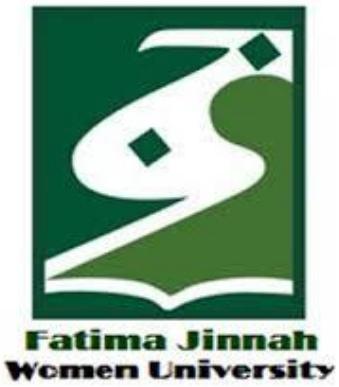


CLOUD COMPUTING LAB: 10

Submitted By:

Seerat Fatima

Registration. No:

2023-BSE-060

Submitted To:

Engr. Shoaib

Section:

5B

Task 1 — GitHub CLI Codespace Setup & Authentication

```
PS C:\Users\Waseem> 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.
```

```
PS C:\Users\Waseem> 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? Paste an authentication token
Tip: you can generate a Personal Access Token here https://github.com/settings/tokens
The minimum required scopes are 'repo', 'read:org', 'workflow'.
? Paste your authentication token: ************************************
- gh config set -h github.com git_protocol https
✓ Configured git protocol
✓ Logged in as SeratFatima00
! You were already logged in to this account
PS C:\Users\Waseem>
```

```
PS C:\Users\Waseem> gh codespace list
NAME                          DISPLAY NAME     REPOSITORY            BRANCH  STATE     CREATED AT
verbose-system-wrq7q47rjx7vcgq46  verbose system  SeratFatima00/Lab9  main*   Shutdown  about 1 day ago
```

```
PS C:\Users\Waseem> gh codespace ssh -c verbose-system-wrq7q47rjx7vcgq46
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
Last login: Sat Dec 27 14:52:50 2025 from ::1
```

