

Name: Sadaf Riaz

Roll no: 2023-BSE-077

Section: BSE5-B

Lab 10

Task 1 — GitHub CLI Codespace Setup & Authentication

- **1.1 Installing GitHub CLI — Screenshot: task1_gh_install.png**

```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

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

- **1.2 Authenticating GH CLI — Screenshot: task1_gh_auth_login.png**

```
Press Enter to open https://github.com/login/device in your browser...
PS C:\Users\Lenovo> 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: 5DFC-EBE6
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
✓ Logged in as SadafRiaz-077
You were already logged in to this account
PS C:\Users\Lenovo>
```

- **1.3 Listing Codespaces — Screenshot: task1_codespace_list.png**

```
@SadafRiaz-077 ~ /workspaces/lab-10 (main) $ gh codespace list
NAME          DISPLAY NAME      REPOSITORY      BRANCH STATE      CREATED AT
shiny-couscous-wrq7qxp9p5wqh575g shiny couscous SadafRiaz-077/lab-10 main Available about 14 minutes ago
@SadafRiaz-077 ~ /workspaces/lab-10 (main) $
```

- **1.4 Connecting to Codespace via SSH — Screenshot:**
task1_codespace_ssh_connected.png

```
shiny-couscous-wrq7qxp9p5wqh575g shiny couscous SadafRiaz-077/lab-10 main Available about 14 min
@SadafRiaz-077 → /workspaces/lab-10 (main) $ gh codespace ssh -c shiny-couscous-wrq7qxp9p5wqh575g
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: Wed Dec 10 16:33:11 2025 from ::1
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

Task 2 — AWS CLI, Terraform CLI & Provider Setup

- **2.1 Installing AWS CLI — Screenshot: task2_aws_install_and_version.png**

```
inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
@SadafRiaz-077 → /workspaces/lab-10 (main) $ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
@SadafRiaz-077 → /workspaces/lab-10 (main) $ aws --version
aws-cli/2.32.13 Python/3.13.9 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **2.2 Configuring AWS CLI — Screenshot: task2_aws_configure_and_files.png**

```
Last login: Wed Dec 10 16:33:40 2025 from ...
@SadafRiaz-077 → /workspaces/lab-10 (main) $ aws configure
AWS Access Key ID [None]: AKIAV4QTXR5SQMMRGNPB
AWS Secret Access Key [None]: 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
Default region name [None]: me-central-1
Default output format [None]: json
```

- **2.3 Verifying AWS Caller Identity — Screenshot: task2_aws_get_caller_identity.png**

```
@SadafRiaz-077 → /workspaces/lab-10 (main) $ aws sts get-caller-identity
{
    "UserId": "AIDAV4QTXR5S5GAC37G6Z",
    "Account": "404842057573",
    "Arn": "arn:aws:iam::404842057573:user/lab10-user"
}
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **2.4 Installing Terraform CLI — Screenshot:**

