

**Name : Zuha Irfan**

**Roll No : 2023-BSE-073**

**Subject: Cloud Computing**

**Semester & Section : V-B**

**Lab :10**

## **Task 1 — GitHub CLI Codespace Setup & Authentication**

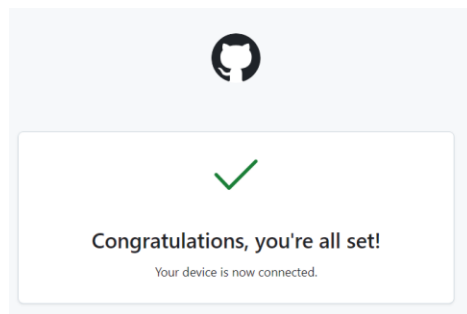
- task1\_gh\_install.png

```
PS C:\Users\S Y S> winget install --id GitHub.cli
Found an existing package already installed. Trying to upgrade the installed package...
No available upgrade found.
No newer package versions are available from the configured sources.
```

- 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

```
! You were already logged in to this account
PS C:\Users\S Y S> gh codespace list
NAME      DISPLAY_NAME  REPOSITORY  BRANCH  STATE      CREATED_AT
laughing-tr... laughing ...  Zuha-Irfa... main*   Shutdown   about 13 ...
ubiquitous-... ubiquitous... Zuha-Irfa... main*   Shutdown   about 6 h...
verbose-inv... verbose i...  Zuha-Irfa... main    Available  about 14 ...
glorious-ha... glorious ...  Zuha-Irfa... main    Available  about 9 m...
```

- 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
           % Done   0         0      0     0         0         0      0     0
100 60.0M 100 60.0M    0     0  170M      0  0:00:03 0:00:03 0:00:00 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/_awscli.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

```

```

  inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
@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]: AKIAQYF75YQJWLUAYQW0
AWS Secret Access Key [None]: Da7Vm9x1Ba+pfBN2G/Eudyuy/ksIUQGRdBcz/dGu
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 = AKIAQYF75YQJWLUAQWOW
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": "AIDAQYF75YQJZIJXFDSSQK",
  "Account": "051942114323",
  "Arn": "arn:aws:iam::051942114323:user/Admin"
}
```

- task2\_terraform\_install\_and\_version.png

```
@Zuha-Irfan → /workspaces/LAB10 (main) $ sudo apt -g - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
--2023-12-16 20:22:46-- https://apt.releases.hashicorp.com/gpg
Hashicorp apt releases hashicorp.com [apt.releases.hashicorp.com], 18.172.78.12, 18.172.78.43, 18.172.78.129, ...
Connecting to apt.releases.hashicorp.com [apt.releases.hashicorp.com]:18.172.78.13:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3986 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

-
100%[*****] 3.80K --.-KB/s in 0s

2023-12-16 20:22:46 (355 MB/s) - written to stdout [3986/3986]

@Zuha-Irfan → /workspaces/LAB10 (main) $ echo deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(grep -oP '(?=<u>Ubuntu</u>)' /etc/os-release | sed 's/Ubuntu/signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
deb [arch=amd64 signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
@Zuha-Irfan → /workspaces/LAB10 (main) $ sudo apt update
Get:1 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease [3608 B]
Get:2 https://dl.yunpao.com/debian stable InRelease
Get:3 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]
Get:4 https://repo.anaconda.com/repos/anaconda-stable InRelease [3961 B]
Get:5 https://dl.yunpao.com/debian stable/main amd64 Packages [13.1 kB]
Get:6 https://repo.anaconda.com/repos/anaconda-stable/main all Packages [13.9 kB]
Get:7 https://apt.releases.hashicorp.com noble/main amd64 Packages [163 kB]
Get:8 https://repo.anaconda.com/repos/anaconda-stable/main amd64 Packages [1027 B]
Get:9 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main all Packages [567 B]
Get:10 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main amd64 Packages [75.1 kB]
Get:11 http://archive.ubuntu.com/ubuntu noble InRelease [246 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:13 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2816 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [531 kB]
Get:17 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [517 kB]
Get:18 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [13.2 MB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [7115 kB]
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1282 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1388 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [125.1 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [3026 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1494 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [15.9 kB]
Get:26 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [40.4 kB]
Get:27 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [30.3 kB]
Fetched 53.6 MB in 8s (6711 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
49 packages can be upgraded. Run 'apt list --upgradable' to see them.

