CLOUD COMPUTING LAB
BSE (V-B)

Submitted By:

Musfira Farooq

Roll No:

2023-BSE-045

Submitted To:

Sir Muhammad Shoaib

LAB 10

GH CLI Codespaces + AWS + Terraform: VPC & Subnet Automation

Task 1 — GitHub CLI, Codespace setup and authentication

Objective: Install GH CLI, authenticate for Codespaces, and connect to a Codespace.

- Install GitHub CLI

```
PS C:\Users\Musfi> winget install --id GitHub.cli
Found an existing package already installed. Trying
```

- GH CLI authentication for Codespaces

```
PS C:\Users\Musfi> 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 Musfira-0514
! You were already logged in to this account
PS C:\Users\Musfi>
```

- List available Codespaces

```
PS C:\Users\Musfi> gh codespace list
NAME                                DISPLAY NAME           REPOSITORY          BRANCH  STATE     CREATED AT
effective-space-orbit-97gj757p49vvcpr9v  effective space orbit  Musfira-0514/lab-9  main*   Shutdown  about 1 day ago
PS C:\Users\Musfi>
```

- Connect to Codespace via SSH

```
PS C:\Users\Musfi> gh codespace ssh -c effective-space-orbit-97gj757p49vvcpr9v
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: Tue Dec 30 22:18:14 2025 from ::1
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

Task 2 — Install AWS CLI & Terraform, Provider Setup

Objective: Install AWS CLI & Terraform CLI and configure the Terraform provider.

- Terraform installation and version check

```
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
Fetched 30.6 MB in 0s (154 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) ...
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ which terraform
terraform --version
/usr/bin/terraform
Terraform v1.14.3
on linux_amd64
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Provider file creation (main.tf)

```
@Musfira-0514 → /workspaces/lab-9 (main) $ vim main.tf

@Musfira-0514 → /workspaces/lab-9 (main) $
```

- Provider block configuration

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

- Terraform initialization

```
@Musfira-0514 → /workspaces/lab-9 (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!
```

- Terraform lock file

```
@Musfira-0514 → /workspaces/lab-9 (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",
  ]
}
@Musfira-0514 → /workspaces/lab-9 (main) $
```

- Terraform directory structure

```
@Musfira-0514 → /workspaces/lab-9 (main) $ ls .terraform/
providers
@Musfira-0514 → /workspaces/lab-9 (main) $
```

Task 3 — VPC & Subnet Creation Using Terraform

Objective: Create VPC and subnet resources and verify using AWS CLI.

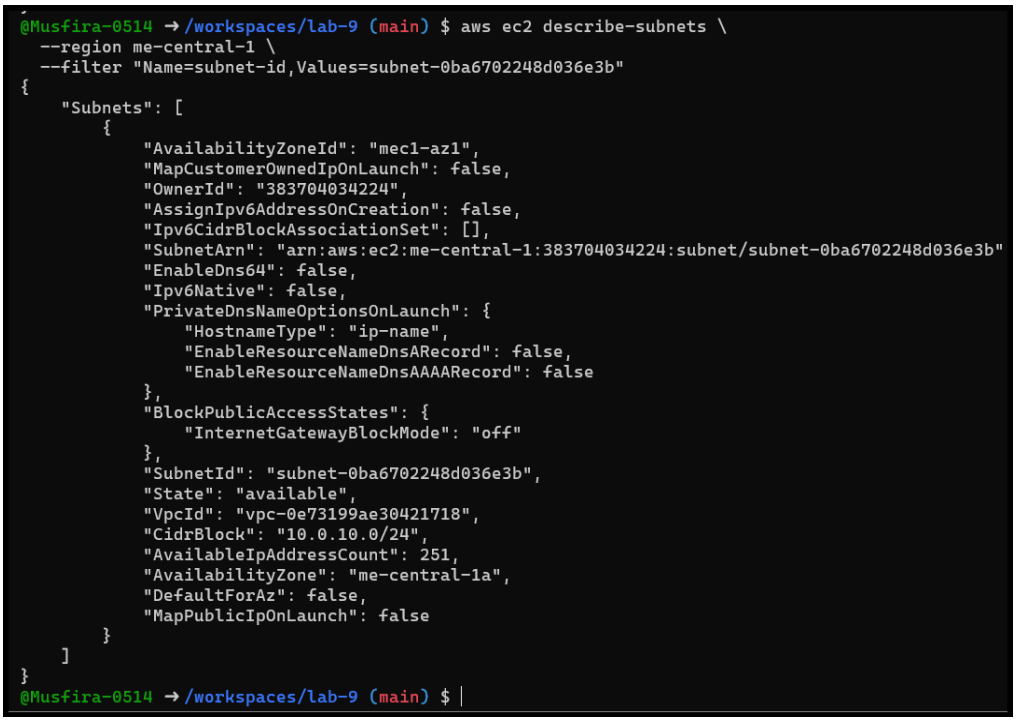
- Add VPC & subnet resources to main.tf

```
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"
}
```