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

```
  inflating: aws/dist/awscli/botocore/data/migrationhub-config/2019-06-30/paginators-1.json
  inflating: aws/dist/awscli/botocore/data/migrationhub-config/2019-06-30/endpoint-rule-set-1.json
  inflating: aws/dist/awscli/botocore/data/glue/2017-03-31/service-2.json
  inflating: aws/dist/awscli/botocore/data/glue/2017-03-31/paginators-1.json
  inflating: aws/dist/awscli/botocore/data/glue/2017-03-31/endpoint-rule-set-1.json
  inflating: aws/dist/awscli/botocore/data/glue/2017-03-31/paginators-1.sdk-extras.json
  inflating: aws/dist/awscli/botocore/data/glue/2017-03-31/completions-1.json
  inflating: aws/dist/awscli/botocore/data/license-manager/2018-08-01/paginators-1.json
  inflating: aws/dist/awscli/botocore/data/license-manager/2018-08-01/service-2.json
  inflating: aws/dist/awscli/botocore/data/license-manager/2018-08-01/endpoint-rule-set-1.json
  inflating: aws/dist/awscli/botocore/data/globalaccelerator/2018-08-08/endpoint-rule-set-1.json
  inflating: aws/dist/awscli/botocore/data/globalaccelerator/2018-08-08/paginators-1.json
  inflating: aws/dist/awscli/botocore/data/globalaccelerator/2018-08-08/service-2.json
  inflating: aws/dist/awscli/botocore/.changes/next-release/api-change-connect-59117.json
  inflating: aws/dist/awscli/customizations/wizard/wizards/configure/_main.yml
  inflating: aws/dist/awscli/customizations/wizard/wizards/dynamodb/new-table.yml
  inflating: aws/dist/awscli/customizations/wizard/wizards/iam/new-role.yml
  inflating: aws/dist/awscli/customizations/wizard/wizards/lambda/new-function.yml
  inflating: aws/dist/awscli/customizations/wizard/wizards/events/new-rule.yml
  inflating: aws/dist/awscli/customizations/sso/index.html
  inflating: aws/dist/awscli/data/metadata.json
  inflating: aws/dist/awscli/data/cli.json
  inflating: aws/dist/awscli/data/ac.index
  inflating: aws/dist/awscli/topics/config-vars.rst
  inflating: aws/dist/awscli/topics/topic-tags.json
  inflating: aws/dist/awscli/topics/s3-case-insensitivity.rst
  inflating: aws/dist/awscli/topics/return-codes.rst
  inflating: aws/dist/awscli/topics/s3-faq.rst
  inflating: aws/dist/awscli/topics/s3-config.rst
  inflating: aws/dist/awscli/topics/ddb-expressions.rst
  inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/METADATA
  inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/RECORD
  inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/INSTALLER
  inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/top_level.txt
  inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/WHEEL
  inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/AUTHORS.rst
  inflating: aws/dist/prompt_toolkit-3.0.51.dist-info/licenses/LICENSE
  inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
  inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
  inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
  inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
  inflating: aws/dist/wheel-0.45.1.dist-info/WHEEL
  inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
  inflating: aws/dist/wheel-0.45.1.dist-info/RECORD
  inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
@SeratFatima00 /workspaces/Lab9 (main) $ sudo ./aws/install
Found preexisting AWS CLI installation: /usr/local/aws-cli/v2/current. Please rerun install script with --update flag.
@SeratFatima00 /workspaces/Lab9 (main) $ aws --version
aws-cli/2.32.24 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws configure
AWS Access Key ID [****************ITV4]: A[redacted]
AWS Secret Access Key [****************zN+f]: [redacted]
Default region name [eu-north-1]: eu-north-1
Default output format [json]: json
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ cat ~/.aws/credentials
[default]
aws_access_key_id = [redacted]4
aws_secret_access_key = [redacted]f
@SeratFatima00 /workspaces/Lab9 (main) $ cat ~/.aws/config
[default]
region = eu-north-1
output = json
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws sts get-caller-identity
{
    "UserId": "A[redacted]",
    "Account": "[redacted]",
    "Arn": "arn:aws:iam::791666871664:user/Admin"
}
```

```
Get:6 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main amd64 Packages [77.4 kB]
Get:7 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble/main all Packages [643 B]
Get:8 https://dl.yarnpkg.com/debian stable/main amd64 Packages [11.8 kB]
Get:9 https://dl.yarnpkg.com/debian stable/main all Packages [11.8 kB]
Get:10 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:12 https://apt.releases.hashicorp.com noble/main amd64 Packages [266 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [1752 kB]
Get:14 http://archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:15 http://archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:16 http://archive.ubuntu.com/ubuntu noble/universe amd64 Packages [19.3 MB]
Get:17 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [33.1 kB]
Get:18 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [1183 kB]
Get:19 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [2898 kB]
Get:20 http://archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [331 kB]
Get:21 http://archive.ubuntu.com/ubuntu noble/main amd64 Packages [1808 kB]
Get:22 http://archive.ubuntu.com/ubuntu noble/restricted amd64 Packages [117 kB]
Get:23 http://archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [35.9 kB]
Get:24 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1950 kB]
Get:25 http://archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [3059 kB]
Get:26 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [2130 kB]
Get:27 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [49.5 kB]
Get:28 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [34.6 kB]
Fetched 35.7 MB in 5s (7590 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
@SeratFatima00 ➜ /workspaces/Lab9 (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 51 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 1.14.3-1 [30.6 MB]
Fetched 30.6 MB in 0s (157 MB/s)
Selecting previously unselected package terraform.
(Reading database ... 58629 files and directories currently installed.)
Preparing to unpack .../terraform_1.14.3-1_amd64.deb ...
Unpacking terraform (1.14.3-1) ...
Setting up terraform (1.14.3-1) ...
@SeratFatima00 ➜ /workspaces/Lab9 (main) $ which terraform
/usr/bin/terraform
@SeratFatima00 ➜ /workspaces/Lab9 (main) $ terraform --version
Terraform v1.14.3
on linux_amd64
```

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

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (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.
```