@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"]
}
```

- task2\_vim\_save\_main\_tf.png



- task2\_terraform\_init\_output.png

```
@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.25.0...
- Installed hashicorp/aws v6.25.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

- task2\_terraform\_lock\_hcl.png

```
@Zuha-Irfan → /workspaces/LAB10 (main) $ cat .terraform.lock.hcl
# This file is maintained automatically by "terraform init".
# Manual edits may be lost in future updates.

provider "registry.terraform.io/hashicorp/aws" {
  version = "6.25.0"
  hashes = [
    "h1:0XEc9eHELD/BtPnybqkzzaS3bYp2HSv9LwAfaGyCp0U=",
    "zh:0f9621f719ec2051eabb94ca59aa4f13574487fbc1517b183293431c9d388e38",
    "zh:2ffbedb2e3afcd82da8bfc540bd74e9611527bdafe00d6d1885f62e7d13bac74",
    "zh:30fb4ab8b4af19da7b9ce95cb41fa9399f81383e1adc91801b770e7eeab651c3",
    "zh:377cbaffe3ec8aa5bb594071df0e91f17ac9292a325ed73cebd69fe78c51f7ec",
    "zh:3b65f5c98e03f1bfc5b71fa69521e785552ff9656860b25e211287910874037d",
    "zh:4478fab7b111c40a9a2a9db6ec5331618cc8e5a8b591f651095c77b87e9f22b1",
    "zh:4fdaa559c57aed5d24fa3d5cb59fed59e1e689c21d038fd336a3ba93b258803f",
    "zh:7a751ecd0f2654746dd4041d0f6d894c3a1876a152ba4bb7805ec2c715259065",
    "zh:866725b83f8d5587dab0559ac208ee6c181746871faa99ce551b535e19c7bb6a",
    "zh:9b12af85486a96aedd8d7984b0ff811a4b42e3d88dad1a3fb4c0b580d04fa425",
    "zh:b16e3e2a8ccba4ceeeee961c708ef572c4a65e001eaf09d08fa14cef01ab179",
    "zh:dc897b2037bbb7f8d6456a4aa1ed82cbd4daddb173a184efdfec8c03a57557771",
    "zh:de2344f23c980093a46dda3185f9052cda950d1b8ca9cf3c6e16b8c45fa23779",
    "zh:ef538ec8a917715a1804c6735d44b756c32972d4fab71e15df87a59eb75dd57c",
    "zh:f25cdfdac6798e7de4a1d3dd577a97c1ca200a12317a1fd5a4b9ea54cb05e868",
  ]
}
```

- task2\_terraform\_dir\_ls.png

```
@Zuha-Irfan → /workspaces/LAB10 (main) $ ls .terraform/
providers
```

## Task 3 — VPC/Subnet Creation, Initialization, Verification

- task3\_main\_tf\_resource\_add.png

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

```

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-08cbeded203de2207]

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-08cbeded203de2207"
}

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-08cbeded203de2207",
      "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-00cbded203de2207"
{
  "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-00cbded203de2207",
      "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-00cbded203de2207]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-03ed6c2085c712d1a]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-8cb18658aba92d3c]

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_dedup                          = 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               = (known after apply)
  + owner_id                              = (known after apply)
  + private_dns_hostnames_type_on_launch  = "ap-south-1"
  + tags.all                             = (known after apply)
  + vpc_id                                = "vpc-03ed6c2085c712d1a"
}

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

```

@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform destroy -target=aws_subnet.dev_subnet_1_existing
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-00cbded203de2207]
data.aws_subnet.existing_subnet_1_existing: Refreshing state... [id=subnet-0cddb18650aba92d3c]