- Terraform apply — VPC & subnet creation

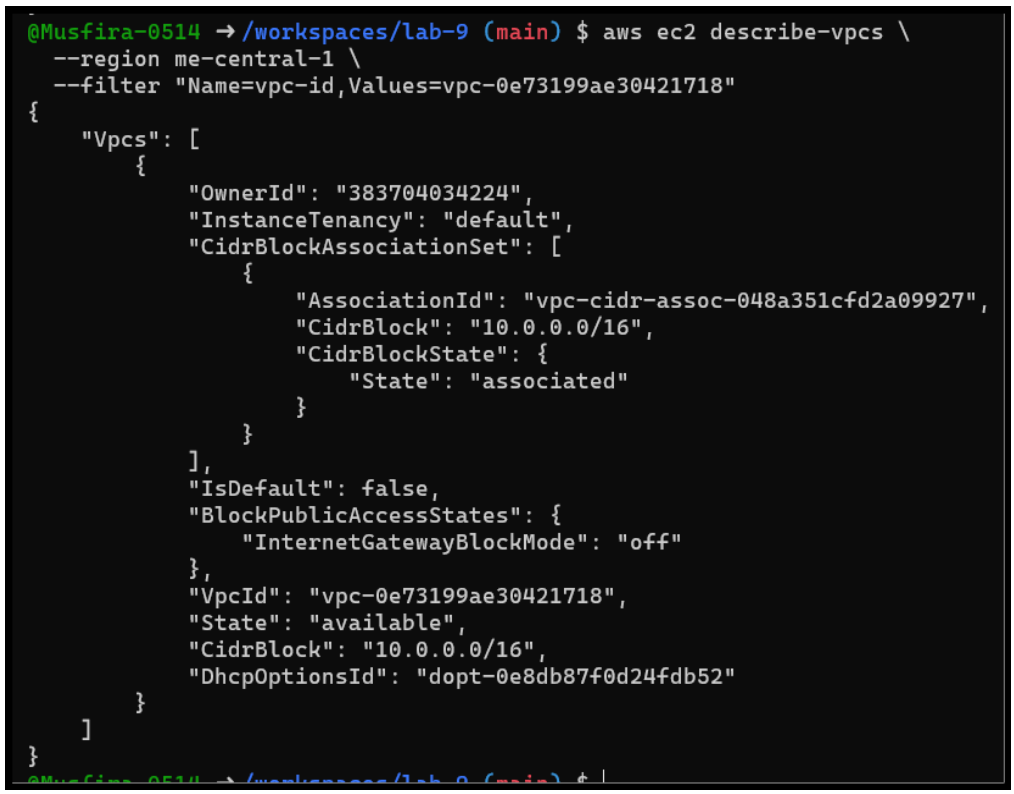
```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ terraform apply

Terraform used the selected providers to generate the following execution plan.
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                              = "me-central-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                                         = "me-central-1"
      + tags_all                                       = (known after apply)
      + vpc_id                                         = (known after apply)
    }

  # aws_vpc.development_vpc will be created
  + resource "aws_vpc" "development_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + region                               = "me-central-1"
      + tags_all                             = (known after apply)
    }

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

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

  Enter a value: yes

aws_vpc.development_vpc: Creating...
aws_vpc.development_vpc: Creation complete after 1s [id=vpc-0e73199ae30421718]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-0ba6702248d036e3b

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

- Verify subnet using AWS CLI