```
@SeratFatima00 /workspaces/Lab9 (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 = "6.27.0"
  hashes = [
    "h1:bixp2PSsP5ZGBczGCxcbSDn6lF5QFlUXlNroq9cdab4=",
    "zh:177a24b806c72e8484b5cabc93b2b38e3d770ae6f745a998b54d6619fd0e8129",
    "zh:4ac4a85c14fb868a3306b542e6a56c10bd6c6d5a67bc0c9b8f6a9060cf5f3be7",
    "zh:552652185bc85c8ba1da1d65dea47c454728a5c6839c458b6dcd3ce71c19ccfc",
    "zh:60284b8172d09aee91eae0856f09855eaf040ce3a58d6933602ae17c53f8ed04",
    "zh:6be38d156756ca61fb8e7c752cc5d769cd709686700ac4b230f40a6e95b5dbc9",
    "zh:7a409138fae4ef42e3a637e37cb9efedf96459e28a3c764fc4e855e8db9a7485",
    "zh:8070cf5224ed1ed3a3e9a59f7c30ff88bf071c7567165275d477c1738a56c064",
    "zh:894439ef340a9a79f69cd759e27ad11c7826adeca27be1b1ca82b3c9702fa300",
    "zh:89d035eebf08a97c89374ff06040955ddc09f275ecca609d0c9d58d149bef5cf",
    "zh:985b1145d724fc1f38369099e4a5087141885740fd6c0b1dbc492171e73c2e49",
    "zh:9b12af85486a96aedd8d7984b0ff811a4b42e3d88dad1a3fb4c0b580d04fa425",
    "zh:a80b47ae8d1475201c86bd94a5dcb9dd4da5e8b73102a90820b68b66b76d50fd",
    "zh:d3395be1556210f82199b9166a6b2e677cee9c4b67e96e63f6c3a98325ad7ab0",
    "zh:db0b869d09657f6f1e4110b56093c5fcdf9dbdd97c020db1e577b239c0adcbce",
    "zh:ffc72e680370ae7c21f9bd3082c6317730df805c6797427839a6b6b7e9a26a01",
  ]
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ ls .terraform/
providers
```

Task 3 — VPC/Subnet Creation, Initialization, Verification

```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
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 = "me-central-1a"
}
~
```

```
@SeratFatima00 /workspaces/Lab3 (main) $ terraform apply
aws_vpc.development_vpc: Refreshing state... [id=vpc-0c7eef7daaca4cdbf]

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                              = "eu-north-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                                         = "eu-north-1"
      + tags_all                                       = (known after apply)
      + vpc_id                                         = "vpc-0c7eef7daaca4cdbf"
    }

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 1s [id=subnet-0945d4eba56bc9fcf]

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

```
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-subnets --filter "Name=subnet-id,Values=subnet-0945d4eba56bc9fcf"
{
    "Subnets": [
        {
            "AvailabilityZoneId": "eun1-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "791666871664",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:eu-north-1:791666871664:subnet/subnet-0945d4eba56bc9fcf",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0945d4eba56bc9fcf",
            "State": "available",
            "VpcId": "vpc-0c7eef7daaca4cdbf",
            "CidrBlock": "10.0.10.0/24",
            "AvailableIpAddressCount": 251,
            "AvailabilityZone": "eu-north-1a",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": false
        }
    ]
}
@SeratFatima00 /workspaces/Lab9 (main) $ aws ec2 describe-vpcs --filter "Name=vpc-id,Values=vpc-0c7eef7daaca4cdbf"
{
    "Vpcs": [
        {
            "OwnerId": "791666871664",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0c683bcd95cc04100",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
```