task2_terraform_install_and_version.png

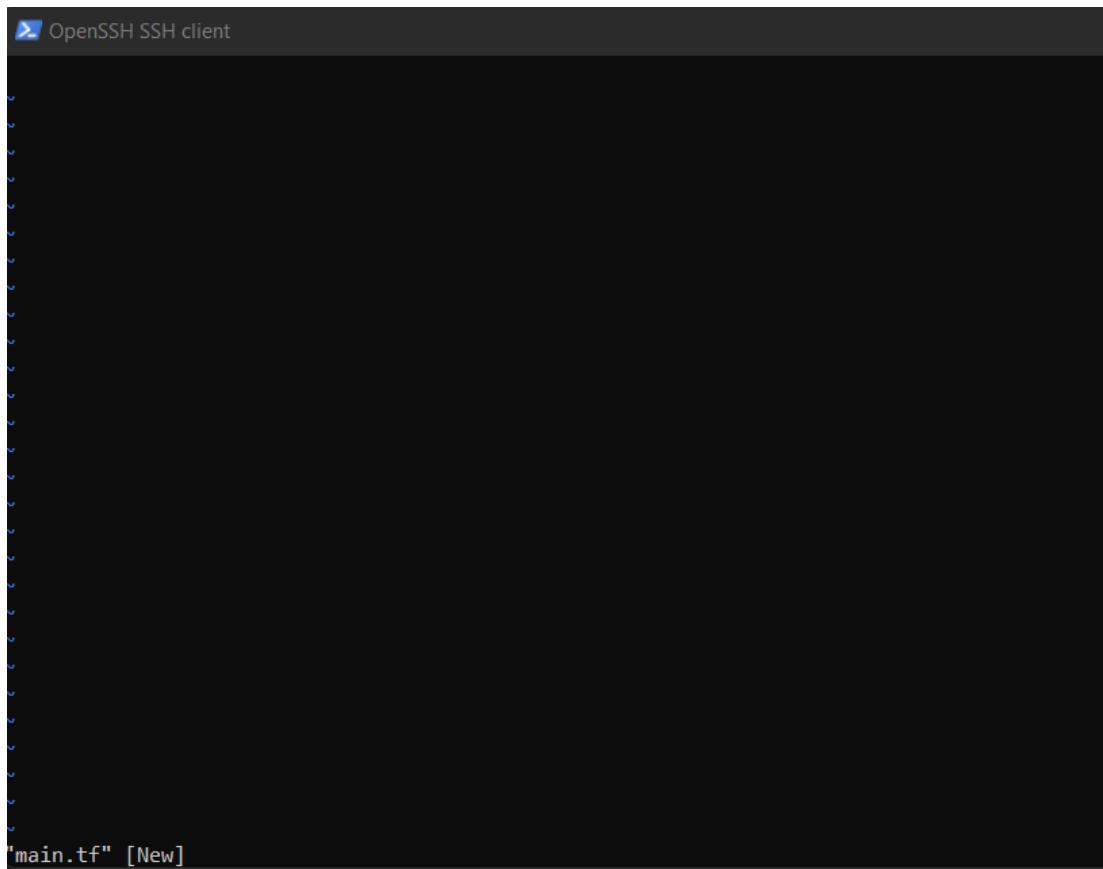
```
@SadafRiaz-077 ~ /workspaces/lab-10 (main) $ wget -O - https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
architecture signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(grep -oP '(?<=UBUNTU_CODENAME)=.*' /etc/os-release || lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt update
sudo apt install terraform
which terraform
terraform --version--2025-12-10 17:38:42-- https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 18.172.78.30, 18.172.78.65, 18.172.78.129, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)[18.172.78.30]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

[Progress Bar] 100%[=====] 3.89K ---KB/s in 0s

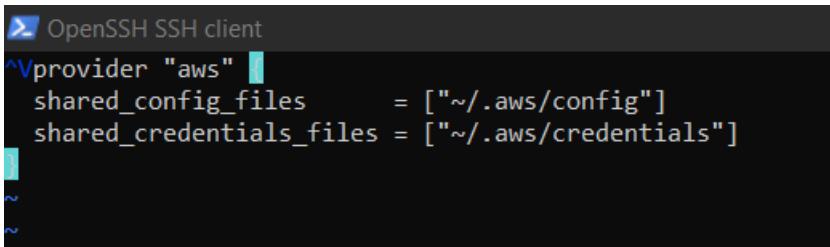
2025-12-10 17:38:43 (737 MB/s) - written to stdout [3980/3980]

@SadafRiaz-077 ~ /workspaces/lab-10 (main) $ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(grep -oP '(?<=UBUNTU_CODENAME)=.*' /etc/os-release || lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list
deb [arch=amd64 signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com noble main
@SadafRiaz-077 ~ /workspaces/lab-10 (main) $ sudo apt update
```