```
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 describe-subnets \
  --region me-central-1 \
  --filter "Name=subnet-id,Values=subnet-0ba6702248d036e3b"
{
    "Subnets": [
        {
            "AvailabilityZoneId": "mec1-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "383704034224",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:me-central-1:383704034224:subnet/subnet-0ba6702248d036e3b"
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-0ba6702248d036e3b",
            "State": "available",
            "VpcId": "vpc-0e73199ae30421718",
            "CidrBlock": "10.0.10.0/24",
            "AvailableIpAddressCount": 251,
            "AvailabilityZone": "me-central-1a",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": false
        }
    ]
}
@Musfira-0514 → /workspaces/lab-9 (main) $
```

- Verify VPC using AWS CLI

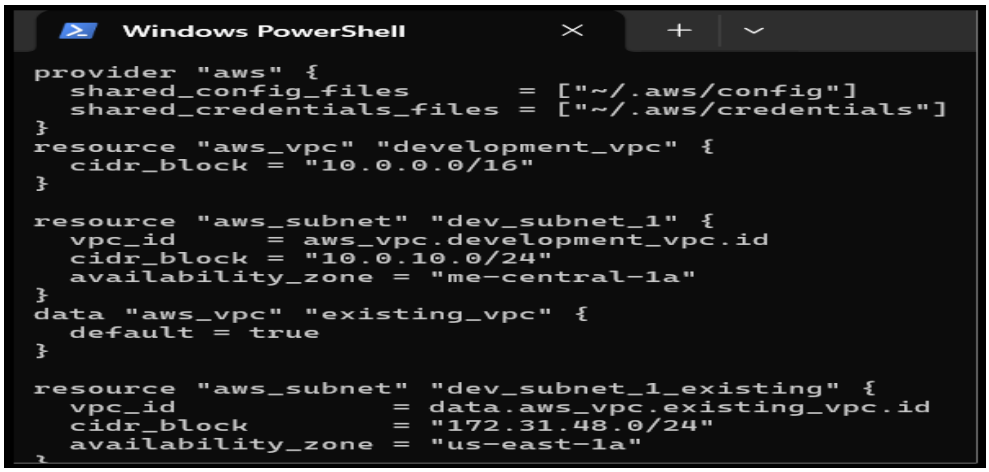


```
@Musfira-0514 → /workspaces/lab-9 (main) $ aws ec2 describe-vpcs \
  --region me-central-1 \
  --filter "Name=vpc-id,Values=vpc-0e73199ae30421718"
{
    "Vpcs": [
        {
            "OwnerId": "383704034224",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-048a351cfd2a09927",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-0e73199ae30421718",
            "State": "available",
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-0e8db87f0d24fdb52"
        }
    ]
}
```

Task 4 — Data Sources, Targeted Destroy & Tagging

Objective: Use data sources, targeted destroy, refresh, and tagging.

- Add data source & resource in main.tf



- Apply datasource-based resource

- Targeted destroy of subnet resource

```
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 th

  The -target option is not for routine use, and is provided only for exceptional si
  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

aws_subnet.dev_subnet_1_existing: Destroying... [id=subnet-0c1b915d665e0b4ac]
aws_subnet.dev_subnet_1_existing: Destruction complete after 2s

  Warning: Applied changes may be incomplete

  The plan was created with the -target option in effect, so some changes requested
  verify that no other changes are pending:
      terraform plan

  Note that the -target option is not suitable for routine use, and is provided only
  as part of an error message.


Destroy complete! Resources: 1 destroyed.
```

- Terraform refresh state

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0e73199ae30421718]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0ba6702248d036e3b]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0e1785b676d76a482]
```

- Terraform apply after refresh

```
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-0040fb57066a5710f]

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

- Destroy all resources

