

Name : Zuha Irfan

Roll No : 2023-BSE-073

Subject: Cloud Computing

Semester & Section : V-B

Lab :10

Task 1 — GitHub CLI Codespace Setup & Authentication

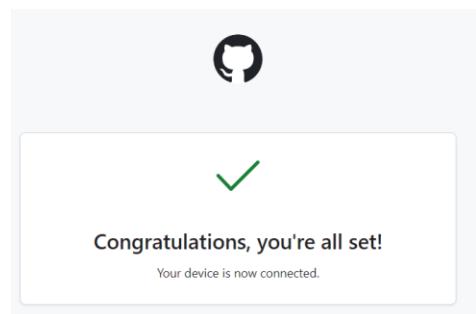
- task1_gh_install.png

```
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... Zuh... main* Shutdown about 13 ...
ubiquitous-... ubiquitous... Zuh... main* Shutdown about 6 h...
verbose-inv... verbose i... Zuh... main Available about 14 ...
glorious-ha... glorious... Zuh... 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 6.8.0-1830-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

```
[root@Irfan ~]# curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received =Xferd  Average Speed   Time   Time  Current
          Dload  Upload Total   Spent   Left  Speed
100 60.0M  100 60.0M    0     0  176M  0 --::-- --::-- 171M
```

```
[2023-06-12T10:44:11.144Z] + 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/_wasmcrt.abi3.so
  inflating: aws/dist/_ruamel_yaml.cpython-313-x86_64-linux-gnu.so
  inflating: aws/dist/liblzma.so.1
  inflating: aws/dist/libbz2.so.1
  inflating: aws/dist/libffi.so.6
  inflating: aws/dist/libuuid.so.1
  inflating: aws/dist/libutinfo.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
```

```
Installing: 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

```
@Zuhairf -> /workspaces/LAB10 (main) $ aws configure  
AWS Access Key ID [None]: AKIAQYF75YQJWLUAQYQW0  
AWS Secret Access Key [None]: Da7Vm9xlBa+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 = 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

```
output = json  
@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

```
49 packages can be upgraded. Run 'apt list --upgradable' to see them.
@Zuhairfan → /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) ...
@Zuhairfan → /workspaces/LAB10 (main) $ which terraform
/usr/bin/terraform
@Zuhairfan → /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

```
~  
~  
:wq|
```

- 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/BtPNybqkzsaS3bYp2HSv9LwAfaGyCpOU=",
    "zh:0f9621f719ec2051eabb94ca59aa4f13574487fbc1517b183293431c9d388e38",
    "zh:2ffbedb2e3afcd82da8bfc540bd74e9611527bdfad00d6d1885f62e7d13bac74",
    "zh:30fb4ab8b4af19da7b9ce95cb41fa9399f81383e1adc91801b770e7eeab651c3",
    "zh:377cbaf3e3ec8aa5bb594071df0e91f17ac9292a325ed73cebd69fe78c51f7ec",
    "zh:3b65f5c98e03flbfc5b71fa69521e785552ff9656860b25e2112879108740837d",
    "zh:4478fab7b111c40a9a2a9db6ec5331618cc8e5a8b591f651095c77b87e9f22b1",
    "zh:4fdaa559c57aed5d2fa3d5cb59fed59e1e689c21d038fd336a3ba93b258803f",
    "zh:7a751ecd0f2654746dd4041d0f6d894c3a1876a152ba4bb7805ec2c715259065",
    "zh:866725b83f8d5587dab0559ac208ee6c181746871faa99ce551b533e19c7bb6a",
    "zh:9b12af85486a96aedd8d7984b0ff811a4b42e3d88dad1a3fb4c0b580d04fa425",
    "zh:b16e3e2a8ccba4ceeeee961c708ef572c4a65e0001ea09d08fa14cef01ab179",
    "zh:dc897b2037bb7f8d6456a4aa1ed82cbd4daddb173a184efdfe8c03a57557771",
    "zh:de2344f23c980093a46dda3185f9052cda950d1b8c9cf3c6e16b8c45fa23779",
    "zh:ef538ec8a917715a1804c6735d44b756c32972d4fab71e15df87a59eb75dd57c",
    "zh:f25cdfdac6798e7de4a1dd577a97c1ca200a12317a1fd5a4b9ea54cb05e868",
  ]
}
```

