Fatima Jinnah
Women University

Lab 10

Lab title: GH CLI Codespaces + AWS + Terraform: CLI Automation of VPC/Subnet Creation

Submitted to: Engr. Mohammad Shoaib

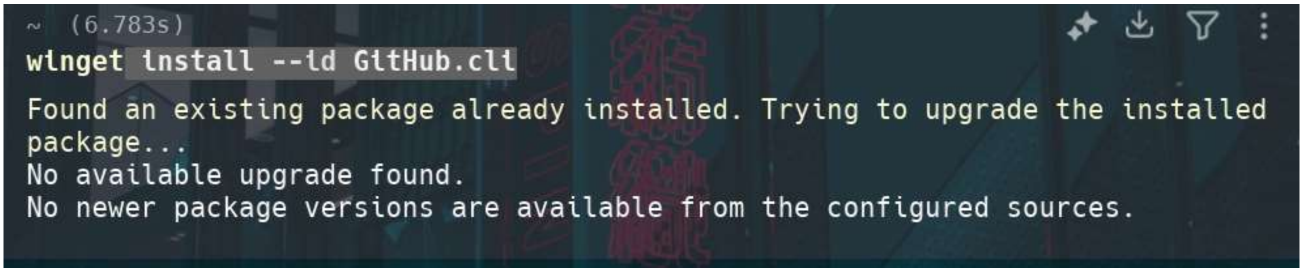
Submitted by: Anara Hayat

Reg#No: 2023-BSE-008

Task 1 — GitHub CLI Codespace Setup & Authentication

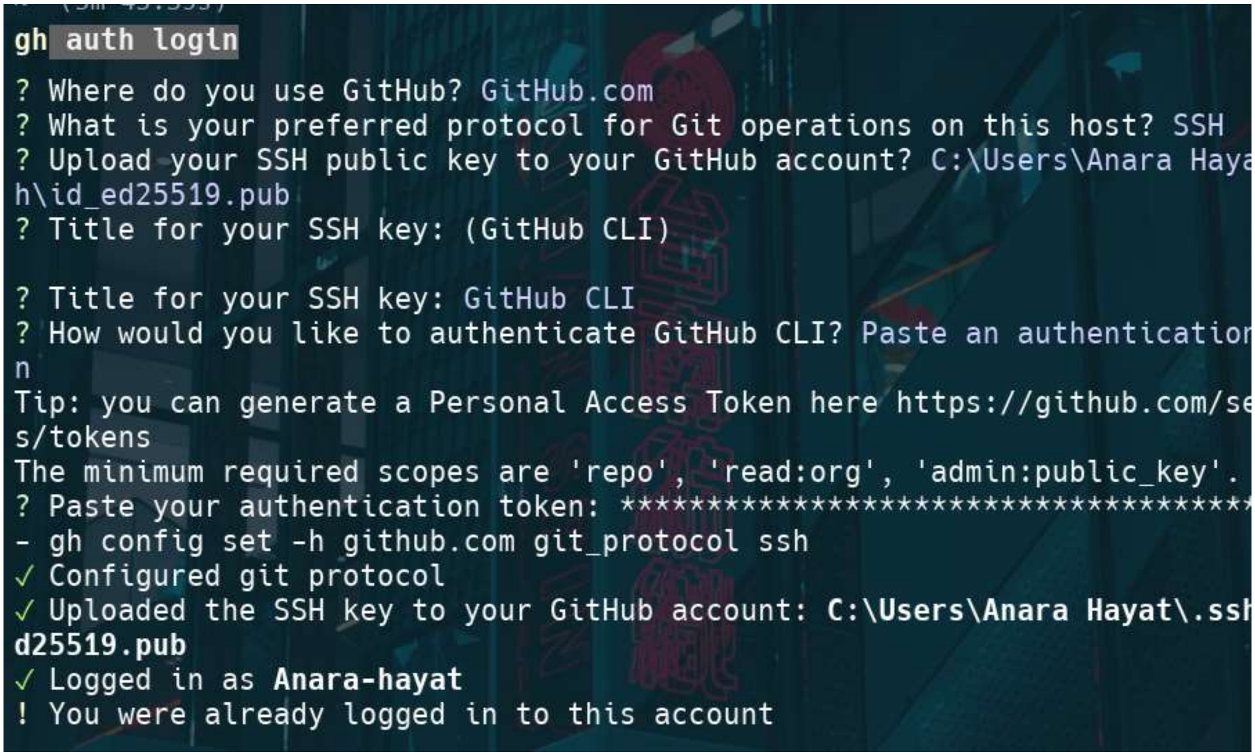
Task 1 — GitHub CLI Codespace Setup & Authentication