```
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.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_subnet.dev_subnet_1_existing: Destroying... [id=subnet-0040fb57066a5710f]
aws_subnet.dev_subnet_1: Destroying... [id=subnet-0ba6702248d036e3b]
aws_subnet.dev_subnet_1: Destruction complete after 0s
aws_vpc.development_vpc: Destroying... [id=vpc-0e73199ae30421718]
aws_vpc.development_vpc: Destruction complete after 1s
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
```

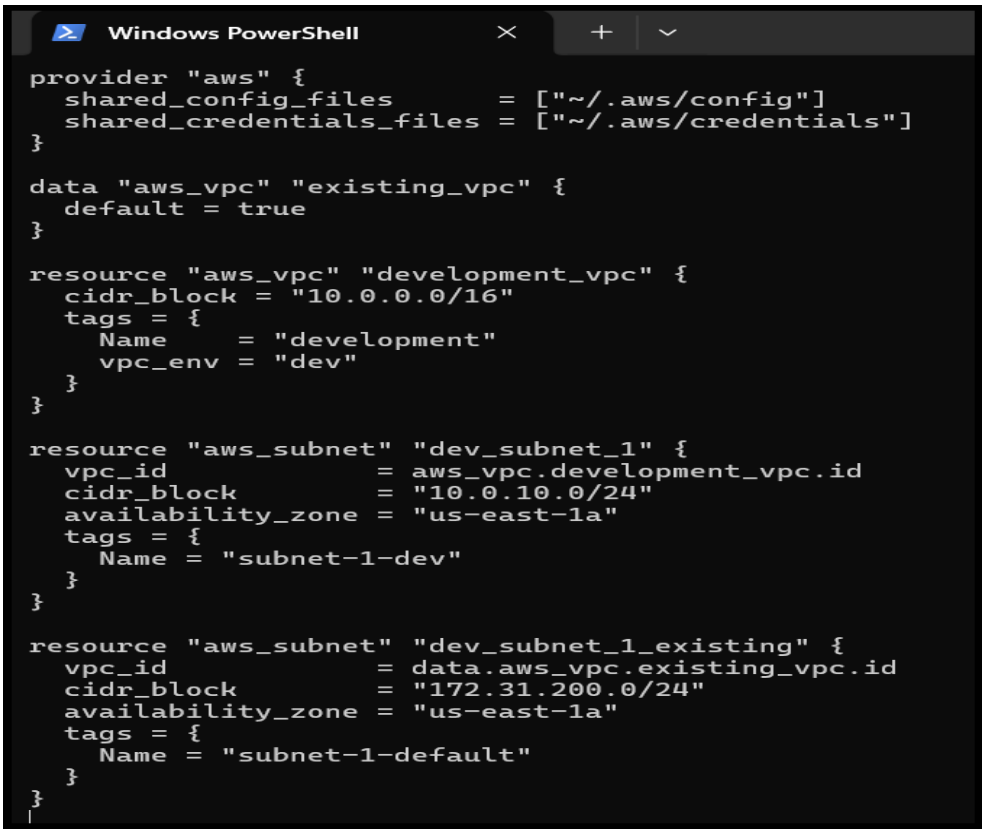
- Terraform plan output

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ terraform plan
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0e1785b676d76a482]

Terraform used the selected providers to generate the following execution plan.
  + 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                              = "me-central-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                                         = "us-east-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                              = "us-east-1a"
      + availability_zone_id                           = (known after apply)
      + cidr_block                                     = "172.31.200.0/24"
      + 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                                         = "us-east-1"
      + tags_all                                       = (known after apply)
      + vpc_id                                         = "vpc-0e1785b676d76a482
    }

  # aws_vpc.development_vpc will be created
  + resource "aws_vpc" "development_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (known after apply)
      + default_security_group_id            = (known after apply)
      + dhcp_options_id                      = (known after apply)
      + enable_dns_hostnames                 = (known after apply)
      + enable_dns_support                   = true
      + enable_network_address_usage_metrics = (known after apply)
      + id                                   = (known after apply)
      + instance_tenancy                     = "default"
      + ipv6_association_id                  = (known after apply)
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + region                               = "us-east-1"
      + tags_all                             = (known after apply)
    }

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

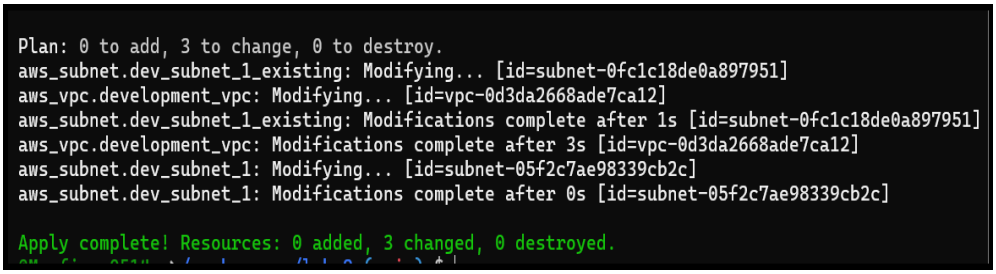
- Terraform apply after destroy

```
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 2s [id=subnet-05f2c7ae98339cb2c]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Resource tagging in main.tf