- 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

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

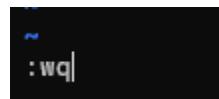
```

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"
}

```



- task3_terraform_apply_vpc_subnet.png

```

@Zuhu-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_aaa_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_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

```

@Zuhu-Irfan ~ /workspaces/LAB10 (main) $ aws ec2 describe-subnets --filter "Name=subnet-id,Values=subnet-0cdb18650aba92d3c"
{
    "Subnets": [
        {
            "AvailabilityZoneId": "aps1-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "851942114323",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:ap-south-1:851942114323: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



```
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
}
```

- task4_terraform_apply_datasource_resource.png

```
main.tf: 30L, 6556B written
@Zuha-Irfan → /workspaces/LAB10 (main) $ terraform apply
aws.aws_vpc.existing_vpc: Read complete
aws.aws_subnet.dev_subnet_1: Refreshing state... [id=vpc-00cbeded203de2207]
data.aws_vpc.existing_vpc: Read complete after 6s [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)
      + association_ipv6_address_on_creation = "172.31.48.0/24"
      + availability_zone_id               = "ap-south-1a"
      + cidr_block                         = "172.31.48.0/24"
      + ipv6_native                         = false
      + enable_resource_name_dns_a_record_on_launch = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                 = (known after apply)
      + map_public_ip_on_launch             = false
      + map_ipv6_native                   = false
      + owner_id                           = (known after apply)
      + resource_name_dns_hostname_type_on_launch = "ap-south-1"
      + 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-066d922ba78eb99aa]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

- task4_terraform_destroy_targeted.png

```
zhuo-OptiPlex-5090:~ zhuo$ terraform destroy -target=aws_subnet.dev_subnet_1_existing
data.aws_vpc.existing_vpc: Reading...
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=aws subnet-060d923b7eab99aa]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=aws subnet-060d923b7eab99aa]

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                                = "arn:aws:ec2:ap-south-1:601942114723:subnet/subnet-060d923b7eab99aa"
  assign_ipv6_address_on_creation      = false
  availability_zone                   = "ap-south-1"
  cidr_block                          = "172.31.48.0/24"
  enable_dhcp                         = true
  enable_ip_forwarding                = false
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaa_record_on_launch = false
  id                                  = "aws subnet-060d923b7eab99aa"
  ipv6_native                         = false
  map_public_ip_on_launch             = false
  max_latency                         = null
  private_dns_hostname_type_on_launch = "ap-south-1"
  region                             = "ap-south-1"
  tags                               = []
  tags_all                           = []
  vpc_id                             = "vpc-014edc2085c712d1a"
}

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 your AWS 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=aws subnet-060d923b7eab99aa]
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s

Warning: Applied changes may be incomplete
The 'dev_subnet' created with the -target option is still 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

```
@Zuha-Irfan ~ /workspaces/LAB10 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00cbeded203de2207]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-03ed0c2085c712d1a]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0cdb18650aba92d3c]
```

- task4_terraform_apply_after_refresh.png

```
zhuo-OptiPlex-5090:~ zhuo$ 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 1s [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-1"
  enable_dhcp                         = true
  enable_ip_forwarding                = false
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaa_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_hostname_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-0adeef7a5ele8735a]

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

- task4_terraform_destroy_all.png

```

@Zuhair-Irfan ~ /workspaces/LAB10 (main) $ terraform destroy
data.aws_vpc_existing_vpc: Refreshing state... [id=vpc-08c850e263979ad4c]
aws_vpc_development_vpc: Refreshing state... [id=vpc-03ed0c2885c712d1a]
aws_subnet_dev_subnet_1_existing: Refreshing state... [id=subnet-04b2eb4a6983d466]
aws_subnet_dev_subnet_1: Refreshing state... [id=subnet-048f62280cdf346bd]

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" {
  cidr_block           = "10.0.10.0/24"
  map_public_ip_on_launch = true
  max_private_ip_on_launch = 10
  name                 = "dev-subnet-1"
  private_ip_hostname_type_on_launch = "private"
  region               = "ap-south-1"
  tags_all             = {}
  vpc_id               = "vpc-03ed0c2885c712d1a"
}