1. **Install GitHub CLI:**



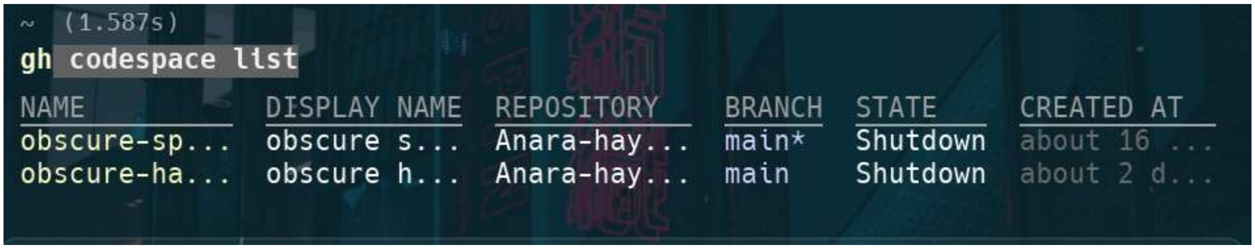
```
~ (6.783s)
winget install --id GitHub.cli
Found an existing package already installed. Trying to upgrade the installed
package...
No available upgrade found.
No newer package versions are available from the configured sources.
```

2. **Authenticate GH CLI for Codespaces:**



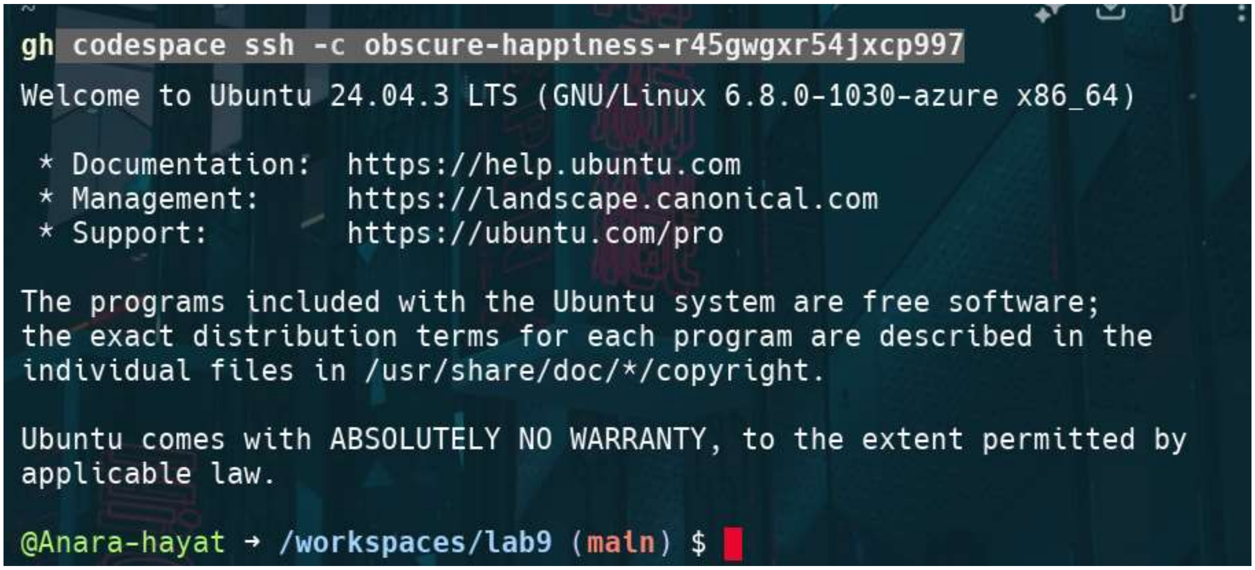
```
gh auth login
? Where do you use GitHub? GitHub.com
? What is your preferred protocol for Git operations on this host? SSH
? Upload your SSH public key to your GitHub account? C:\Users\Anara Haya
h\id_ed25519.pub
? Title for your SSH key: (GitHub CLI)

? Title for your SSH key: GitHub CLI
? How would you like to authenticate GitHub CLI? Paste an authenticatior
n
Tip: you can generate a Personal Access Token here https://github.com/se
s/tokens
The minimum required scopes are 'repo', 'read:org', 'admin:public_key'.
? Paste your authentication token: ************************************
- gh config set -h github.com git_protocol ssh
✓ Configured git protocol
✓ Uploaded the SSH key to your GitHub account: C:\Users\Anara Hayat\.ssh
d25519.pub
✓ Logged in as Anara-hayat
! You were already logged in to this account
```

