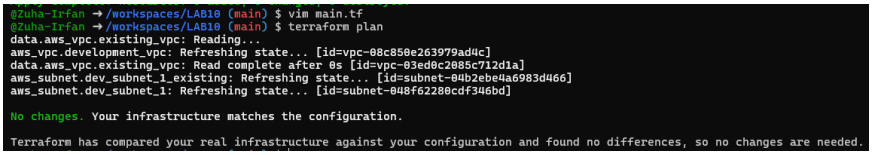
- task4_main_tf_tagging.png



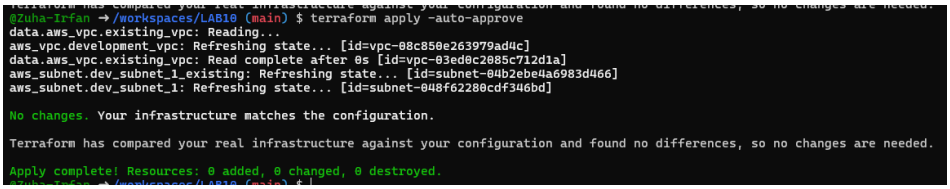
- task4_terraform_apply_tagging.png



- task4_terraform_plan_remove_tag.png



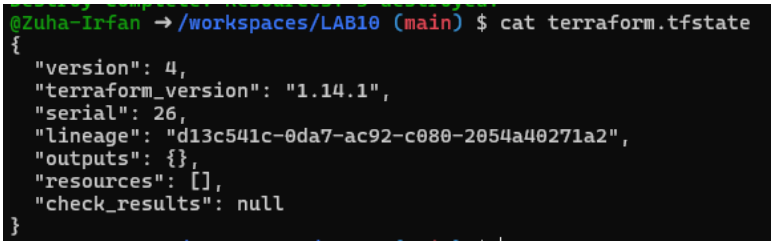
- task4_terraform_apply_remove_tag.png



Task 5 — State File Inspection & Terraform State Commands

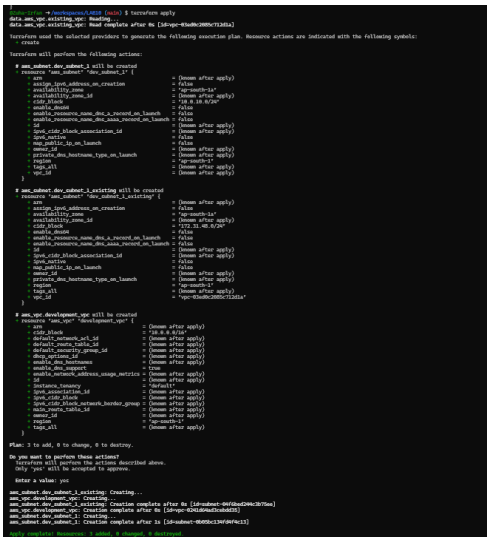
- task5_terraform_destroy.png

- task5_terraform_state_file_empty.png



- task5_terraform_state_backup_prev.png

- task5_terraform_apply_recreated.png



- task5_terraform_state_file_populated.png

- task5_terraform_state_backup_empty.png

```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 26,
  "lineage": "d13c541c-0da7-ac92-c080-2054a40271a2",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