- **2.5 Creating main.tf File — Screenshot: task2_provider_file_creation.png**

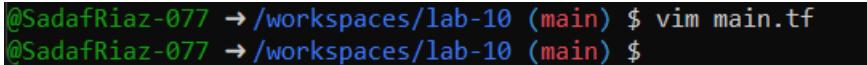


- 2.6 Adding Provider Block — Screenshot: task2_provider_block.png



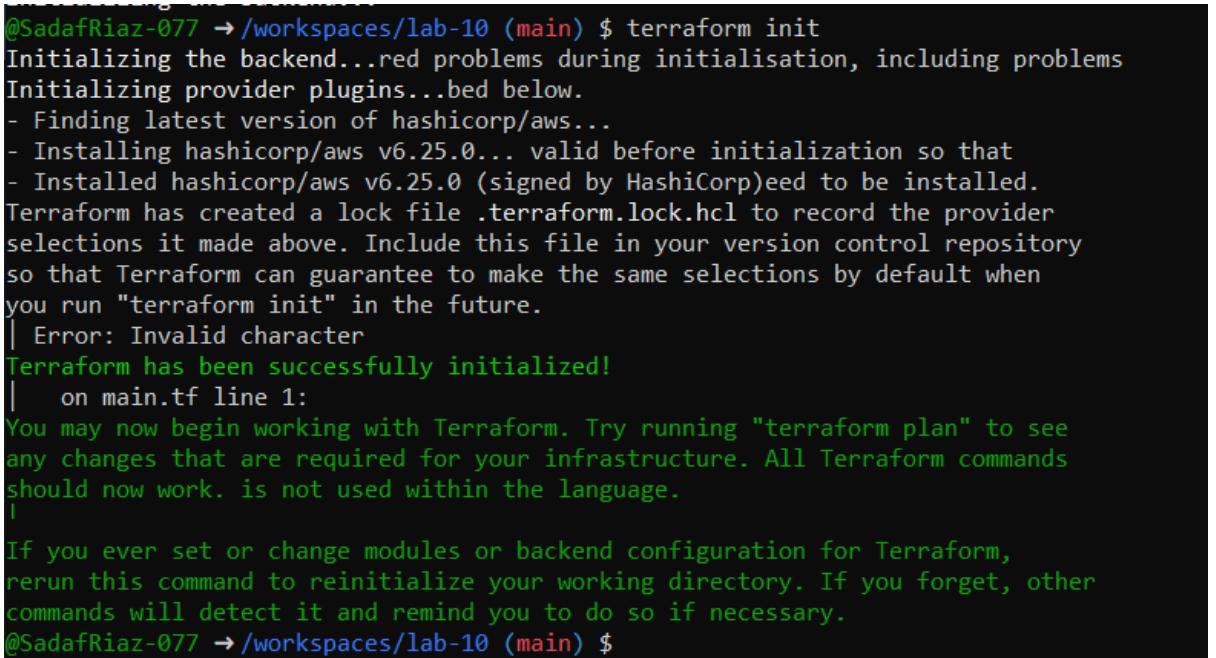
```
(provider "aws"
  shared_config_files      = ["~/.aws/config"]
  shared_credentials_files = ["~/.aws/credentials"])
```

- 2.7 Saving main.tf in Vim — Screenshot: task2_vim_save_main_tf.png



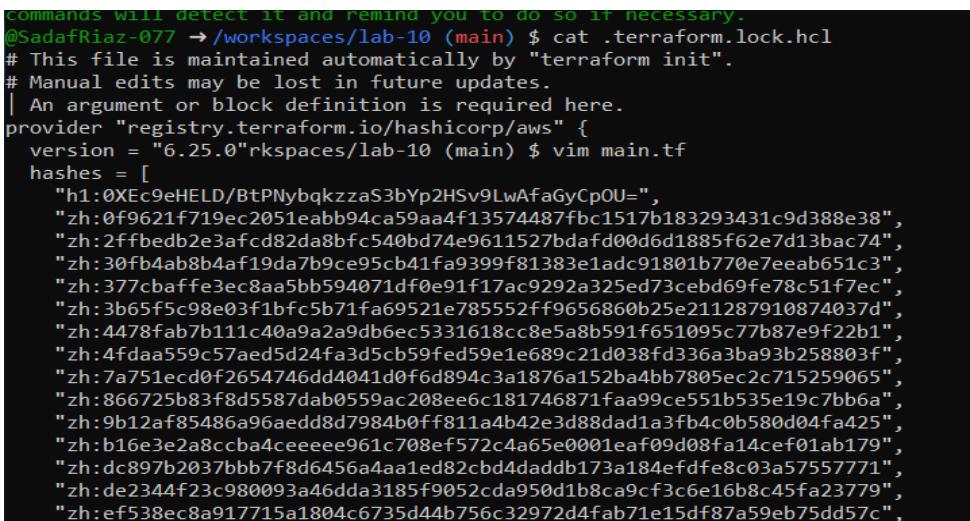
```
@SadafRiaz-077 → /workspaces/lab-10 (main) $ vim main.tf
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- 2.8 Running terraform init — Screenshot: task2_terraform_init_output.png