Task 4 — Data Source, Targeted Destroy, Tags

```hcl
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
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 = "eu-north-1a"
}
data "aws_vpc" "existing_vpc" {
  default = true
}

resource "aws_subnet" "dev_subnet_1_existing" {
  vpc_id            = data.aws_vpc.existing_vpc.id
  cidr_block        = "172.31.48.0/24"
  availability_zone = "eu-north-1a"  # update to valid AZ
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0c7eef7daaca4cdbf]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0945d4eba56bc9fcf]

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                              = "eu-north-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "172.31.48.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "eu-north-1"
      + tags_all                                       = (known after apply)
      + vpc_id                                         = "vpc-0be30d1beb39c1848"
    }

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 2s [id=subnet-048d939a3ce9540d0]

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

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform destroy -target=aws_subnet.dev_subnet_1_existing
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-048d939a3ce9540d0]

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:eu-north-1:791666871664:subnet/subnet-048d939a3ce9540d0" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "eu-north-1a" -> null
      - availability_zone_id                           = "eun1-az1" -> null
      - cidr_block                                     = "172.31.48.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-048d939a3ce9540d0" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "791666871664" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - region                                         = "eu-north-1" -> null
      - tags                                           = {} -> null
      - tags_all                                       = {} -> null
      - vpc_id                                         = "vpc-0be30d1beb39c1848" -> null
      # (4 unchanged attributes hidden)
    }

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 managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0c7eef7daaca4cdbf]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0945d4eba56bc9fcf]
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0c7eef7daaca4cdbf]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0945d4eba56bc9fcf]

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                              = "eu-north-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "172.31.48.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "eu-north-1"
      + tags_all                                       = (known after apply)
      + vpc_id                                         = "vpc-0be30d1beb39c1848"
    }

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-004ad1eb6f7658228]

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

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform destroy
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0c7eef7daaca4cdbf]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-004ad1eb6f7658228]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0945d4eba56bc9fcf]

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:eu-north-1:791666871664:subnet/subnet-0945d4eba56bc9fcf" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "eu-north-1a" -> null
      - availability_zone_id                           = "eun1-az1" -> null
      - cidr_block                                     = "10.0.10.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-0945d4eba56bc9fcf" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "791666871664" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - region                                         = "eu-north-1" -> null
      - tags                                           = {} -> null
      - tags_all                                       = {} -> null
      - vpc_id                                         = "vpc-0c7eef7daaca4cdbf" -> null
        # (4 unchanged attributes hidden)
    }

  # aws_subnet.dev_subnet_1_existing will be destroyed
  - resource "aws_subnet" "dev_subnet_1_existing" {
      - arn                                            = "arn:aws:ec2:eu-north-1:791666871664:subnet/subnet-004ad1eb6f7658228" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "eu-north-1a" -> null
      - availability_zone_id                           = "eun1-az1" -> null
      - cidr_block                                     = "172.31.48.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-004ad1eb6f7658228" -> null
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform plan
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0be30d1beb39c1848]

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                              = "eu-north-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                                         = "eu-north-1"