3. **List available codespaces:**



```
~ (1.587s)
gh codespace list
NAME           DISPLAY NAME  REPOSITORY    BRANCH  STATE     CREATED AT
obscure-sp...  obscure s...  Anara-hay...  main*   Shutdown  about 16 ...
obscure-ha...  obscure h...  Anara-hay...  main    Shutdown  about 2 d...
```

4. **Connect to a codespace via SSH:**

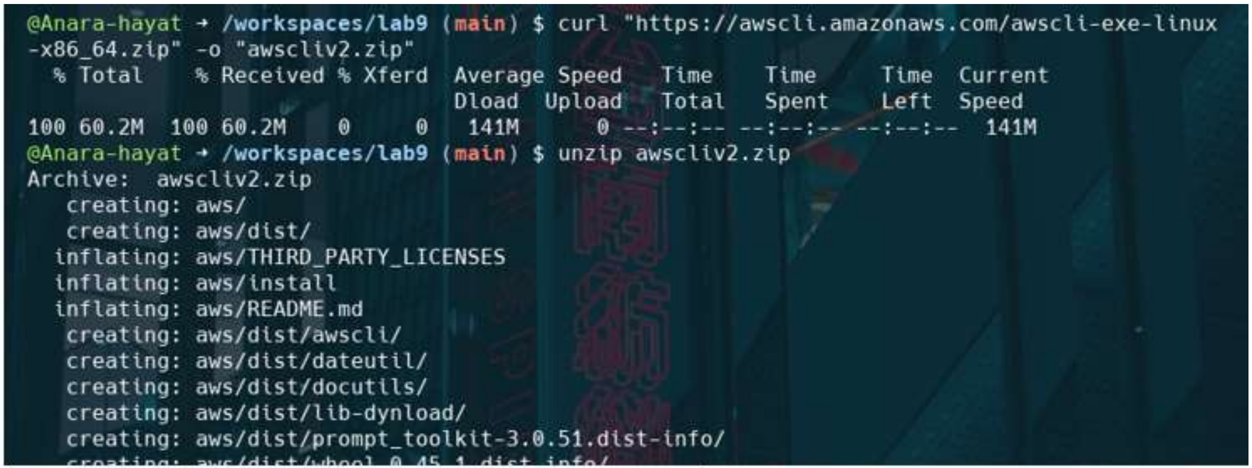


```
gh codespace ssh -c obscure-happiness-r45gwgxr54jxcp997
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-1030-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

@Anara-hayat → /workspaces/lab9 (main) $
```