# aws_vpc "development_vpc" will be destroyed
resource "aws_vpc" "development_vpc" {
  cidr_block           = "172.31.0.0/16"
  default_route_table_id = "rtt-03ed0c2885c712d1a"
  instance_tenancy     = "default"
  max_azs              = 2
  max_subnet_size      = 16
  name                 = "development-vpc"
  region               = "ap-south-1"
  subnet_allocation_id = "sa-03ed0c2885c712d1a"
  tags_all             = {}
  vpc_id               = "vpc-03ed0c2885c712d1a"
}

# data.aws_subnet "dev_subnet_1_existing" will be destroyed
resource "data.aws_subnet" "dev_subnet_1_existing" {
  filter {
    name   = "tag:Name"
    values = ["dev-subnet-1"]
  }
  id                = "subnet-04b2eb4a6983d466"
  map_public_ip_on_launch = true
  max_private_ip_on_launch = 10
  name              = "dev-subnet-1"
  private_ip_hostname_type_on_launch = "private"
  region            = "ap-south-1"
  tags_all          = {}
  vpc_id            = "vpc-03ed0c2885c712d1a"
}

Plan: 0 to add, 0 to change, 2 to destroy.

Do you really want to destroy all resources?
This will destroy the resources listed above. Terraform will attempt to prevent any unintentional destruction, as shown above.
There is no undo. Only "yes" will be accepted to confirm.

Enter a value for yes:

```

- task4_terraform_plan_output.png

```

@Zuhair-Irfan ~ /workspaces/LAB10 (main) $ terraform plan
data.aws_vpc_existing_vpc: Reading...
data.aws_vpc_existing_vpc: Read complete after 0s [id=vpc-03ed0c2885c712d1a]

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" {
  cidr_block           = "10.0.10.0/24"
  map_public_ip_on_launch = true
  max_private_ip_on_launch = 10
  name                 = "dev-subnet-1"
  private_ip_hostname_type_on_launch = "private"
  region               = "ap-south-1"
  tags_all             = {}
  vpc_id               = "vpc-03ed0c2885c712d1a"
}

# aws_subnet "dev_subnet_1_existing" will be created
resource "aws_subnet" "dev_subnet_1_existing" {
  cidr_block           = "10.0.10.0/24"
  map_public_ip_on_launch = true
  max_private_ip_on_launch = 10
  name                 = "dev-subnet-1"
  private_ip_hostname_type_on_launch = "private"
  region               = "ap-south-1"
  tags_all             = {}
  vpc_id               = "vpc-03ed0c2885c712d1a"
}

# aws_vpc "development_vpc" will be created
resource "aws_vpc" "development_vpc" {
  cidr_block           = "172.31.0.0/16"
  default_route_table_id = "rtt-03ed0c2885c712d1a"
  instance_tenancy     = "default"
  max_azs              = 2
  max_subnet_size      = 16
  name                 = "development-vpc"
  region               = "ap-south-1"
  subnet_allocation_id = "sa-03ed0c2885c712d1a"
  tags_all             = {}
  vpc_id               = "vpc-03ed0c2885c712d1a"
}

Plan: 3 to add, 0 to change, 0 to destroy.

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.

```

- task4_terraform_apply_after_destroy.png

```

@Zuhair-Irfan ~ /workspaces/LAB10 (main) $ terraform refresh
data.aws_vpc_existing_vpc: Reading...
aws_vpc_development_vpc: Refreshing state... [id=vpc-08c850e263979ad4c]
data.aws_vpc_existing_vpc: Read complete after 0s [id=vpc-03ed0c2885c712d1a]
aws_subnet_dev_subnet_1_existing: Refreshing state... [id=subnet-04b2eb4a6983d466]
aws_subnet_dev_subnet_1: Refreshing state... [id=subnet-048f62280cdf346bd]
@Zuhair-Irfan ~ /workspaces/LAB10 (main) $ terraform apply auto-approve
data.aws_vpc_existing_vpc: Reading...
aws_vpc_development_vpc: Refreshing state... [id=vpc-08c850e263979ad4c]
data.aws_vpc_existing_vpc: Read complete after 0s [id=vpc-03ed0c2885c712d1a]
aws_subnet_dev_subnet_1_existing: Refreshing state... [id=subnet-04b2eb4a6983d466]
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_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-08c850e263979ad4c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-03ed0c20885c712d1a]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-04b2eb4a6983d466]
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-03ed0c20885c712d1a]
Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:

Terraform will perform the following actions:

# aws_subnet "dev_subnet_1" will be created
resource "aws_subnet" "dev_subnet_1" {
  availability_zone           = "ap-south-1a"
  availability_zone_id        = "ap-south-1"
  cidr_block                  = "10.0.10.0/24"
  enable_dns_support          = false
  enable_dns_support_within_vpc = false
  ipv4_cidr_block_association_id = (known after apply)
  ipv4_native_ip_association_id = (known after apply)
  ipv6_native_ip_association_id = (known after apply)
  private_dns_hostname_type_on_launch = "none"
  region                       = "ap-south-1"
  subnet_id                    = (known after apply)
  vpc_id                       = (known after apply)
}

# aws_subnet "dev_subnet_1" existing will be created
resource "aws_subnet" "dev_subnet_1_existing" {
  arn                           = (known after apply)
  association_address_on_creation = "ap-south-1a"
  availability_zone              = "ap-south-1a"
  availability_zone_id            = "ap-south-1"
  cidr_block                     = "172.31.48.0/24"
  enable_dns_support             = false
  enable_dns_support_within_vpc = false
  enable_ip_transit              = false
  enable_ipv6                   = false
  enable_private_dns             = false
  enable_private_ip              = false
  max_latency                   = 100
  max_tags_all                  = (known after apply)
  private_dns_hostname_type_on_launch = "none"
  region                         = "ap-south-1"
  tags_all                       = (known after apply)
  vpc_id                          = "vpc-03ed0c20885c712d1a"
}

# aws_vpc "development_vpc" will be created
resource "aws_vpc" "development_vpc" {
  enable_dns_resolution_metrics = true
  enable_ip_transit              = false
  enable_ipv6                   = false
  instance_tenancy               = "default"
  ipv4_association_id            = (known after apply)
  ipv4_cidr_block                = "10.0.0.0/16"
  ipv4_cidr_block_association_id = (known after apply)
  ipv4_cidr_block_network_border_group = (known after apply)
  ipv4_native_ip_association_id = (known after apply)
  max_tags_all                   = (known after apply)
  region                         = "ap-south-1"
  tags_all                        = (known after apply)
}

Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Only 'yes' will be accepted to approve.
Enter a value: yes
aws_vpc.development_vpc: Creating...
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-04b2eb4a6983d466]
aws_vpc.development_vpc: Creation complete after 0s [id=vpc-03ed0c20885c712d1a]
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-08c850e263979ad4c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-03ed0c20885c712d1a]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-04b2eb4a6983d466]
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-08c850e263979ad4c]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-03ed0c20885c712d1a]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-04b2eb4a6983d466]
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.
```

Task 5 — State File Inspection & Terraform State Commands

- task5_terraform_destroy.png

```

$ terraform destroy
aws_vpc_development_vpcRefreshing state...
aws_vpc_development_vpc: Refreshing state... [id=arn:aws:vpc:us-east-1:08194211x123:subnet-048f6228bc0f3e6bd]
aws_subnet_dev_subnet_1_existing: Refreshing state... [id=arn:aws:vpc:us-east-1:08194211x123:subnet-048f6228bc0f3e6bd]

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

Terraform will perform the following actions:

# aws_subnet.dev_subnet_1 will be destroyed
resource "aws_subnet" "dev_subnet_1" {
  count = 1
  availability_zone_id = "ap-south-1"
  cidr_block = "10.10.0.0/24"
  enable_ip_forwarding = true
  enable_leak_device_index = true
  enable_resource_name_dns_a_record_on_launch = true
  enable_resource_name_dns_aaaa_record_on_launch = true
  id = "arn:aws:vpc:us-east-1:08194211x123:subnet-048f6228bc0f3e6bd"
  ipv4_cidr_block = "10.10.0.0/24"
  max_allowed_ip_addresses = 512
  name = "dev-subnet-1"
  private_ip_address_type_on_launch = "private"
  region = "ap-south-1"
  tags_all = []
}