- Apply tagging changes



- Plan tag removal

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ vim main.tf
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ terraform plan
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0d3da2668ade7ca12]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0e1785b676d76a482]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0fc1c18de0a897951]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-05f2c7ae98339cb2c]

Terraform used the selected providers to generate the following execution plan. Resource
  ~ 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-0d3da2668ade7ca12"
      ~ 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.
```

- Apply tag removal

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

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

  Enter a value: yes

aws_vpc.development_vpc: Modifying... [id=vpc-0d3da2668ade7ca12]
aws_vpc.development_vpc: Modifications complete after 3s [id=vpc-0d3da2668ade7ca12]

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

Task 5 — Terraform State File Inspection

Objective: Inspect Terraform state and backup files.

- Destroy all resources

```
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.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes

aws_subnet.dev_subnet_1_existing: Destroying... [id=subnet-0fc1c18de0a897951]
aws_subnet.dev_subnet_1: Destroying... [id=subnet-05f2c7ae98339cb2c]
aws_subnet.dev_subnet_1: Destruction complete after 2s
aws_vpc.development_vpc: Destroying... [id=vpc-0d3da2668ade7ca12]
aws_subnet.dev_subnet_1_existing: Destruction complete after 2s
aws_vpc.development_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Empty terraform.tfstate file

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 34,
  "lineage": "ac3fff77-6008-5282-e5b8-332f0327d8c0",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- terraform.tfstate.backup (previous state)

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 29,
  "lineage": "ac3fff77-6008-5282-e5b8-332f0327d8c0",
  "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:us-east-1:383704034224:vpc/vpc-0e1785b676d76a482",
            "cidr_block": "172.31.0.0/16",
            "cidr_block_associations": [
              {
                "association_id": "vpc-cidr-assoc-064d2b7cac5f2375b",
                "cidr_block": "172.31.0.0/16",
                "state": "associated"
              }
            ],
```

```
  "mode": "managed",
  "type": "aws_subnet",
  "name": "dev_subnet_1",
  "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
  "instances": [
    {
      "schema_version": 1,
      "attributes": {
        "arn": "arn:aws:ec2:us-east-1:383704034224:subnet/subnet-05f2c7ae98339cb2c",
        "assign_ipv6_address_on_creation": false,
        "availability_zone": "us-east-1a",
        "availability_zone_id": "use1-az1",
        "cidr_block": "10.0.10.0/24",
        "customer_owned_ipv4_pool": "",
        "enable_dns64": false,
        "enable_lni_at_device_index": 0,
        "enable_resource_name_dns_a_record_on_launch": false,
        "enable_resource_name_dns_aaaa_record_on_launch": false,
        "id": "subnet-05f2c7ae98339cb2c",
        "ipv6_cidr_block": "",
        "ipv6_cidr_block_association_id": "",
        "ipv6_native": false,
        "map_customer_owned_ip_on_launch": false,
        "map_public_ip_on_launch": false,
        "outpost_arn": "",
        "owner_id": "383704034224",
        "private_dns_hostname_type_on_launch": "ip-name",
        "region": "us-east-1",
        "tags": {
          "Name": "subnet-1-dev"
        },
        "tags_all": {
          "Name": "subnet-1-dev"
        },
        "timeouts": null,
        "vpc_id": "vpc-0d3da2668ade7ca12"
      },
```

- Recreate resources using terraform apply

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

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

  Enter a value: yes

aws_vpc.development_vpc: Creating...
aws_subnet.dev_subnet_1_existing: Creating...
aws_subnet.dev_subnet_1_existing: Creation complete after 1s [id=subnet-05c17d038f17ac539]
aws_vpc.development_vpc: Creation complete after 3s [id=vpc-0068f2ef0b1f9fb26]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-0300cd8b41e163fbd]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Populated terraform.tfstate file

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 38,
  "lineage": "ac3fff77-6008-5282-e5b8-332f0327d8c0",
  "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:us-east-1:383704034224:vpc/vpc-0e1785b676d76a482",
            "cidr_block": "172.31.0.0/16",
            "cidr_block_associations": [
              {
                "association_id": "vpc-cidr-assoc-064d2b7cac5f2375b",
                "cidr_block": "172.31.0.0/16",
                "state": "associated"
              }
            ],
```

```
,,
{
  "mode": "managed",
  "type": "aws_subnet",
  "name": "dev_subnet_1",
  "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
  "instances": [
    {
      "schema_version": 1,
      "attributes": {
        "arn": "arn:aws:ec2:us-east-1:383704034224:subnet/subnet-0300cd8b41e163fbd",
        "assign_ipv6_address_on_creation": false,
        "availability_zone": "us-east-1a",
        "availability_zone_id": "use1-az1",
        "cidr_block": "10.0.10.0/24",
        "customer_owned_ipv4_pool": "",
        "enable_dns64": false,
        "enable_lni_at_device_index": 0,
        "enable_resource_name_dns_a_record_on_launch": false,
        "enable_resource_name_dns_aaaa_record_on_launch": false,
        "id": "subnet-0300cd8b41e163fbd",
        "ipv6_cidr_block": "",
        "ipv6_cidr_block_association_id": "",
        "ipv6_native": false,
        "map_customer_owned_ip_on_launch": false,
        "map_public_ip_on_launch": false,
        "outpost_arn": "",
        "owner_id": "383704034224",
        "private_dns_hostname_type_on_launch": "ip-name",
        "region": "us-east-1",
        "tags": {
          "Name": "subnet-1-dev"
        },
        "tags_all": {
          "Name": "subnet-1-dev"
        },
        "timeouts": null,
        "vpc_id": "vpc-0068f2ef0b1f9fb26"
      },
      "sensitive_attributes": [],
      "identity_schema_version": 0,
      "identity": {
        "account_id": "383704034224",
        "id": "subnet-0300cd8b41e163fbd",
```