      + tags_all                                       = (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)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "eu-north-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "172.31.48.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "eu-north-1"
      + tags_all                                       = (known after apply)
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0be30d1beb39c1848]

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                              = "eu-north-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                                         = "eu-north-1"
      + tags_all                                       = (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)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "eu-north-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "172.31.48.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
      + id                                             = (known after apply)
      + ipv6_cidr_block_association_id                 = (known after apply)
      + ipv6_native                                    = false
      + map_public_ip_on_launch                        = false
      + owner_id                                       = (known after apply)
      + private_dns_hostname_type_on_launch            = (known after apply)
      + region                                         = "eu-north-1"
      + tags_all                                       = (known after apply)
```

```
provider "aws" {
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"]
}
resource "aws_vpc" "development_vpc" {
  cidr_block = "10.0.0.0/16"
   tags = {
     Name: "development"
     vpc_env = "dev"
  }
}

resource "aws_subnet" "dev_subnet_1" {
  vpc_id            = aws_vpc.development_vpc.id
  cidr_block        = "10.0.10.0/24"
  availability_zone = "eu-north-1a"
  tags = {
     Name: "subnet-1-dev"
  }
}
data "aws_vpc" "existing_vpc" {
  default = true
}

resource "aws_subnet" "dev_subnet_1_existing" {
  vpc_id            = data.aws_vpc.existing_vpc.id
  cidr_block        = "172.31.48.0/24"
  availability_zone = "eu-north-1a"  # update to valid AZ
  tags = {
     Name: "subnet-1-default"
  }
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-070925d0f08fb8b7e]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0cd02a613fa352e95]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0ccd91eef17eae9e7]
@SeratFatima00 /workspaces/Lab9 (main) $ terraform apply -auto-approve
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-070925d0f08fb8b7e]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0cd02a613fa352e95]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0ccd91eef17eae9e7]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_subnet.dev_subnet_1 will be updated in-place
  ~ resource "aws_subnet" "dev_subnet_1" {
        id                                             = "subnet-0ccd91eef17eae9e7"
      ~ tags                                           = {
          + "Name" = "subnet-1-dev"
        }
      ~ tags_all                                       = {
          + "Name" = "subnet-1-dev"
        }
        # (20 unchanged attributes hidden)
    }

  # aws_subnet.dev_subnet_1_existing will be updated in-place
  ~ resource "aws_subnet" "dev_subnet_1_existing" {
        id                                             = "subnet-0cd02a613fa352e95"
      ~ tags                                           = {
          + "Name" = "subnet-1-default"
        }
      ~ tags_all                                       = {
          + "Name" = "subnet-1-default"
        }
        # (20 unchanged attributes hidden)
    }

  # aws_vpc.development_vpc will be updated in-place
  ~ resource "aws_vpc" "development_vpc" {
        id                                   = "vpc-070925d0f08fb8b7e"
      ~ tags                                 = {
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform plan
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-070925d0f08fb8b7e]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0cd02a613fa352e95]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0ccd91eef17eae9e7]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_vpc.development_vpc will be updated in-place
  ~ resource "aws_vpc" "development_vpc" {
        id                                   = "vpc-070925d0f08fb8b7e"
      ~ tags                                 = {
            "Name"    = "development"
          - "vpc_env" = "dev" -> null
        }
      ~ tags_all                             = {
          - "vpc_env" = "dev" -> null
            # (1 unchanged element hidden)
        }
        # (19 unchanged attributes hidden)
    }

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

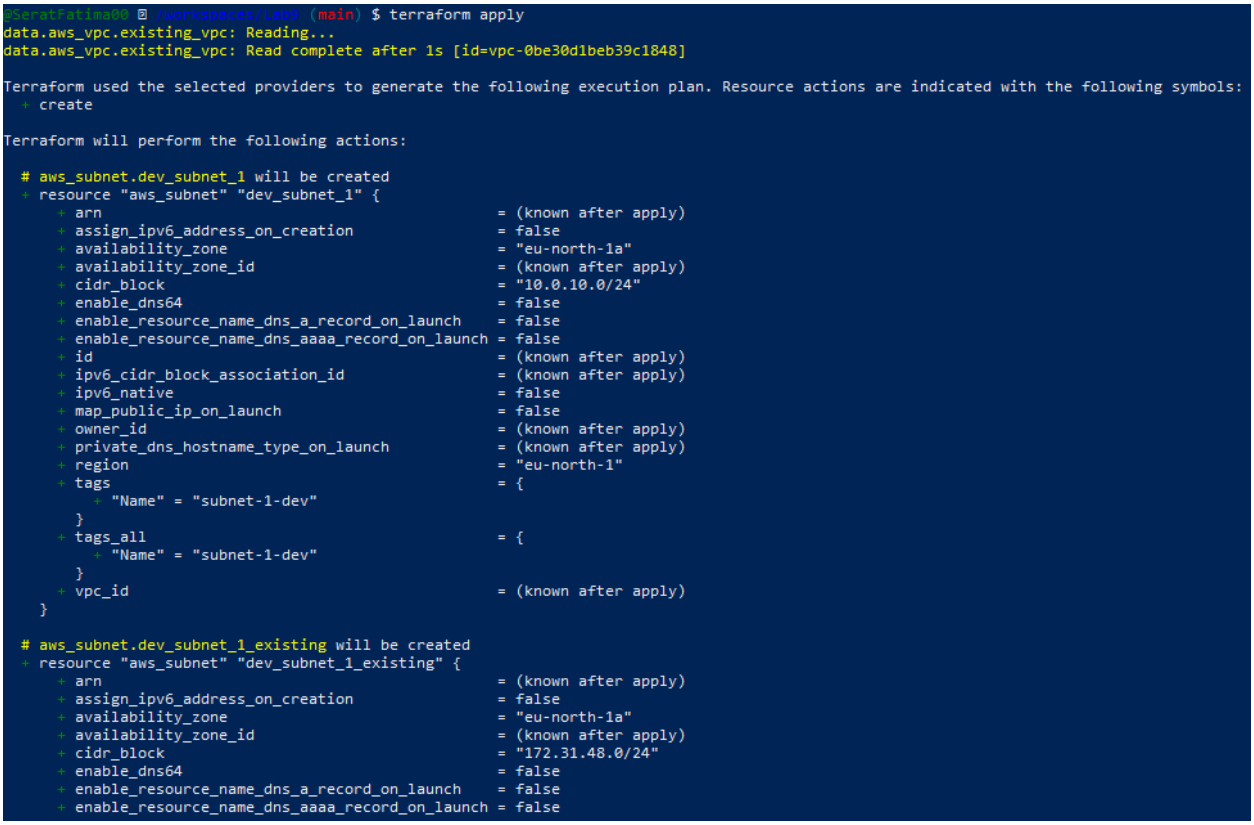
```
Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions if you run "terraform apply" now.
@SeratFatima00 /workspaces/Lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-070925d0f08fb8b7e]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0cd02a613fa352e95]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0ccd91eef17eae9e7]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
  ~ update in-place