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

Terraform will perform the following actions:

# aws_subnet.dev_subnet_1_existing will be destroyed
resource "aws_subnet" "dev_subnet_1_existing" {
  arn                               = "aws:arn::ec2:ap-south-1:80196211623:subnet/subnet-006022ba76a99ba" → null
  assign_ipv6_address_on_creation   = false → null
  availability_zone                 = "ap-south-1a" → null
  availability_zone_id              = "ap-south-1a" → null
  cidr_block                       = "172.31.48.0/24" → null
  enable_dns                        = false → null
  enable_dns64                    = false → null
  enable_resource_name_dns_a_record_on_launch = false → null
  enable_resource_name_dns_aaaa_record_on_launch = false → null
  id                               = "subnet-006022ba76a99ba" → null
  ipv6_native                      = false → null
  map_customer_owned_ip_on_launch  = false → null
  map_public_ip_on_launch          = false → null
  owner_id                        = "80196211623" → null
  private_dns_hostnames_type_on_launch = "private" → null
  region                          = "ap-south-1" → null
  tags                            = {} → null
  tags_all                        = {} → null
  vpc_id                          = "vpc-00cbded203de2207" → null
}

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

Warning: Resource targeting is in effect

You are creating a plan with the -target option, which means that the result of this plan may not represent all of the changes requested by the current configuration.

The -target option is not for routine use, and is provided only for exceptional situations such as recovering from errors or mistakes, or when Terraform specifically suggests to use it as part of an error message.

Do you really want to destroy all resources?
  Terraform will destroy all resources managed by infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes
aws_subnet.dev_subnet_1_existing: Destroying... [id=subnet-006022ba76a99ba]
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s

Warning: Applied changes may be incomplete

The plan was created with the -target option in effect, so some changes requested in the configuration may have been ignored and the output values may not be fully updated. Run the following command to verify that no other changes are pending:
  terraform plan

Note that the -target option is not suitable for routine use, and is provided only for exceptional situations such as recovering from errors or mistakes, or when Terraform specifically suggests to use it as part of an error message.

Destroy Complete!
Resources: 1 destroyed.

```

- task4\_terraform\_refresh\_state.png

```

Destroy Complete!
Resources: 1 destroyed.
@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00cbded203de2207]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-03ed0c2085c712d1a]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cddb18650aba92d3c]

```

- task4\_terraform\_apply\_after\_refresh.png

```

@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00cbded203de2207]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-03ed0c2085c712d1a]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cddb18650aba92d3c]

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_dns                        = 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_hostnames_type_on_launch = (known after apply)
  region                          = "ap-south-1"
  tags                            = (known after apply)
  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-0adeef7a5e1e8735a]

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

```

- task4\_terraform\_destroy\_all.png

- task4\_terraform\_plan\_output.png

- task4\_terraform\_apply\_after\_destroy.png



- task4\_main\_tf\_tagging.png

```

@Zuha-Irfan → /workspaces/LAB10 (main) $ vim main.tf
@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-88c850e263979ad4c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-83ed0c2085c712d1a]
aws_subnet.dev_subnet_1.existing: Refreshing state... [id=subnet-04b2ebe4a6983d466]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-048f62280cdf346bd]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed

```

- task4\_terraform\_apply\_tagging.png

```

@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-88c850e263979ad4c]
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" {
  availability_zone = "us-east-1a"           = (known after apply)
  cidr_block       = "10.0.10.0/24"         = (known after apply)
  enable_dns64     = false                  = (known after apply)
  enable_dns_hostnames = false              = (known after apply)
  enable_resource_name_dns_a_record_on_launch = false          = (known after apply)
  id               = (known after apply)
  ipam_pool_id     = (known after apply)
  ipam_pool_block_association_id = (known after apply)
  ipam_pool_ip_on_launch = false            = (known after apply)
  ipv6_address_assignment_id = (known after apply)
  ipv6_enabled     = false                  = (known after apply)
  private_dns_hostnames_type_on_launch = "up-south-1" = (known after apply)
  region          = "us-east-1"            = (known after apply)
  tags            = (known after apply)
  vpc_id          = (known after apply)
}