- Empty backup after restore

```
,
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 34,
  "lineage": "ac3fff77-6008-5282-e5b8-332f0327d8c0",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Terraform state list

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ terraform state list
data.aws_vpc.existing_vpc
aws_subnet.dev_subnet_1
aws_subnet.dev_subnet_1_existing
aws_vpc.development_vpc
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Terraform state show resource

```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ terraform state show aws_vpc.development_vpc
# aws_vpc.development_vpc:
resource "aws_vpc" "development_vpc" {
    arn                                  = "arn:aws:ec2:us-east-1:383704034224:vpc/vpc-0068f2ef0b1f9fb26"
    assign_generated_ipv6_cidr_block     = false
    cidr_block                           = "10.0.0.0/16"
    default_network_acl_id               = "acl-0544b7f36cfdbff81"
    default_route_table_id               = "rtb-0a57d52e055f0ac4c"
    default_security_group_id            = "sg-018d6200bc3d2e900"
    dhcp_options_id                      = "dopt-0dd4d2d7dafce6976"
    enable_dns_hostnames                 = false
    enable_dns_support                   = true
    enable_network_address_usage_metrics = false
    id                                   = "vpc-0068f2ef0b1f9fb26"
    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-0a57d52e055f0ac4c"
    owner_id                             = "383704034224"
    region                               = "us-east-1"
    tags                                 = {
        "Name" = "development"
    }
    tags_all                             = {
        "Name" = "development"
    }
}
```

Task 6 — Terraform Outputs & Attributes

Objective: Extract and report resource attributes using Terraform outputs.

- Basic outputs (IDs & ARNs)

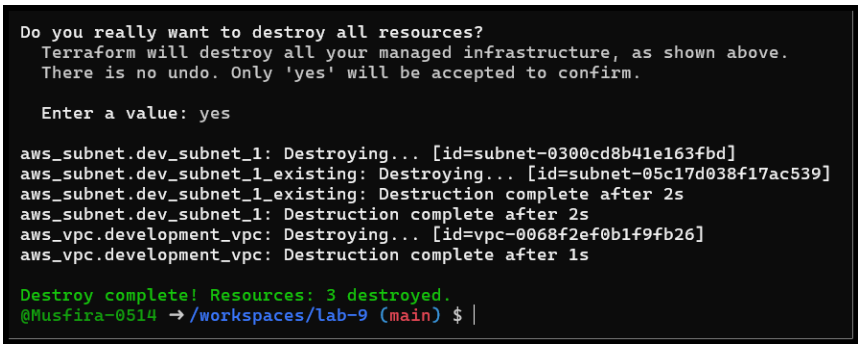
```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes


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

Outputs:

dev-subnet-arn = "arn:aws:ec2:us-east-1:383704034224:subnet/subnet-0300cd8b41e163fbd"
dev-subnet-id = "subnet-0300cd8b41e163fbd"
dev-vpc-arn = "arn:aws:ec2:us-east-1:383704034224:vpc/vpc-0068f2ef0b1f9fb26"
dev-vpc-id = "vpc-0068f2ef0b1f9fb26"
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Expanded outputs (CIDR, region, tags, tags_all)