Terraform will perform the following actions:

  # aws_vpc.development_vpc will be updated in-place
  ~ resource "aws_vpc" "development_vpc" {
        id                                   = "vpc-070925d0f08fb8b7e"
      ~ tags                                 = {
            "Name"    = "development"
          - "vpc_env" = "dev" -> null
        }
```

Task 5 — State File Inspection & Terraform State Commands

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform destroy
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-070925d0f08fb8b7e]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0cd02a613fa352e95]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0ccd91eef17eae9e7]

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:eu-north-1:791666871664:subnet/subnet-0ccd91eef17eae9e7" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "eu-north-1a" -> null
      - availability_zone_id                           = "eun1-az1" -> null
      - cidr_block                                     = "10.0.10.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-0ccd91eef17eae9e7" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "791666871664" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - region                                         = "eu-north-1" -> null
      - tags                                           = {
          - "Name" = "subnet-1-dev"
        } -> null
      - tags_all                                       = {
          - "Name" = "subnet-1-dev"
        } -> null
      - vpc_id                                         = "vpc-070925d0f08fb8b7e" -> null
        # (4 unchanged attributes hidden)
    }

  # aws_subnet.dev_subnet_1_existing will be destroyed
  - resource "aws_subnet" "dev_subnet_1_existing" {
      - arn                                            = "arn:aws:ec2:eu-north-1:791666871664:subnet/subnet-0cd02a613fa352e95" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "eu-north-1a" -> null
      - availability_zone_id                           = "eun1-az1" -> null
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 30,
  "lineage": "533d4fbc-2b1b-0d50-6a57-3c6107ba1da0",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 25,
  "lineage": "533d4fbc-2b1b-0d50-6a57-3c6107ba1da0",
  "outputs": {},
  "resources": [
    {
      "mode": "data",
      "type": "aws_vpc",
      "name": "existing_vpc",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "arn": "arn:aws:ec2:eu-north-1:791666871664:vpc/vpc-0be30d1beb39c1848",
            "cidr_block": "172.31.0.0/16",
            "cidr_block_associations": [
              {
                "association_id": "vpc-cidr-assoc-04f111177fd2b5bce",
                "cidr_block": "172.31.0.0/16",
                "state": "associated"
              }
            ],
            "default": true,
            "dhcp_options_id": "dopt-0b0d23efc3a45451c",
            "enable_dns_hostnames": true,
            "enable_dns_support": true,
            "enable_network_address_usage_metrics": false,
            "filter": null,
            "id": "vpc-0be30d1beb39c1848",
            "instance_tenancy": "default",
            "ipv6_association_id": "",
            "ipv6_cidr_block": "",
            "main_route_table_id": "rtb-0e647fa4ed429fbc3",
            "owner_id": "791666871664",
            "region": "eu-north-1",
            "state": null,
            "tags": {},
            "timeouts": null
          },
          "sensitive_attributes": [],
          "identity_schema_version": 0
        }
      ]
```



```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0be30d1beb39c1848]

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                              = "eu-north-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                                         = "eu-north-1"
      + tags                                           = {
          + "Name" = "subnet-1-dev"
        }
      + tags_all                                       = {
          + "Name" = "subnet-1-dev"
        }
      + 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)
      + assign_ipv6_address_on_creation                = false
      + availability_zone                              = "eu-north-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "172.31.48.0/24"
      + enable_dns64                                   = false
      + enable_resource_name_dns_a_record_on_launch    = false
      + enable_resource_name_dns_aaaa_record_on_launch = false
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 34,
  "lineage": "533d4fbc-2b1b-0d50-6a57-3c6107ba1da0",
  "outputs": {},
  "resources": [
    {
      "mode": "data",
      "type": "aws_vpc",
      "name": "existing_vpc",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "arn": "arn:aws:ec2:eu-north-1:791666871664:vpc/vpc-0be30d1beb39c1848",
            "cidr_block": "172.31.0.0/16",
            "cidr_block_associations": [
              {
                "association_id": "vpc-cidr-assoc-04f111177fd2b5bce",
                "cidr_block": "172.31.0.0/16",
                "state": "associated"
              }
            ],
            "default": true,
            "dhcp_options_id": "dopt-0b0d23efc3a45451c",
            "enable_dns_hostnames": true,
            "enable_dns_support": true,
            "enable_network_address_usage_metrics": false,
            "filter": null,
            "id": "vpc-0be30d1beb39c1848",
            "instance_tenancy": "default",
            "ipv6_association_id": "",
            "ipv6_cidr_block": "",
            "main_route_table_id": "rtb-0e647fa4ed429fbc3",
            "owner_id": "791666871664",
            "region": "eu-north-1",
            "state": null,
            "tags": {},
            "timeouts": null
          },
          "sensitive_attributes": [],
          "identity_schema_version": 0
        }
      ]
    },
    {
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 30,
  "lineage": "533d4fbc-2b1b-0d50-6a57-3c6107ba1da0",
  "outputs": {},
  "resources": [],
  "check_results": null
}
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform state list
data.aws_vpc.existing_vpc
aws_subnet.dev_subnet_1
aws_subnet.dev_subnet_1_existing
aws_vpc.development_vpc
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ terraform state show aws_vpc.development_vpc
# aws_vpc.development_vpc:
resource "aws_vpc" "development_vpc" {
    arn                                  = "arn:aws:ec2:eu-north-1:791666871664:vpc/vpc-0861e001cde7fa292"
    assign_generated_ipv6_cidr_block     = false
    cidr_block                           = "10.0.0.0/16"
    default_network_acl_id               = "acl-0594be9eed655f6fb"
    default_route_table_id               = "rtb-0ca5e04d90275de31"
    default_security_group_id            = "sg-02bc2e9f500401105"
    dhcp_options_id                      = "dopt-0b0d23efc3a45451c"
    enable_dns_hostnames                 = false
    enable_dns_support                   = true
    enable_network_address_usage_metrics = false
    id                                   = "vpc-0861e001cde7fa292"
    instance_tenancy                     = "default"
    ipv6_association_id                  = null
    ipv6_cidr_block                      = null
    ipv6_cidr_block_network_border_group = null
    ipv6_ipam_pool_id                    = null
    ipv6_netmask_length                  = 0
    main_route_table_id                  = "rtb-0ca5e04d90275de31"
    owner_id                             = "791666871664"
    region                               = "eu-north-1"
    tags                                 = {
        "Name" = "development"
    }
    tags_all                             = {
        "Name" = "development"
    }
}
```