```
@SadafRiaz-077 → /workspaces/lab-10 (main) $ terraform init
Initializing the backend...red problems during initialisation, including problems
Initializing provider plugins...bed below.
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.25.0... valid before initialization so that
- Installed hashicorp/aws v6.25.0 (signed by HashiCorp)eed to be installed.
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.
| Error: Invalid character
Terraform has been successfully initialized!
|   on main.tf line 1:
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. is not used within the language.
|
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.
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- 2.9 Viewing .terraform.lock.hcl — Screenshot: task2_terraform_lock_hcl.png



```
Commands will detect it and remind you to do so if necessary.
@SadafRiaz-077 → /workspaces/lab-10 (main) $ cat .terraform.lock.hcl
# This file is maintained automatically by "terraform init".
# Manual edits may be lost in future updates.
| An argument or block definition is required here.
provider "registry.terraform.io/hashicorp/aws" {
  version = "6.25.0"
  hashes = [
    "h1:0XEc9eHELD/BtPNybqkzzaS3bYp2HSv9LwAfaGyCpOU=",
    "zh:0F9621f719ec2051eabb94ca59aa4f13574487fbc1517b183293431c9d388e38",
    "zh:2ffbedb2e3afcd82da8bfc540bd74e9611527bdafd00d6d1885f62e7d13bac74",
    "zh:30fb4ab8b4af19da7b9ce95cb41fa9399f81383e1adc91801b770e7eeab651c3",
    "zh:377cbaffe3ec8aa5bb594071df0e91f17ac9292a325ed73cebd69fe78c51f7ec",
    "zh:3b65f5c98e03f1bfc5b71fa69521e785552ff9656860b25e211287910874037d",
    "zh:4478fab7b111c40a9a2a9db6ec5331618cc8e5a8b591f651095c77b87e9ff22b1",
    "zh:4fdaa559c57aed5d24fa3d5cb59fed59e1e689c21d038fd336a3ba93b258803f",
    "zh:7a751ecd0f2654746dd4041d0f6d894c3a1876a152ba4bb7805ec2c715259065",
    "zh:866725b83f8d5587dab0559ac208ee6c181746871faa99ce551b535e19c7bb6a",
    "zh:9b12af85486a96aedd8d7984b0ff811a4b42e3d88dad1a3fb4c0b580d04fa425",
    "zh:b16e3e2a8ccba4ceeeee961c708ef572c4a65e0001eaf09d08fa14cef01ab179",
    "zh:dc897b2037bbbb7f8d6456a4aa1ed82cbd4daddb173a184efdfe8c03a57557771",
    "zh:de2344f23c980093a46dda3185f9052cda950d1b8ca9cf3c6e16b8c45fa23779",
    "zh:ef538ec8a917715a1804c6735d44b756c32972d4fab71e15df87a59eb75dd57c",
```

- 2.10 Listing .terraform Directory — Screenshot: task2_terraform_dir_ls.png

```

}
@SadafRiaz-077 → /workspaces/lab-10 (main) $ ls .terraform/
providers
@SadafRiaz-077 → /workspaces/lab-10 (main) $

```

Task 3 — VPC & Subnet Creation + Verification

- 3.1 Adding VPC and Subnet Code — Screenshot: task3_main_tf_resource_add.png

```

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

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"
}
~
```

- 3.2 Running terraform apply — Screenshot:

task3_terraform_apply_vpc_subnet.png

```

+ ipv4_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 3s [id=vpc-07078de37bd1a9fb4]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-08e72f5b69df420ee]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **3.3 Verifying Subnet Using AWS CLI — Screenshot:**

task3_aws_cli_verify_subnet.png

```
SadafRiaz-077 → /workspaces/lab-10 (main) $ aws ec2 describe-subnets --filter "Name=subnet-id,Values=subnet-08e72f5b69df420ee"
{
    "Subnets": [
        {
            "AvailabilityZoneId": "mecl-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "404842057573",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:me-central-1:404842057573:subnet/subnet-08e72f5b69df420ee",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-08e72f5b69df420ee",
            "State": "available",
            "VpcId": "vpc-07078de37bd1a9fb4",
            "CidrBlock": "10.0.10.0/24",
            "AvailableIpAddressCount": 251,
            "AvailabilityZone": "me-central-1a"
        }
    ]
}
```

- **3.4 Verifying VPC Using AWS CLI — Screenshot: task3_aws_cli_verify_vpc.png**