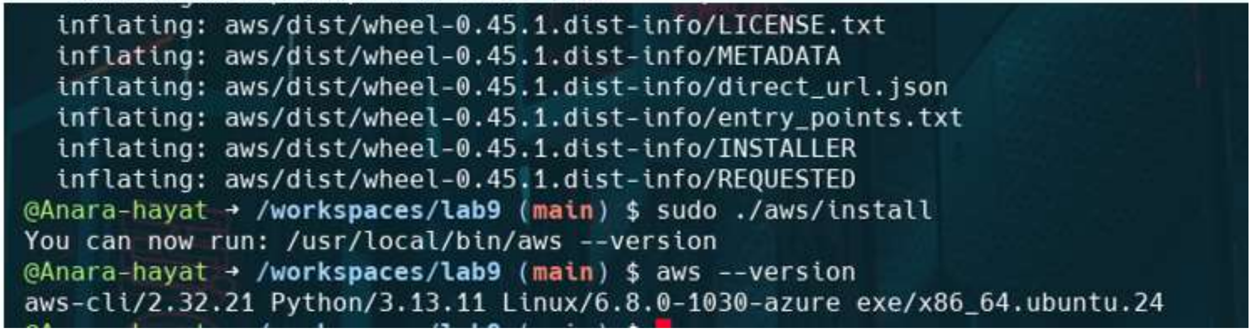
Task 2 — Install AWS CLI, Terraform CLI, Provider Setup

A. Install AWS CLI (Skip if already installed)

1. **In Codespace shell, install AWS CLI:**

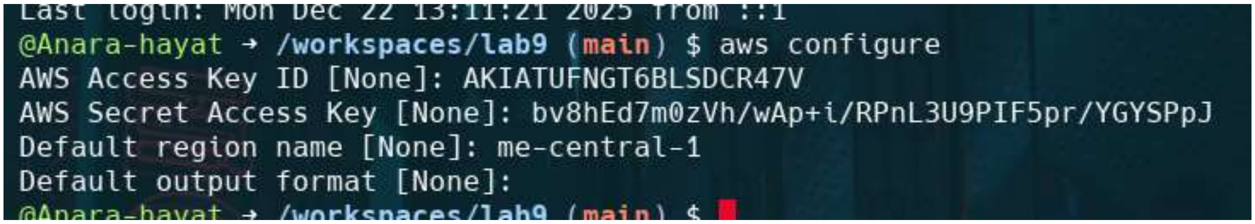


```
@Anara-hayat → /workspaces/lab9 (main) $ curl "https://awscli.amazonaws.com/awscli-exe-linux
-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 60.2M  100 60.2M    0     0   141M      0 --:--:-- --:--:-- --:--:--  141M
@Anara-hayat → /workspaces/lab9 (main) $ unzip awscliv2.zip
Archive:  awscliv2.zip
   creating: aws/
   creating: aws/dist/
  inflating: aws/THIRD_PARTY_LICENSES
  inflating: aws/install
  inflating: aws/README.md
   creating: aws/dist/awscli/
   creating: aws/dist/dateutil/
   creating: aws/dist/docutils/
   creating: aws/dist/lib-dynload/
   creating: aws/dist/prompt_toolkit-3.0.51.dist-info/
   creating: aws/dist/wheel-0.45.1.dist-info/
```