Task 6 — Terraform Outputs & Attributes Reporting

```
@SeratFatima00 /workspaces/Lab9 (main) $ vim main.tf
@SeratFatima00 /workspaces/Lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0861e001cde7fa292]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0ab656a704dd0b75f]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-063fea4d7df3654f3]

Changes to Outputs:
  + dev-subnet-arn = "arn:aws:ec2:eu-north-1:791666871664:subnet/subnet-063fea4d7df3654f3"
  + dev-subnet-id  = "subnet-063fea4d7df3654f3"
  + dev-vpc-arn    = "arn:aws:ec2:eu-north-1:791666871664:vpc/vpc-0861e001cde7fa292"
  + dev-vpc-id     = "vpc-0861e001cde7fa292"

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:eu-north-1:791666871664:subnet/subnet-063fea4d7df3654f3"
dev-subnet-id = "subnet-063fea4d7df3654f3"
dev-vpc-arn = "arn:aws:ec2:eu-north-1:791666871664:vpc/vpc-0861e001cde7fa292"
dev-vpc-id = "vpc-0861e001cde7fa292"
```

```
Apply cancelled.
@SeratFatima00 /workspaces/Lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0861e001cde7fa292]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0ab656a704dd0b75f]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-063fea4d7df3654f3]

Changes to Outputs:
  + dev-subnet-cidr_block = "10.0.10.0/24"
  + dev-subnet-region     = "eu-north-1a"
  + dev-subnet-tags_all   = {
      + Name = "subnet-1-dev"
    }
  + dev-subnet-tags_name  = "subnet-1-dev"
  + dev-vpc-cidr_block    = "10.0.0.0/16"
  + dev-vpc-region        = "eu-north-1"
  + dev-vpc-tags_all      = {
      + Name = "development"
    }
  + dev-vpc-tags_name     = "development"

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:eu-north-1:791666871664:subnet/subnet-063fea4d7df3654f3"
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-id = "subnet-063fea4d7df3654f3"
dev-subnet-region = "eu-north-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-arn = "arn:aws:ec2:eu-north-1:791666871664:vpc/vpc-0861e001cde7fa292"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-id = "vpc-0861e001cde7fa292"
dev-vpc-region = "eu-north-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
```