# aws_subnet.dev_subnet_1_existing will be destroyed
resource "aws_subnet" "dev_subnet_1_existing" {
  count = 1
  availability_zone_id = "ap-south-1"
  cidr_block = "10.10.0.0/24"
  enable_ip_forwarding = true
  enable_leak_device_index = true
  enable_resource_name_dns_a_record_on_launch = true
  enable_resource_name_dns_aaaa_record_on_launch = true
  id = "arn:aws:vpc:us-east-1:08194211x123:subnet-048f6228bc0f3e6bd"
  ipv4_cidr_block = "10.10.0.0/24"
  max_allowed_ip_addresses = 512
  name = "dev-subnet-1"
  private_ip_address_type_on_launch = "private"
  region = "ap-south-1"
  tags_all = []
}

# aws_vpc_development_vpc will be destroyed
resource "aws_vpc" "development_vpc" {
  count = 1
  enable_dhcp_options = true
  enable_ip_forwarding = true
  enable_leak_device_index = true
  enable_resource_name_dns_a_record_on_launch = true
  enable_resource_name_dns_aaaa_record_on_launch = true
  enable_resource_name_dns_mx_record_on_launch = true
  enable_resource_name_dns_ns_record_on_launch = true
  enable_resource_name_dns_ptr_record_on_launch = true
  enable_resource_name_dns_svcs_record_on_launch = true
  enable_resource_name_dns_vrrp_record_on_launch = true
  enable_dns_hijacking_type_on_launch = "private"
  enable_dns_support = true
  enable_ip_transit = true
  enable_ip_transit_peering = true
  enable_ip_transit_vif = true
  instance_tenancy = "default"
  max_route_table_id = "rtt-007a15e7908e3217"
  name = "development-vpc"
  region = "ap-south-1"
  tags_all = []
}

Plan: 0 to add, 0 to change, 3 to destroy.

Do you really want to destroy all resources?
Terraform will destroy all your managed infrastructure, as shown above.
Do you want to continue? [y/N]: y

Enter a value: yes

aws_subnet.dev_subnet_1_existing: Destroying... [id=arn:aws:vpc:us-east-1:08194211x123:subnet-048f6228bc0f3e6bd]
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s
aws_vpc.development_vpc: Destroying... [id=arn:aws:vpc:us-east-1:08194211x123:vpc-88c956a267979d4d]
aws_vpc.development_vpc: Destruction complete after 1s
aws_subnet.dev_subnet_1_existing: Destroying after its parent is destroyed... [id=arn:aws:vpc:us-east-1:08194211x123:subnet-048f6228bc0f3e6bd]
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s

Destroy complete! Resources: 1 destroyed.

```

- task5_terraform_state_file_empty.png

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

- task5_terraform_state_backup_prev.png

- task5 terraform apply recreated.png

```

data.aws_vpc.dev_subnet_1 [data] $ terraform apply
data.aws_vpc.dev_subnet_1: Read complete after 0s [id=dev-subnet-0000000000000000]
No changes are planned for this resource.

Terraform will perform the following actions:

# aws_subnet dev_subnet_1 will be created
resource "aws_subnet" "dev_subnet_1" {
  vpc_id           = data.aws_vpc.dev_subnet_1.id
  availability_zone_id = data.aws_vpc.dev_subnet_1.availability_zone_id
  subnet_name      = "dev-subnet-1"
  map_public_ip_on_launch = true
  enable_dns_support = true
  enable_dns_hostnames = true
  private_ip_hostname_type_on_launch = "private"
  tags_all = {
    Name = "dev-subnet-1"
  }
}

# aws_subnet dev_subnet_1_existing will be replaced
resource "aws_subnet" "dev_subnet_1_existing" {
  vpc_id           = data.aws_vpc.dev_subnet_1.id
  availability_zone_id = data.aws_vpc.dev_subnet_1.availability_zone_id
  subnet_name      = "dev-subnet-1"
  map_public_ip_on_launch = true
  enable_dns_support = true
  enable_dns_hostnames = true
  private_ip_hostname_type_on_launch = "private"
  tags_all = {
    Name = "dev-subnet-1"
  }
}

# aws_vpc development_vpc will be created
resource "aws_vpc" "development_vpc" {
  cidr_block        = "172.16.0.0/16"
  enable_dns_support = true
  enable_dns_hostnames = true
  max_network_interfaces = 50
  subnet_allocation_id = "subnet-0000000000000000"
  subnet_association_id = "subnet-0000000000000000"
  subnet_map_public_ip_on_launch = true
  subnet_name = "development"
  subnet_size = 16
  tags_all = {
    Name = "development"
  }
}

Plan: 3 to add, 0 to change, 0 to destroy.