```
SadafRiaz-077 → /workspaces/lab-10 (main) $ aws ec2 describe-vpcs --filter "Name=vpc-id,Values=vpc-07078de37bd1a9fb4"
{
    "Vpcs": [
        {
            "OwnerId": "404842057573",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-0e98faeced65c0524",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-07078de37bd1a9fb4",
            "State": "available",
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-00025b89642cbf410"
        }
    ]
}
```

Task 4 — Data Source, Targeted Destroy & Tags

- **4.1 Adding Data Source + New Subnet**

Screenshot:task4_main_tf_datasource_resource_add.png

```

    cidr_block = "10.0.10.0/24"
    availability_zone = "me-central-1a"
}
odata "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 = "me-central-1a"
}

```

-- TINSERT --

- **4.2 Applying Data Source Resource — Screenshot:**

task4_terraform_apply_datasource_resource.png

```

+ map_public_ip_on_launch           = raise
+ owner_id                          = (known after apply)
+ private_dns_hostname_type_on_launch = raise
+ region                            = "me-central-1"
+ tags_all                           = (known after apply)
+ vpc_id                             = "vpc-0ed011693d93d59bc"
}

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 0s [id=subnet-0a8704aa36ab04cb9]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **4.3 Targeted Destroy Command :Screenshot:task4_terraform_destroy_targeted.png**

```

Warning: Applied changes may be incomplete

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

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

Destroy complete! Resources: 1 destroyed.
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **4.4 Running terraform refresh — Screenshot: task4_terraform_refresh_state.png**

```
destroy completed: resources: 1 destroyed.
@SadafRiaz-077 → /workspaces/lab-10 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-07078de37bd1a9fb4]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0ed011693d93d59bc]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-08e72f5b69df420ee]
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **4.5 terraform apply After Refresh —**

Screenshot:task4_terraform_apply_after_refresh.png

```
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-061cd68a6c3dfc949]

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

- **4.6 Destroying All Resources — Screenshot: task4_terraform_destroy_all.png**

```
- enable_dns_support          = true -> null
- enable_network_address_usage_metrics = false -> null
- id                         = "vpc-07078de37bd1a9fb4" -> null
- instance_tenancy           = "default" -> null
- ipv6_netmask_length        = 0 -> null
- main_route_table_id        = "rtb-0deb8ab3fdcaa9cb2b" -> null
- owner_id                   = "404842057573" -> null
- region                     = "me-central-1" -> null
- tags                       = {} -> null
- tags_all                   = {} -> null
  # (4 unchanged attributes hidden)
}

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-061cd68a6c3dfc949]
aws_subnet.dev_subnet_1: Destroying... [id=subnet-08e72f5b69df420ee]
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s
aws_subnet.dev_subnet_1: Destruction complete after 1s
aws_vpc.development_vpc: Destroying... [id=vpc-07078de37bd1a9fb4]
aws_vpc.development_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
```

- **4.7 terraform plan Output — Screenshot: task4_terraform_plan_output.png**

```
@SadafRiaz-077 → /workspaces/lab-10 (main) $ terraform plan
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0ed011693d93d59bc]

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                 = "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_subnet.dev_subnet_1_existing will be created
```

- **4.8 terraform apply After Destroy — Screenshot:**

task4_terraform_apply_after_destroy.png

```
+ uncp_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: 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 0s [id=subnet-0fafaab0b8e0461ae]
aws_vpc.development_vpc: Creation complete after 1s [id=vpc-001ca861a575d1b1b]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-03f3cc3d34a7e7855]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **4.9 Adding Tags to Resources — Screenshot: task4_main_tf_tagging.png**

```

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 = "me-central-1a"

  tags = {
    Name = "subnet-1-dev"
  }
}

# Existing VPC data source
data "aws_vpc" "existing_vpc" {
  default = true
}

resource "aws_subnet" "dev_subnet_1_existing" {
  vpc_id      = data.aws_vpc.existing_vpc.id
  cidr_block  = "10.0.20.0/24" # avoid conflicting default VPC range
  availability_zone = "me-central-1a"

  tags = {
    Name = "subnet-1-default"
  }
}
~
```