# aws_subnet.dev_subnet_1.existing will be created
+ resource "aws_subnet" "dev_subnet_1.existing" {
  availability_zone = "us-east-1a"           = (known after apply)
  cidr_block       = "10.0.10.0/24"         = (known after apply)
  enable_dns64     = false                  = (known after apply)
  enable_dns_hostnames = false              = (known after apply)
  enable_resource_name_dns_a_record_on_launch = false          = (known after apply)
  id               = (known after apply)
  ipam_pool_id     = (known after apply)
  ipam_pool_block_association_id = (known after apply)
  ipam_pool_ip_on_launch = false            = (known after apply)
  ipv6_address_assignment_id = (known after apply)
  ipv6_enabled     = false                  = (known after apply)
  private_dns_hostnames_type_on_launch = "up-south-1" = (known after apply)
  region          = "us-east-1"            = (known after apply)
  tags            = (known after apply)
  vpc_id          = (known after apply)
}

# aws_vpc.development_vpc will be created
+ resource "aws_vpc" "development_vpc" {
  cidr_block = "10.0.0.0/16" = (known after apply)
  default_network_acl_id = (known after apply)
  default_route_table_id = (known after apply)
  default_security_group_id = (known after apply)
  enable_dns_hostnames = (known after apply)
  enable_dns64 = (known after apply)
  enable_resource_name_dns_a_record_on_launch = (known after apply)
  id = (known after apply)
  ipam_pool_id = (known after apply)
  ipam_pool_block_association_id = (known after apply)
  ipam_pool_ip_on_launch = (known after apply)
  ipv6_address_assignment_id = (known after apply)
  ipv6_enabled = (known after apply)
  private_dns_hostnames_type_on_launch = (known after apply)
  region = (known after apply)
  tags = (known after apply)
  vpc_id = (known after apply)
}

Plan: 3 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_vpc.development_vpc: Creating...
aws_subnet.dev_subnet_1.existing: Creating...
aws_vpc.development_vpc: Creation complete after 0s [id=vpc-88c850e263979ad4c]
aws_subnet.dev_subnet_1: Creation complete after 0s [id=subnet-04b2ebe4a6983d466]
aws_subnet.dev_subnet_1: Creation complete after 0s [id=subnet-048f62280cdf346bd]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed

```

- task4\_terraform\_plan\_remove\_tag.png

```

@Zuha-Irfan → /workspaces/LAB10 (main) $ vim main.tf
@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform plan
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-88c850e263979ad4c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-83ed0c2085c712d1a]
aws_subnet.dev_subnet_1.existing: Refreshing state... [id=subnet-04b2ebe4a6983d466]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-048f62280cdf346bd]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

```

- task4\_terraform\_apply\_remove\_tag.png

```

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.
@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform apply -auto-approve
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-88c850e263979ad4c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-83ed0c2085c712d1a]
aws_subnet.dev_subnet_1.existing: Refreshing state... [id=subnet-04b2ebe4a6983d466]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-048f62280cdf346bd]

No changes. Your infrastructure matches the configuration.

Terraform has compared your real infrastructure against your configuration and found no differences, so no changes are needed.

Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
@Zuha-Irfan → /workspaces/LAB10 (main) $

```

## Task 5 — State File Inspection & Terraform State Commands

- task5\_terraform\_destroy.png



- task5\_terraform\_state\_file\_populated.png

- task5\_terraform\_state\_backup\_empty.png

- task5\_terraform\_state\_list.png