Terraform will perform the actions described above.

Review carefully before applying.

Enter a value: yes

aws_subnet dev_subnet_1 Existing: Creating...
aws_subnet dev_subnet_1_existing: Creation complete after 0s [id=dev-subnet-0000000000000000]
aws_subnet dev_subnet_1: Creating...
aws_subnet dev_subnet_1: Creation complete after 0s [id=dev-subnet-0000000000000000]

data.aws_vpc.dev_subnet_1 [data] $ terraform apply

```

- task5_terraform_state_file_populated.png

```

{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 26,
  "lineage": "d13c541c-0da7-ac92-c080-2054a40271a2",
  "outputs": {},
  "resources": [],
  "check_results": null
}

```

- 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

```
g7000-19-00 ~ /workspaces/LAB10 (main) $ terraform state show data.aws_vpc.existing_vpc
# data.aws_vpc "existing_vpc" {
  arn = "arn:aws:ec2:ap-south-1:051942114323:vpc/vpc-03ed0c2085c712dia"
  cidr_block = "172.31.0.0/16"
  cidr_block_associations {
    [
      association_id = "vpc-cidr-assoc-0776d6799df7edd89"
      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-03ed0c2085c712dia"
    instance_tenancy = "default"
    ipv4_cidr_block_id = "vpc-03ed0c2085c712dia"
    ipv6_cidr_block =
    main_route_table_id = "rtb-03daf3b514a31afab"
    owner_id = "051942114323"
    region = "ap-south-1"
    tags =
  ]
}

g7000-19-00 ~ /workspaces/LAB10 (main) $ terraform state show aws_vpc.development_vpc
# aws_vpc "development_vpc" {
  arn = "arn:aws:ec2:ap-south-1:051942114323:vpc/vpc-0201d64ad3cebdd35"
  assign_generated_ipv6_cidr_block = false
  cidr_block = "10.0.0.0/16"
  default_new_subnet_id = "acl-031031fd013bbcc29"
  default_security_group_id = "sg-01bbdbd2f2f92f1ae"
  dhcp_options_id = "dopt-0016e1434ba594b91"
  enable_dns_hostnames = false
  enable_dns_support = false
  enable_network_address_usage_metrics = false
  id = "vpc-0201d64ad3cebdd35"
  instance_tenancy = "default"
  ipv4_cidr_block =
  ipv6_association_id =
  ipv6_main_ipam_pool_id =
  ipv6_max_subnet_length =
  main_route_table_id = "rtb-02b81fd013bbcc29"
  owner_id = "051942114323"
  region = "ap-south-1"
  tags_all =
}

g7000-19-00 ~ /workspaces/LAB10 (main) $ terraform state show aws_subnet.dev_subnet_1
# aws_subnet "dev_subnet_1" {
  arn = "arn:aws:ec2:ap-south-1:051942114323:subnet/subnet-000bc13afde4c13"
  assign_ipv6_address_on_creation = false
  availability_zone = "ap-south-1a"
  availability_zone_id = "az-000bc13afde4c13"
  cidr_block = "10.0.0.0/24"
  customer_owned_ip_pool =
  enable_dns64 = false
  enable_ini_at_device_index = 0
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaaa_record_on_launch = false
  id = "subnet-000bc13afde4c13"
  ipv4_cidr_block =
  ipv6_cidr_block_association_id =
  ipv6_natural_gateway =
  map_customer_owned_ip_on_launch =
  map_public_ip_on_launch =
  outpost_arn =
  owner_id = "051942114323"
  private_dns_hostname_type_on_launch = "ip-name"
  region = "ap-south-1"
  tags_all =
  vpc_id = "vpc-0201d64ad3cebdd35"