- **4.10 Applying Tags — Screenshot: task4_terraform_apply_tagging.png**

```

Note: You didn't use the -out option to save this plan, so Terraform can't guarantee to take exactly these actions
@sadariaz-077 ~ /workspaces/lab-10 (main) $ terraform apply -auto-approve
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-001ca861a575d1b1b]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-03f3cc3d3a7e7855]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0ed011693d93d59bc]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with:
+ 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                   = "me-central-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                            = "me-central-1"
      + tags                               = {
          + "Name" = "subnet-1-default"
        }
      + tags_all                           = {
          + "Name" = "subnet-1-default"
        }
      + vpc_id                            = "vpc-0ed011693d93d59bc"
    }

Plan: 1 to add, 0 to change, 0 to destroy.
aws_subnet.dev_subnet_1_existing: Creating...
aws_subnet.dev_subnet_1_existing: Creation complete after 1s [id=subnet-070168b8ce69b2181]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@sadariaz-077 ~ /workspaces/lab-10 (main) $
```

- **4.11 Removing Tag (vpc_env) — Screenshot:**
task4_terraform_plan_remove_tag.png

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@sadafriaz-077 → /workspaces/lab-10 (main) $ terraform plan
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-001ca861a575d1b1b]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0ed011693d93d59bc]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-070168b8ce69b2181]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-03f3cc3d34a7e7855]

No changes. Your infrastructure matches the configuration.
```

- **4.12 terraform apply After Tag Removal — Screenshot:**
task4_terraform_apply_remove_tag.png

```
@sadafriaz-077 → /workspaces/lab-10 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-001ca861a575d1b1b]
data.aws_vpc.existing_vpc: Read complete after 2s [id=vpc-0ed011693d93d59bc]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-070168b8ce69b2181]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-03f3cc3d34a7e7855]

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.
@sadafriaz-077 → /workspaces/lab-10 (main) $
```

Task 5 — State Inspection & Terraform State Commands

- **5.1 Destroying All Resources — Screenshot: task5_terraform_destroy.png**

```
# aws_vpc.development_vpc will be destroyed
- resource "aws_vpc" "development_vpc" {
    - arn
    - assign_generated_ipv6_cidr_block
    - cidr_block
    - default_network_acl_id
    - default_route_table_id
    - default_security_group_id
    - dhcp_options_id
    - enable_dns_hostnames
    - enable_dns_support
    - enable_network_address_usage_metrics
    - id
    - instance_tenancy
    - ipv6_netmask_length
    - main_route_table_id
    - owner_id
    - region
    - tags
        - "Name" = "development"
        - "vpc_env" = "dev"
    } -> null
- tags_all
    - "Name" = "development"
    - "vpc_env" = "dev"
} -> null
# (4 unchanged attributes hidden)
}

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-070168b8ce69b2181]
aws_subnet.dev_subnet_1: Destroying... [id=subnet-03f3cc3d34a7e7855]
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s
aws_subnet.dev_subnet_1: Destruction complete after 1s
aws_vpc.development_vpc: Destroying... [id=vpc-001ca861a575d1b1b]
aws_vpc.development_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
```

- **5.2 terraform.tfstate After Destroy — Screenshot:**

task5_terraform_state_file_empty.png

```
Destroy complete! Resources: 3 destroyed.
@sadafriaz-077 → /workspaces/lab-10 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 32,
  "lineage": "5f0f5fe9-8b91-d760-48dd-3bfb91f9e713",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@sadafriaz-077 → /workspaces/lab-10 (main) $
```

- **5.3 terraform.tfstate.backup Contents — Screenshot:**

task5_terraform_state_backup_prev.png

```
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 27,
  "lineage": "5f0f5fe9-8b91-d760-48dd-3bfb91f9e713",
  "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:me-central-1:404842057573:vpc/vpc-0ed011693d93d59bc",
            "cidr_block": "172.31.0.0/16",
            "cidr_block_associations": [
              {
                "association_id": "vpc-cidr-assoc-0fc13ce2206ae6fe2",
                "cidr_block": "172.31.0.0/16",
                "state": "associated"
              }
            ],
            "default": true,
            "dhcp_options_id": "dopt-00025b89642cbf410",
            "enable_dns_hostnames": true,
            "enable_dns_support": true,
            "enable_network_address_usage_metrics": false,
            "filter": null,
            "id": "vpc-0ed011693d93d59bc",
            "instance_tenancy": "default",
            "ipv6_association_id": "",
            "ipv6_cidr_block": "",
            "main_route_table_id": "rtb-0328c2e3c3d0f1551",
            "owner_id": "404842057573",
            "region": "me-central-1",
            "status": "available"
          }
        }
      ]
    }
  ]
}
```

- **5.4 Recreating Resources — Screenshot: task5_terraform_apply_recreated.png**