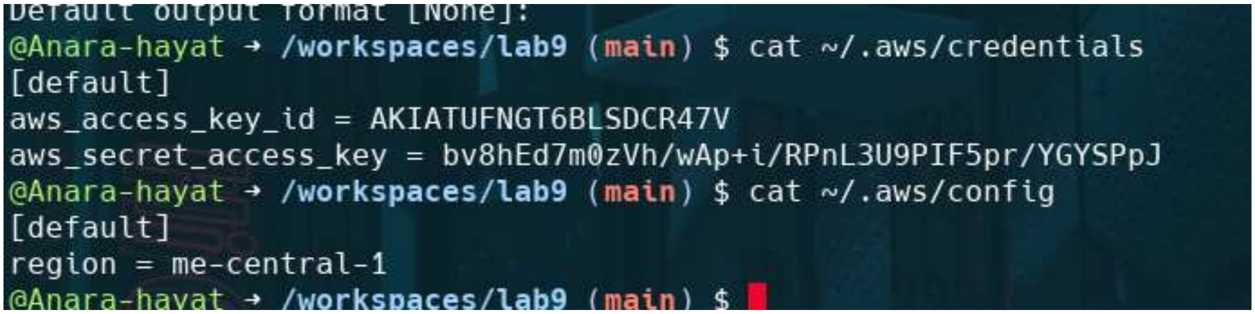


```
  inflating: aws/dist/wheel-0.45.1.dist-info/LICENSE.txt
  inflating: aws/dist/wheel-0.45.1.dist-info/METADATA
  inflating: aws/dist/wheel-0.45.1.dist-info/direct_url.json
  inflating: aws/dist/wheel-0.45.1.dist-info/entry_points.txt
  inflating: aws/dist/wheel-0.45.1.dist-info/INSTALLER
  inflating: aws/dist/wheel-0.45.1.dist-info/REQUESTED
@Anara-hayat → /workspaces/lab9 (main) $ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
@Anara-hayat → /workspaces/lab9 (main) $ aws --version
aws-cli/2.32.21 Python/3.13.11 Linux/6.8.0-1030-azure exe/x86_64.ubuntu.24
```

2. **Configure AWS CLI:**

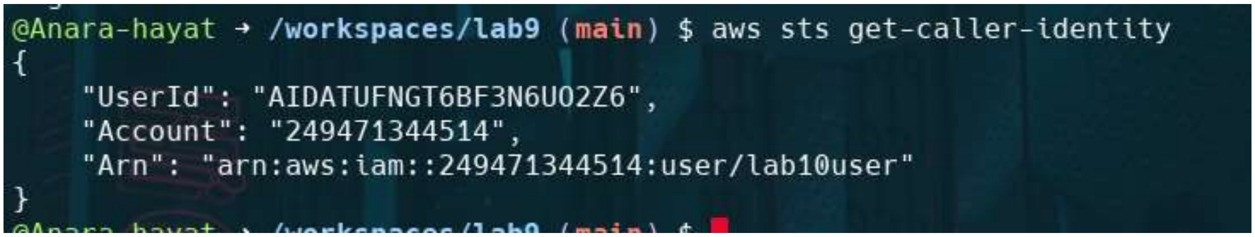


```
Last login: Mon Dec 22 13:11:21 2025 from ::1
@Anara-hayat → /workspaces/lab9 (main) $ aws configure
AWS Access Key ID [None]: AKIATUFNGT6BLSDCR47V
AWS Secret Access Key [None]: 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
Default region name [None]: me-central-1
Default output format [None]:
@Anara-hayat → /workspaces/lab9 (main) $
```



```
Default output format [None]:
@Anara-hayat → /workspaces/lab9 (main) $ cat ~/.aws/credentials
[default]
aws_access_key_id = AKIATUFNGT6BLSDCR47V
aws_secret_access_key = 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
@Anara-hayat → /workspaces/lab9 (main) $ cat ~/.aws/config
[default]
region = me-central-1
@Anara-hayat → /workspaces/lab9 (main) $
```