```
aws_vpc:development_vpc
```

- task5\_terraform\_state\_show\_resource.png

```

$ ssh -i tf-aws -o /workspaces/LAB10 (main) $ terraform state show data.aws_vpc.existing_vpc
# data.aws_vpc.existing_vpc
data "aws_vpc" "existing_vpc" {
  arn          = "arn:aws:ec2:ap-south-1:851942114323:vpc/vpc-03ed8c2085c712d1a"
  cidr_block   = "172.31.0.0/16"
  cidr_block_associations
  {
    association_id = "vpc-cidr-assoc-8778d6795df7edd89"
    cidr_block     = "172.31.0.0/16"
    state         = "associated"
  }
}
default
dhcp_options_id = "dopt-0016e1434ba594b91"
enable_dns_hostnames = true
enable_dns_support = true
enable_network_address_usage_metrics = false
id = "vpc-03ed8c2085c712d1a"
instance_tenancy = "default"
ipv6_association_id = null
ipv6_cidr_block = null
main_route_table_id = "rtb-83da73b0514a31afb"
owner_id = "851942114323"
region = "ap-south-1"
tags = {}
}

$ ssh -i tf-aws -o /workspaces/LAB10 (main) $ terraform state show aws_vpc.development_vpc
# aws_vpc.development_vpc
resource "aws_vpc" "development_vpc" {
  arn          = "arn:aws:ec2:ap-south-1:851942114323:vpc/vpc-0241d6d4d3ceddd35"
  assign_generated_ipv6_cidr_block = false
  cidr_block   = "10.0.0.0/16"
  default_network_acl_id = "acl-031ddeba8eaf125de"
  default_route_table_id = "rtb-02b51fd01330bec29"
  default_security_group_id = "sg-08dbd09f2c92f4aae"
  dhcp_options_id = "dopt-0016e1434ba594b91"
  enable_dns_hostnames = false
  enable_dns_support = true
  enable_network_address_usage_metrics = false
  id = "vpc-0241d6d4d3ceddd35"
  instance_tenancy = "default"
  ipv6_association_id = null
  ipv6_cidr_block = null
  ipv6_cidr_block_network_border_group = null
  ipv6_isolated = null
  ipv6_pool_id = null
  ipv6_netmask_length = 0
  main_route_table_id = "rtb-02b51fd01330bec29"
  owner_id = "851942114323"
  region = "ap-south-1"
  tags_all = {}
}

$ ssh -i tf-aws -o /workspaces/LAB10 (main) $ terraform state show aws_subnet.dev_subnet.1
# aws_subnet.dev_subnet.1
resource "aws_subnet" "dev_subnet.1" {
  arn          = "arn:aws:ec2:ap-south-1:851942114323:subnet/subnet-0080c134fd04fc13"
  assign_ipv6_address_on_creation = false
  availability_zone = "ap-south-1a"
  availability_zone_id = "ap-south-1a"
  cidr_block   = "10.0.10.0/24"
  cidr_block_associations
  {
    association_id = "vpc-cidr-assoc-8778d6795df7edd89"
    cidr_block     = "10.0.10.0/24"
    state         = "associated"
  }
  customer_owned_ipv4_pool = null
  enable_dns64 = false
  enable_inflight_device_index = false
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaaa_record_on_launch = false
  id = "subnet-0080c134fd04fc13"
  ipv6_cidr_block = null
  ipv6_cidr_block_association_id = null
  ipv6_native = false
  map_customer_owned_ip_on_launch = false
  map_public_ip_on_launch = false
  outpost_arn = null
  owner_id = "851942114323"
  private_dns_hostname_type_on_launch = "ip-name"
  region = "ap-south-1"
  tags_all = {}
  vpc_id = "vpc-0241d6d4d3ceddd35"
}