```

}
@SadafRiaz-077 → /workspaces/lab-10 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0ed011693d93d59bc]

Terraform used the selected providers to generate the following execution plan. Resource actions are
+ 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
        + "Name" = "subnet-1-dev"
    }
    + tags_all
        + "Name" = "subnet-1-dev"
    }
    + vpc_id                            = (known after apply)
}

# aws_subnet.dev_subnet_1_existing will be created

```

- **5.5 terraform.tfstate Populated — Screenshot: task5_terraform_state_file_populated.png**

```

}
@SadafRiaz-077 → /workspaces/lab-10 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 36,
  "lineage": "5f0f5fe9-8b91-d760-48dd-3bfb91f9e713",
  "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:me-central-1:404842057573:vpc/vpc-0ed011693d93d59bc",
            "cidr_block": "172.31.0.0/16",
            "cidr_block_associations": [
              {
                "association_id": "vpc-cidr-assoc-0fc13ce2206ae6fe2",
                "cidr_block": "172.31.0.0/16",
                "state": "associated"
              }
            ],
            "default": true,
            "dhcp_options_id": "dopt-00025b89642cbf410",
            "enable_dns_hostnames": true,
            "enable_dns_support": true,
            "enable_network_address_usage_metrics": false,
            "filter": null,
            "id": "vpc-0ed011693d93d59bc",
            "instance_tenancy": "default",
            "ipv6_association_id": "",
            "ipv6_cidr_block": "",
            "main_route_table_id": "rtb-0328c2e3c3d0f1551",
            "owner_id": "404842057573",
            "region": "me-central-1",
            "status": "available"
          }
        }
      ]
    }
  ]
}
```

- **5.6 terraform.tfstate.backup Empty — Screenshot:**

task5_terraform_state_backup_empty.png

```
@SadafRiaz-077 → /workspaces/lab-10 (main) $ cat terraform.tfstate.backup
{
    "version": 4,
    "terraform_version": "1.14.1",
    "serial": 32,
    "lineage": "5f0f5fe9-8b91-d760-48dd-3bfb91f9e713",
    "outputs": {},
    "resources": [],
    "check_results": null
}
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **5.7 terraform state list — Screenshot: task5_terraform_state_list.png**

```
@SadafRiaz-077 → /workspaces/lab-10 (main) $ terraform state list
data.aws_vpc.existing_vpc
aws_subnet.dev_subnet_1
aws_subnet.dev_subnet_1_existing
aws_vpc.development_vpc
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- **5.8 terraform state show — Screenshot: task5_terraform_state_show_resource.png**

```
aws_vpc.development_vpc
@SadafRiaz-077 → /workspaces/lab-10 (main) $ terraform state show aws_vpc.development_vpc
# aws_vpc.development_vpc:
resource "aws_vpc" "development_vpc" {
    arn                               = "arn:aws:ec2:me-central-1:404842057573:vpc/vpc-0ae86db5667fe97df"
    assign_generated_ipv6_cidr_block   = false
    cidr_block                        = "10.0.0.0/16"
    default_network_acl_id           = "acl-03e4e091cf7b6e519"
    default_route_table_id           = "rtb-0270115c1846630f9"
    default_security_group_id        = "sg-0de4380592baff42b"
    dhcp_options_id                  = "dopt-00025b89642cbf410"
    enable_dns_hostnames            = false
    enable_dns_support               = true
    enable_network_address_usage_metrics = false
    id                                = "vpc-0ae86db5667fe97df"
    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-0270115c1846630f9"
    owner_id                          = "404842057573"
    region                            = "me-central-1"
    tags {
        "Name"      = "development"
        "vpc_env"   = "dev"
    }
    tags_all                         = {
        "Name"      = "development"
        "vpc_env"   = "dev"
    }
}
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

Task 6 — Terraform Outputs

- **6.1 Basic Outputs (ID + ARN) — Screenshot: task6_terraform_outputs_basic.png**