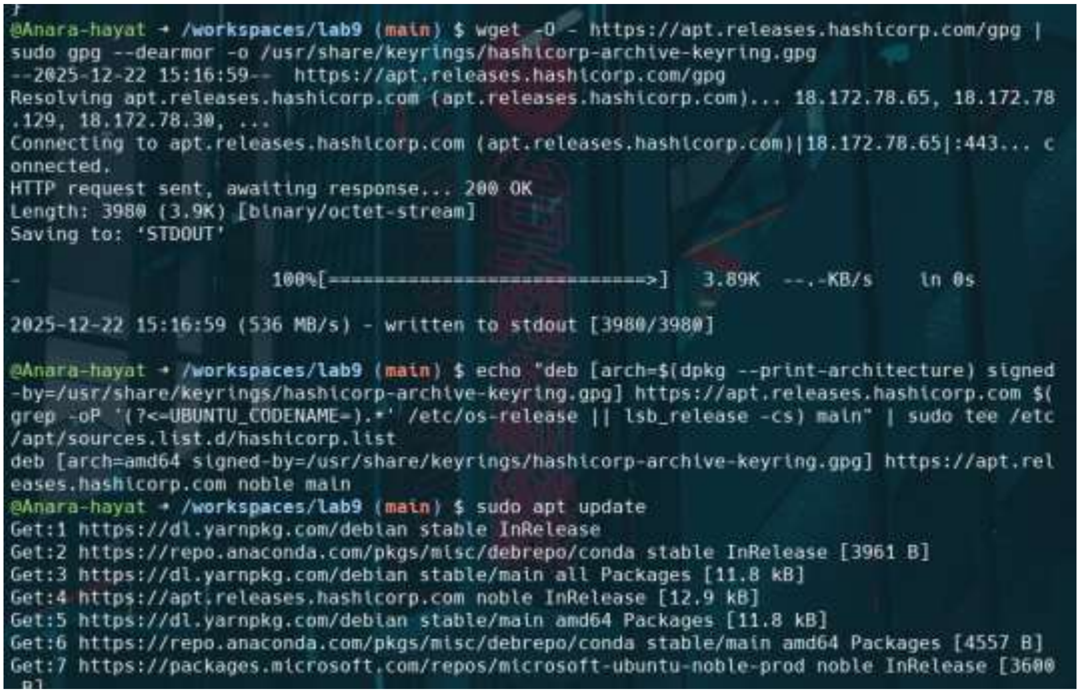
3. **Verify AWS CLI connectivity:**



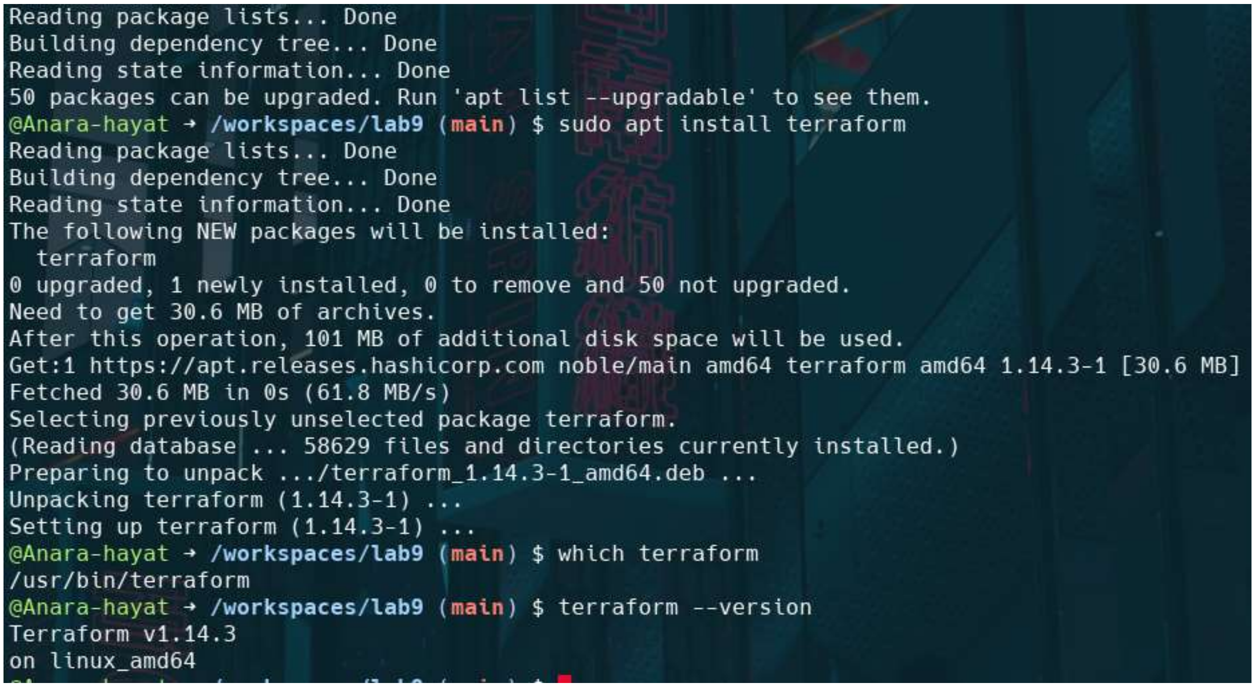
```
@Anara-hayat → /workspaces/lab9 (main) $ aws sts get-caller-identity
{
    "UserId": "AIDATUFNGT6BF3N6U02Z6",
    "Account": "249471344514",
    "Arn": "arn:aws:iam::249471344514:user/lab10user"
}
@Anara-hayat → /workspaces/lab9 (main) $
```