g7000-19-00 ~ /workspaces/LAB10 (main) $ terraform state show aws_subnet.dev_subnet_1_existing
# aws_subnet "dev_subnet_1_existing" {
  arn = "arn:aws:ec2:ap-south-1:051942114323:subnet/subnet-00f6bed24fc3b75ee"
  assign_ipv6_address_on_creation = false
  availability_zone = "ap-south-1a"
  availability_zone_id = "az-00f6bed24fc3b75ee"
  cidr_block = "10.0.0.0/24"
  customer_owned_ip_pool =
  enable_dns64 = false
  enable_ini_at_device_index = 0
  enable_resource_name_dns_a_record_on_launch = false
  enable_resource_name_dns_aaaa_record_on_launch = false
  id = "subnet-00f6bed24fc3b75ee"
  ipv4_cidr_block =
  ipv6_cidr_block_association_id =
  ipv6_natural_gateway =
  map_customer_owned_ip_on_launch =
  map_public_ip_on_launch =
  outpost_arn =
  owner_id = "051942114323"
  private_dns_hostname_type_on_launch = "ip-name"
  region = "ap-south-1"
  tags_all =
  vpc_id = "vpc-03ed0c2085c712dia"
```

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 = "10.0.0.0/24"
  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 = "172.31.48.0/24"
  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
}
```

```

$ cd ./task6/outputs/main (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.existing_vpc: Refreshing state... [id=vpc-02401d64ad3cebd35]
data.aws_subnet.dev_subnet_1: Refreshing state... [id= subnet-00d5bc13ffdff4c13]
data.aws_subnet.dev_subnet_1: Refreshing state... [id= subnet-00d5bc13ffdff4c13]
data.aws_subnet.dev_subnet_1: Refreshing state... [id= subnet-00d5bc13ffdff4c13]
Changes to Outputs:
  + dev-subnet-arn = "arn:aws:ec2:ap-south-1:051942114323:subnet/subnet-00d5bc13ffdff4c13"
  + dev-vpc-arn = "arn:aws:ec2:ap-south-1:051942114323:vpc/vpc-02401d64ad3cebd35"
  + dev-vpc-id = "vpc-02401d64ad3cebd35"
You can apply this plan to save these new output values to the Terraform state, without changing any real infrastructure.
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

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

Outputs:
  + dev-subnet-arn = "arn:aws:ec2:ap-south-1:051942114323:subnet/subnet-00d5bc13ffdff4c13"
  + dev-vpc-arn = "arn:aws:ec2:ap-south-1:051942114323:vpc/vpc-02401d64ad3cebd35"
  + dev-vpc-id = "vpc-02401d64ad3cebd35"

```

- task6_expanded_outputs.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
  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     = "172.31.48.0/24"
  availability_zones = ["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
}

output "dev-vpc-cidr-block" {
  value = aws_vpc.development_vpc.cidr_block
}

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

output "dev-vpc-tags-name" {
  value = aws_vpc.development_vpc.tags["Name"]
}

output "dev-vpc-tags-all" {
  value = aws_vpc.development_vpc.tags_all
}

output "dev-subnet-region" {
  value = aws_subnet.dev_subnet_1.availability_zone
}

output "dev-subnet-tags-name" {
  value = aws_subnet.dev_subnet_1.tags["Name"]
}

output "dev-subnet-tags-all" {
  value = aws_subnet.dev_subnet_1.tags_all
}

...

```

```

data aws_vpc.existing_vpc { id = "vpc-02401d64ad3cebd35" }
data aws_subnet.dev_subnet_1 { id = "subnet-00d5bc13ffdff4c13" }
data aws_subnet.dev_subnet_1_existing { id = "subnet-00d5bc13ffdff4c13" }

Terraform used the selected provider to generate the following execution plan. Resource actions are indicated with the following symbols:
  + resource creation
  - resource update-in-place
  ~ resource update
  X resource destruction

Terraform will perform the following actions:
  # aws_subnet.dev_subnet_1 will be updated in-place
  ~ resource "aws_subnet" "dev_subnet_1_existing" {
      id          = "subnet-00d5bc13ffdff4c13"
      name        = "dev-subnet-1"
      - tags_all   =
      + tags_all   = [
          + "Name=dev-subnet-1"
        ]
      # (0 unchanged attributes hidden)
    }

  # aws_subnet.dev_subnet_1_existing will be updated in-place
  ~ resource "aws_subnet" "dev_subnet_1_existing" {
      id          = "subnet-00d5bc13ffdff4c13"
      name        = "dev-subnet-existing"
      - tags_all   =
      + tags_all   = [
          + "Name=dev-subnet-existing"
        ]
      # (0 unchanged attributes hidden)
    }

  # aws_vpc.development_vpc will be updated in-place
  ~ resource "aws_vpc" "development_vpc" {
      cidr_block = "10.0.0.0/16"
      - tags_all   =
      + tags_all   = [
          + "Name=development-vpc"
        ]
      - tags_name =
      + tags_name = "development-vpc"
      # (0 unchanged attributes hidden)
    }

Plan: 0 to add, 3 to change, 0 to destroy.