$ ssh -i tf-aws -o /workspaces/LAB10 (main) $ terraform state show aws_subnet.dev_subnet.1_existing
# aws_subnet.dev_subnet.1_existing
resource "aws_subnet" "dev_subnet.1_existing" {
  arn          = "arn:aws:ec2:ap-south-1:851942114323:subnet/subnet-00f6bd24dc3b75ee"
  assign_ipv6_address_on_creation = false
  availability_zone = "ap-south-1a"
  availability_zone_id = "ap-south-1a"
  cidr_block   = "172.31.0.0/24"
  cidr_block_associations
  {
    association_id = "vpc-cidr-assoc-8778d6795df7edd89"
    cidr_block     = "172.31.0.0/24"
    state         = "associated"
  }
  customer_owned_ipv4_pool = null
  enable_dns64 = false
  enable_inflight_device_index = false
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaaa_record_on_launch = false
  id = "subnet-00f6bd24dc3b75ee"
  ipv6_cidr_block = null
  ipv6_cidr_block_association_id = null
  ipv6_native = false
  map_customer_owned_ip_on_launch = false
  map_public_ip_on_launch = false
  outpost_arn = null
  owner_id = "851942114323"
  private_dns_hostname_type_on_launch = "ip-name"
  region = "ap-south-1"
  tags_all = {}
  vpc_id = "vpc-03ed8c2085c712d1a"
}

```

## Task 6 — Terraform Outputs & Attributes Reporting

- task6\_terraform\_outputs\_basic.png

```

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

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

# Subnet in 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: fetch default VPC
data "aws_vpc" "existing_vpc" {
  default = true
}

# Subnet in existing default VPC
resource "aws_subnet" "dev_subnet.1_existing" {
  vpc_id            = data.aws_vpc.existing_vpc.id
  cidr_block        = "172.31.0.0/24"
  availability_zone = "ap-south-1a"
}

output "dev-vpc-id" {
  value = aws_vpc.development_vpc.id
}

output "dev-subnet-id" {
  value = aws_subnet.dev_subnet.1.id
}

output "dev-vpc-arn" {
  value = aws_vpc.development_vpc.arn
}

output "dev-subnet-arn" {
  value = aws_subnet.dev_subnet.1.arn
}

```

- task6\_expanded\_outputs.png

[illegible]

## Cleanup — Delete Resources & State Verification

- [cleanup\\_destroy\\_resources.png](#)

```
terraform destroy

data.aws_vpc.existing_vpc: Reading
data.aws_development_vpc: Refreshing state... [id=vpc-8241d64ad3cebddd35]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-8241d64ad3cebddd35]
aws_subnet.dev_subnet-1: Refreshing state... [id=subnet-0b05bc134fd4f4c13]
aws_subnet.dev_subnet-1: Refreshing state... [id=subnet-0b05bc134fd4f4c13]

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

Terraform will perform the following actions:

# aws_subnet.dev_subnet-1 will be destroyed
resource "aws_subnet" "dev_subnet-1" {
  arn              = "arn:aws:ec2:ap-south-1:051902114323:subnet/subnet-0b05bc134fd4f4c13" -> null
  assign_ipv6_address_on_creation = false -> null
  availability_zone = "ap-south-1a" -> null
  cidr_block        = "10.0.10.0/24" -> null
  enable_dns        = false -> null
  enable_dns_hostnames = false -> null
  enable_nat_gateway = false -> null
  enable_resource_name_drift_recording_on_launch = false -> null
  id               = "subnet-0b05bc134fd4f4c13" -> null
  ipod_nat_gateway = false -> null
  map_customer_managed_ip_on_launch = false -> null
  map_public_ip_on_launch = false -> null
  owner_id         = "051902114323" -> null
  private_dns_hostnames_type_on_launch = "ip-name" -> null
  region          = "ap-south-1" -> null
  tags            = {
    "Name" = "dev-subnet-1"
  } -> null
  tags_all       = {
    "Name" = "dev-subnet-1"
  } -> null
  vpc_id         = "vpc-8241d64ad3cebddd35" -> null
}