Cleanup — Delete Resources & State Verification

```
@SeratFatima00 /workspaces/Lab8 (main) $ terraform destroy
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0861e001cde7fa292]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0be30d1beb39c1848]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0ab656a704dd0b75f]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-063fea4d7df3654f3]

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:eu-north-1:791666871664:subnet/subnet-063fea4d7df3654f3" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "eu-north-1a" -> null
      - availability_zone_id                           = "eun1-az1" -> null
      - cidr_block                                     = "10.0.10.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
      - enable_resource_name_dns_a_record_on_launch    = false -> null
      - enable_resource_name_dns_aaaa_record_on_launch = false -> null
      - id                                             = "subnet-063fea4d7df3654f3" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "791666871664" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - region                                         = "eu-north-1" -> null
      - tags                                           = {
          - "Name" = "subnet-1-dev"
        } -> null
      - tags_all                                       = {
          - "Name" = "subnet-1-dev"
        } -> null
      - vpc_id                                         = "vpc-0861e001cde7fa292" -> null
        # (4 unchanged attributes hidden)
    }

  # aws_subnet.dev_subnet_1_existing will be destroyed
  - resource "aws_subnet" "dev_subnet_1_existing" {
      - arn                                            = "arn:aws:ec2:eu-north-1:791666871664:subnet/subnet-0ab656a704dd0b75f" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "eu-north-1a" -> null
      - availability_zone_id                           = "eun1-az1" -> null
      - cidr_block                                     = "172.31.48.0/24" -> null
      - enable_dns64                                   = false -> null
```

```
@SeratFatima00 /workspaces/Lab9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 41,
  "lineage": "533d4fbc-2b1b-0d50-6a57-3c6107ba1da0",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@SeratFatima00 /workspaces/Lab9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 36,
  "lineage": "533d4fbc-2b1b-0d50-6a57-3c6107ba1da0",
  "outputs": {
    "dev-subnet-arn": {
      "value": "arn:aws:ec2:eu-north-1:791666871664:subnet/subnet-063fea4d7df3654f3",
      "type": "string"
    },
    "dev-subnet-cidr_block": {
      "value": "10.0.10.0/24",
      "type": "string"
    },
    "dev-subnet-id": {
      "value": "subnet-063fea4d7df3654f3",
      "type": "string"
    },
    "dev-subnet-region": {
      "value": "eu-north-1a",
      "type": "string"
    },
    "dev-subnet-tags_all": {
      "value": {
        "Name": "subnet-1-dev"
      },
      "type": [
        "map",
        "string"
      ]
    },
    "dev-subnet-tags_name": {
      "value": "subnet-1-dev",
      "type": "string"
    },
```