Changes to Outputs:
  + dev-subnet-arn = "arn:aws:ec2:ap-south-1:051942114323:subnet/subnet-00d5bc13ffdff4c13"
  + dev-vpc-arn = "arn:aws:ec2:ap-south-1:051942114323:vpc/vpc-02401d64ad3cebd35"
  + dev-vpc-id = "vpc-02401d64ad3cebd35"
  + dev-vpc-cidr-block = "10.0.0.0/16"
  + dev-vpc-region = "ap-south-1"
  + dev-vpc-tags-name = "development-vpc"
  + dev-vpc-tags-all = [
      + "Name=development-vpc"
    ]
  + dev-subnet-arn = "arn:aws:ec2:ap-south-1:051942114323:subnet/subnet-00d5bc13ffdff4c13"
  + dev-vpc-arn = "arn:aws:ec2:ap-south-1:051942114323:vpc/vpc-02401d64ad3cebd35"
  + dev-vpc-id = "vpc-02401d64ad3cebd35"
  + dev-vpc-cidr-block = "10.0.0.0/16"
  + dev-vpc-region = "ap-south-1"
  + dev-vpc-tags-name = "development-vpc"
  + dev-vpc-tags-all = [
      + "Name=development-vpc"
    ]

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

Outputs:
  + dev-subnet-arn = "arn:aws:ec2:ap-south-1:051942114323:subnet/subnet-00d5bc13ffdff4c13"
  + dev-vpc-arn = "arn:aws:ec2:ap-south-1:051942114323:vpc/vpc-02401d64ad3cebd35"
  + dev-vpc-id = "vpc-02401d64ad3cebd35"
  + dev-vpc-cidr-block = "10.0.0.0/16"
  + dev-vpc-region = "ap-south-1"
  + dev-vpc-tags-name = "development-vpc"
  + dev-vpc-tags-all = [
      + "Name=development-vpc"
    ]

```

Cleanup — Delete Resources & State Verification

- cleanup_destroy_resources.png

- cleanup_state_files.png

```
[zuhra-irfan ~]# cd /workspaces/LAB010 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 1,
  "lineage": "d13c54lc-0da7-ac92-c080-2054a40271a2",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@zuhra-irfan ~]# cd /workspaces/LAB010 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 35,
  "lineage": "d13c54lc-0da7-ac92-c080-2054a40271a2",
  "outputs": {},
  "resources": [
    {
      "dev-subnet-arm": {
        "value": "azurerm_subnet_ec2_ap-south-1:051942114323:subnet/subnet-0b85bc134fd4ff4c13",
        "type": "string"
      },
      "dev-subnet-cidr_block": {
        "value": "10.0.0.0/24",
        "type": "string"
      },
      "dev-subnet-id": {
        "value": "subnet-0b85bc134fd4ff4c13",
        "type": "string"
      },
      "dev-subnet-region": {
        "value": "ap-south-1a",
        "type": "string"
      },
      "dev-subnet-tags_all": {
        "value": {
          "Name": "dev-subnet-1"
        },
        "type": [
          "map",
          "string"
        ]
      },
      "dev-subnet-tags_name": {
        "value": "dev-subnet-1",
        "type": "string"
      },
      "dev-vpc-arm": {
        "value": "azurerm_vpc_ec2_ap-south-1:051942114323:vpc/vpc-0241dd64ad3cebd35",
        "type": "string"
      },
      "dev-vpc-id": {
        "value": "vpc-0241dd64ad3cebd35",
        "type": "string"
      },
      "dev-vpc-region": {
        "value": "ap-south-1a",
        "type": "string"
      },
      "dev-vpc-tags_all": {
        "value": [
          {
            "Name": "dev",
            "value": "development-vpc"
          }
        ],
        "type": [
          "map",
          "string"
        ]
      },
      "dev-vpc-tags_name": {
        "value": "development-vpc",
        "type": "string"
      }
    }
  ]
}
```