# aws_subnet.dev_subnet-1.existing will be destroyed
resource "aws_subnet" "dev_subnet-1.existing" {
  arn              = "arn:aws:ec2:ap-south-1:051902114323:subnet/subnet-0b05bc134fd4f4c13" -> null
  assign_ipv6_address_on_creation = false -> null
  availability_zone = "ap-south-1a" -> null
  cidr_block        = "10.0.10.0/24" -> null
  enable_dns        = false -> null
  enable_dns_hostnames = false -> null
  enable_nat_gateway = false -> null
  enable_resource_name_drift_recording_on_launch = false -> null
  id               = "subnet-0b05bc134fd4f4c13" -> null
  ipod_nat_gateway = false -> null
  map_customer_managed_ip_on_launch = false -> null
  map_public_ip_on_launch = false -> null
  owner_id         = "051902114323" -> null
  private_dns_hostnames_type_on_launch = "ip-name" -> null
  region          = "ap-south-1" -> null
  tags            = {
    "Name" = "dev-subnet-existing"
  } -> null
  tags_all       = {
    "Name" = "dev-subnet-existing"
  } -> null
  vpc_id         = "vpc-8241d64ad3cebddd35" -> null
}

# aws_vpc.development_vpc will be destroyed
resource "aws_vpc" "development_vpc" {
  arn              = "arn:aws:ec2:ap-south-1:051902114323:vpc/vpc-8241d64ad3cebddd35" -> null
  assign_generated_ipv6_cidr_block = false -> null
  cidr_block       = "10.0.0.0/16" -> null
  default_network_acl_id = "acl-b01234567890123456" -> null
  default_route_table_id = "rtb-b01234567890123456" -> null
  default_subnet_id     = "subnet-b01234567890123456" -> null
  dhcp_options_id       = "dopt-b01234567890123456" -> null
  enable_dns_support    = true -> null
  enable_dns_hostnames = false -> null
  enable_nat_gateway    = false -> null
  instance_tenancy      = "default" -> null
  ipv6_address_pool_id   = "ip6-pool-b01234567890123456" -> null
  ipv6_prefix_length     = "16" -> null
  main_route_table_id    = "rtb-b01234567890123456" -> null
  owner_id              = "051902114323" -> null
  region               = "ap-south-1" -> null
  tags                 = {
    "Name" = "dev-vpc"
  } -> null
  tags_all             = {
    "Name" = "development-vpc"
  } -> null
}
```

- [cleanup\\_state\\_files.png](#)

```
$Zuha-1rfaan → /workspaces/LAB10 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 40,
  "lineage": "d13c841c-8da7-ac92-c880-2054a40271a2",
  "outputs": {},
  "resources": [
    {
      "check_results": null
    }
  ]
}

$Zuha-1rfaan → /workspaces/LAB10 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 38,
  "lineage": "d13c841c-8da7-ac92-c880-2054a40271a2",
  "outputs": {
    "dev-subnet-arn": {
      "value": "arn:aws:ec2:ap-south-1:051902114323:subnet/subnet-0b05bc134fd4f4c13",
      "type": "string"
    },
    "dev-subnet-cidr_block": {
      "value": "10.0.10.0/24",
      "type": "string"
    },
    "dev-subnet-id": {
      "value": "subnet-0b05bc134fd4f4c13",
      "type": "string"
    },
    "dev-subnet-region": {
      "value": "ap-south-1a",
      "type": "string"
    },
    "dev-subnet-tags_all": {
      "value": {
        "Name": "dev-subnet-1"
      },
      "type": [
        "map"
      ],
      "string": "string"
    },
    "dev-subnet-tags_name": {
      "value": "dev-subnet-1",
      "type": "string"
    },
    "dev-vpc-arn": {
      "value": "arn:aws:ec2:ap-south-1:051902114323:vpc/vpc-8241d64ad3cebddd35",
      "type": "string"
    },
    "dev-vpc-cidr_block": {
      "value": "10.0.0.0/16",
      "type": "string"
    },
    "dev-vpc-id": {
      "value": "vpc-8241d64ad3cebddd35",
      "type": "string"
    },
    "dev-vpc-region": {
      "value": "ap-south-1",
      "type": "string"
    },
    "dev-vpc-tags_all": {
      "value": {
        "Env": "dev",
        "Name": "development-vpc"
      },
      "type": [
        "map"
      ],
      "string": "string"
    },
    "dev-vpc-tags_name": {
      "value": "development-vpc",
      "type": "string"
    }
  ],
}
```