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

Outputs:

dev-subnet-arn = "arn:aws:ec2:us-east-1:383704034224:subnet/subnet-0300cd8b41e163fbd"
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-id = "subnet-0300cd8b41e163fbd"
dev-subnet-region = "us-east-1a"
dev-subnet-tags_all = tomap({
  "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-arn = "arn:aws:ec2:us-east-1:383704034224:vpc/vpc-0068f2ef0b1f9fb26"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-id = "vpc-0068f2ef0b1f9fb26"
dev-vpc-region = "us-east-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
```

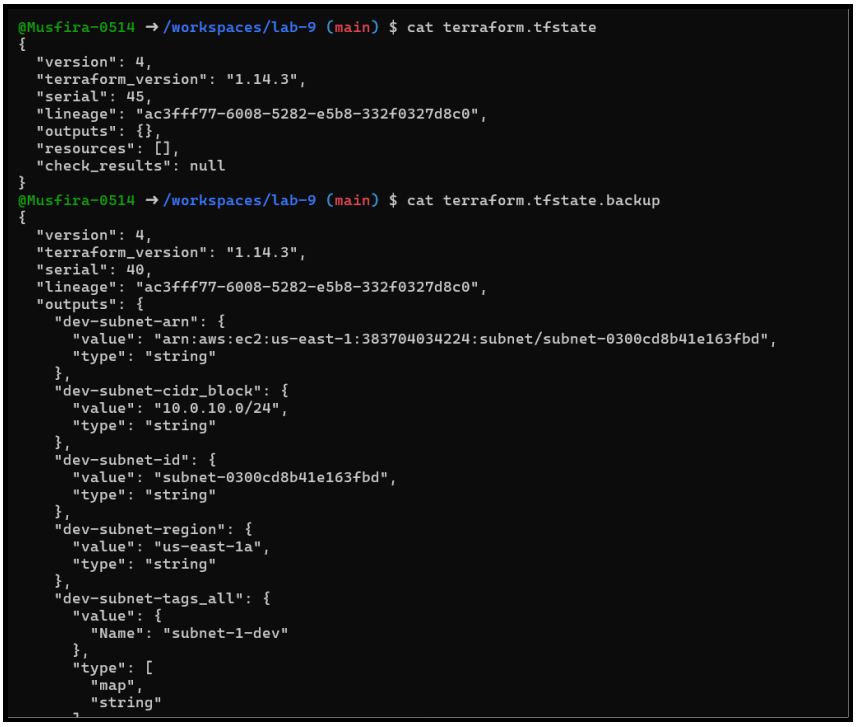
Cleanup — Remove All Resources & Verify State

- Destroy all Terraform-managed resources



```
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

aws_subnet.dev_subnet_1: Destroying... [id=subnet-0300cd8b41e163fbd]
aws_subnet.dev_subnet_1_existing: Destroying... [id=subnet-05c17d038f17ac539]
aws_subnet.dev_subnet_1_existing: Destruction complete after 2s
aws_subnet.dev_subnet_1: Destruction complete after 2s
aws_vpc.development_vpc: Destroying... [id=vpc-0068f2ef0b1f9fb26]
aws_vpc.development_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
@Musfira-0514 ➜ /workspaces/lab-9 (main) $
```

- Verify state files after cleanup



```
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 45,
  "lineage": "ac3fff77-6008-5282-e5b8-332f0327d8c0",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Musfira-0514 ➜ /workspaces/lab-9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 40,
  "lineage": "ac3fff77-6008-5282-e5b8-332f0327d8c0",
  "outputs": {
    "dev-subnet-arn": {
      "value": "arn:aws:ec2:us-east-1:383704034224:subnet/subnet-0300cd8b41e163fbd",
      "type": "string"
    },
    "dev-subnet-cidr_block": {
      "value": "10.0.10.0/24",
      "type": "string"
    },
    "dev-subnet-id": {
      "value": "subnet-0300cd8b41e163fbd",
      "type": "string"
    },
    "dev-subnet-region": {
      "value": "us-east-1a",
      "type": "string"
    },
    "dev-subnet-tags_all": {
      "value": {
        "Name": "subnet-1-dev"
      },
      "type": [
        "map",
        "string"
```