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

Outputs:

dev-subnet-arn = "arn:aws:ec2:me-central-1:404842057573:subnet/subnet-0b174b9d309ef2f74"
dev-subnet-id = "subnet-0b174b9d309ef2f74"
dev-vpc-arn = "arn:aws:ec2:me-central-1:404842057573:vpc/vpc-0ae86db5667fe97df"
dev-vpc-id = "vpc-0ae86db5667fe97df"
```

- **6.2 Expanded Outputs (CIDR, Region, Tags) — Screenshot: task6_expanded_outputs.png**

```
@SadafRiaz-077 → /workspaces/lab-10 (main) $ terraform apply -auto-approve
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0ae86db5667fe97df]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0ed011693d93d59bc]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-061ddd60f1301e17b]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0b174b9d309ef2f74]

Changes to Outputs:
+ dev-subnet-cidr_block = "10.0.10.0/24"
+ dev-subnet-region      = "me-central-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         = "me-central-1"
+ dev-vpc-tags_all       = {
    + Name   = "development"
    + vpc_env = "dev"
}
+ dev-vpc-tags_name      = "development"

You can apply this plan to save these new output values to the Terraform state, without changing any resources.

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

Outputs:

dev-subnet-arn = "arn:aws:ec2:me-central-1:404842057573:subnet/subnet-0b174b9d309ef2f74"
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-id = "subnet-0b174b9d309ef2f74"
dev-subnet-region = "me-central-1a"
dev-subnet-tags_all = tomap({
    "Name" = "subnet-1-dev"
})
dev-subnet-tags_name = "subnet-1-dev"
dev-vpc-arn = "arn:aws:ec2:me-central-1:404842057573:vpc/vpc-0ae86db5667fe97df"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-id = "vpc-0ae86db5667fe97df"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
    "Name" = "development"
    "vpc_env" = "dev"
})
dev-vpc-tags_name = "development"
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

Cleanup — Delete Resources & State Verification

- Destroy All Resources. Save screenshot as: cleanup_destroy_resources.png — destroy output:

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

Changes to Outputs:
- dev-subnet-arn      = "arn:aws:ec2:me-central-1:404842057573:subnet/
- dev-subnet-cidr_block = "10.0.10.0/24" -> null
- dev-subnet-id        = "subnet-0b174b9d309ef2f74" -> null
- dev-subnet-region    = "me-central-1a" -> null
- dev-subnet-tags_all  = {
    - Name = "subnet-1-dev"
} -> null
- dev-subnet-tags_name = "subnet-1-dev" -> null
- dev-vpc-arn          = "arn:aws:ec2:me-central-1:404842057573:vpc/vpc
- dev-vpc-cidr_block   = "10.0.0.0/16" -> null
- dev-vpc-id           = "vpc-0ae86db5667fe97df" -> null
- dev-vpc-region        = "me-central-1" -> null
- dev-vpc-tags_all     = {
    - Name      = "development"
    - vpc_env   = "dev"
} -> null
- dev-vpc-tags_name    = "development" -> null

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-061ddd60f1301e1]
aws_subnet.dev_subnet_1: Destroying... [id=subnet-0b174b9d309ef2f74]
aws_subnet.dev_subnet_1_existing: Destruction complete after 0s
aws_subnet.dev_subnet_1: Destruction complete after 0s
aws_vpc.development_vpc: Destroying... [id=vpc-0ae86db5667fe97df]
aws_vpc.development_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
@SadafRiaz-077 → /workspaces/lab-10 (main) $
```

- Inspect State Files. Save screenshot as: cleanup_state_files.png

```
@SadafRiaz-077 → /workspaces/lab-10 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 44,
  "lineage": "5f0f5fe9-8b91-d760-48dd-3bfb91f9e713",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@SadafRiaz-077 → /workspaces/lab-10 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.1",
  "serial": 39,
  "lineage": "5f0f5fe9-8b91-d760-48dd-3bfb91f9e713",
  "outputs": {
    "dev-subnet-arn": {
      "value": "arn:aws:ec2:me-central-1:404842057573:subnet/subnet-0b174b9d309ef2f74",
      "type": "string"
    },
    "dev-subnet-cidr_block": {
      "value": "10.0.10.0/24",
      "type": "string"
    },
    "dev-subnet-id": {
      "value": "subnet-0b174b9d309ef2f74",
      "type": "string"
    },
    "dev-subnet-region": {
      "value": "me-central-1a",
      "type": "string"
    },
    "dev-subnet-tags_all": {
      "value": {
        "Name": "subnet-1-dev"
      }
    }
  }
}
```