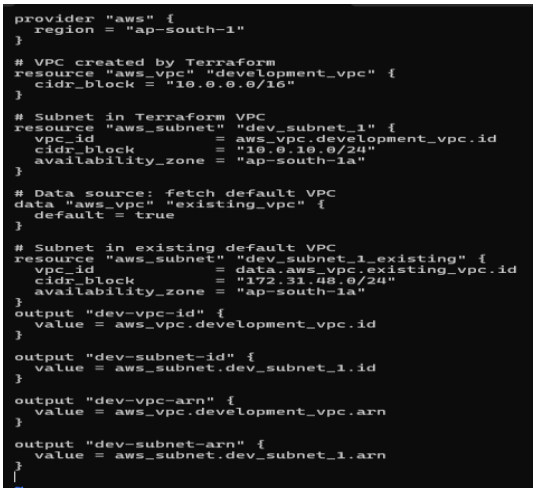
- task5_terraform_state_list.png

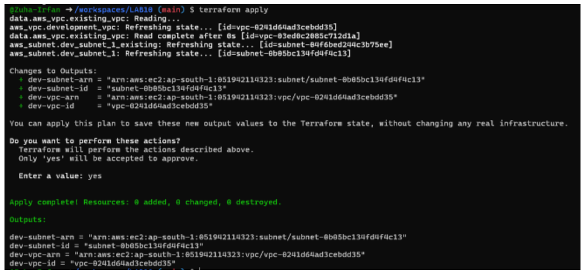
```
@Zuha-Irfan ➜ /workspaces/LAB10 (main) $ terraform state list
data.aws_vpc.existing_vpc
aws_subnet.dev_subnet_1
aws_subnet.dev_subnet_1_existing
aws_vpc.development_vpc
```

- task5_terraform_state_show_resource.png

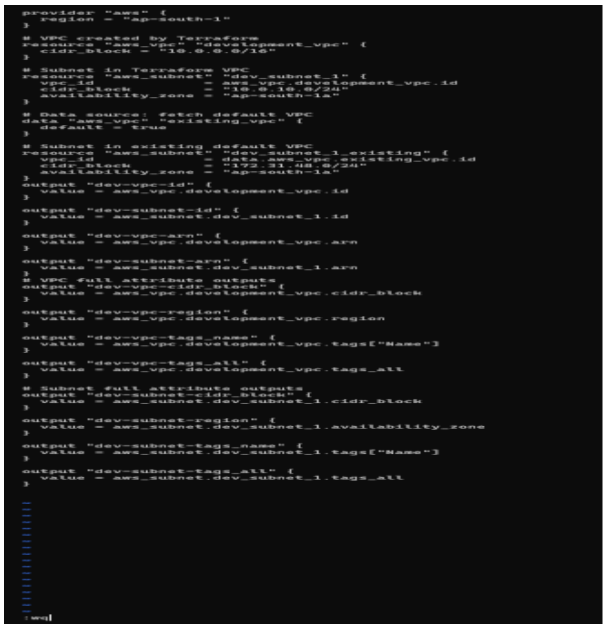
Task 6 — Terraform Outputs & Attributes Reporting

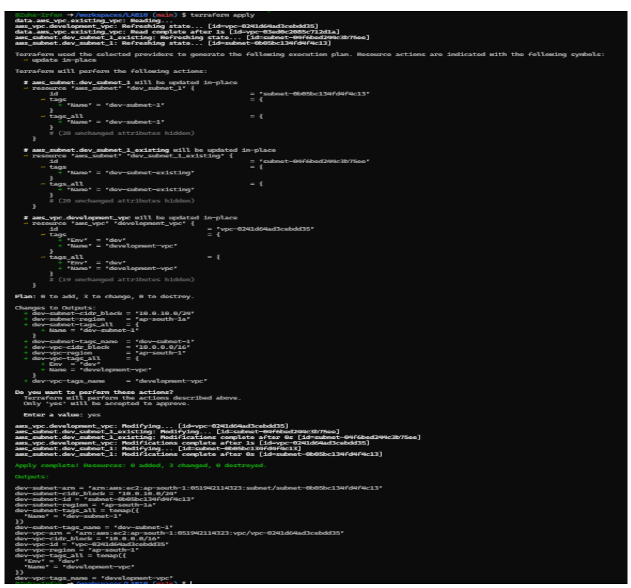
- task6_terraform_outputs_basic.png





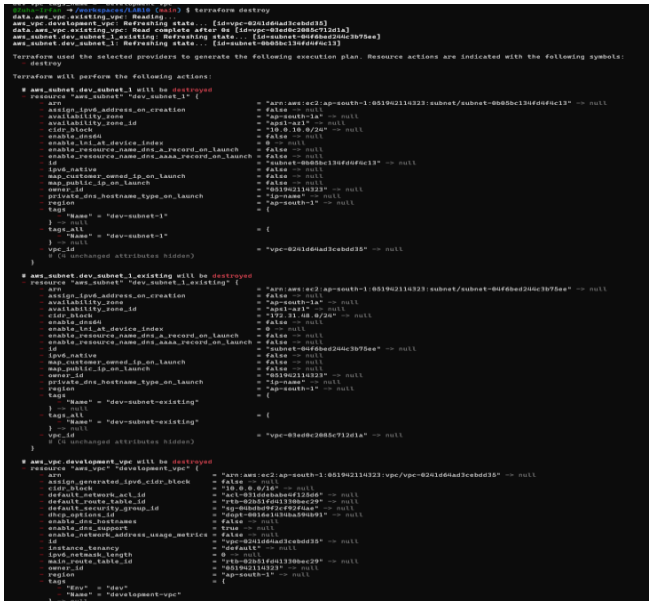
- task6_expanded_outputs.png





Cleanup — Delete Resources & State Verification

- cleanup_destroy_resources.png



- cleanup_state_files.png