B. Install Terraform CLI

1. **Install Terraform CLI:**



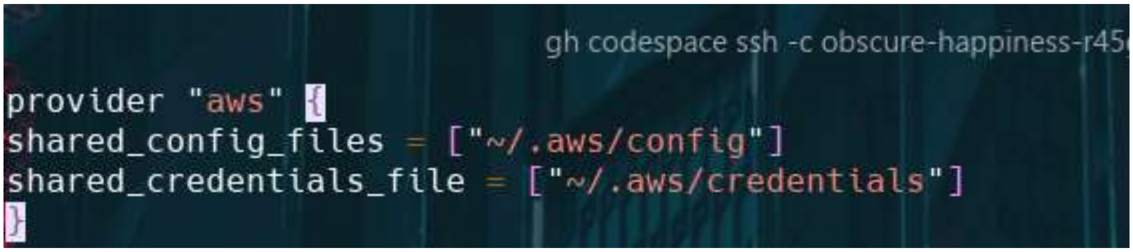
```
@Anara-hayat → /workspaces/lab9 (main) $ wget -O - https://apt.releases.hashicorp.com/gpg |
sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
--2025-12-22 15:16:59--  https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 18.172.78.65, 18.172.78
.129, 18.172.78.30, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|18.172.78.65|:443... c
onnected.
HTTP request sent, awaiting response... 200 OK
Length: 3980 (3.9K) [binary/octet-stream]
Saving to: 'STDOUT'

-                       100%[=============================>]   3.89K  --.-KB/s    in 0s

2025-12-22 15:16:59 (536 MB/s) - written to stdout [3980/3980]

@Anara-hayat → /workspaces/lab9 (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.rel
eases.hashicorp.com noble main
@Anara-hayat → /workspaces/lab9 (main) $ sudo apt update
Get:1 https://dl.yarnpkg.com/debian stable InRelease
Get:2 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable InRelease [3961 B]
Get:3 https://dl.yarnpkg.com/debian stable/main all Packages [11.8 kB]
Get:4 https://apt.releases.hashicorp.com noble InRelease [12.9 kB]
Get:5 https://dl.yarnpkg.com/debian stable/main amd64 Packages [11.8 kB]
Get:6 https://repo.anaconda.com/pkgs/misc/debrepo/conda stable/main amd64 Packages [4557 B]
Get:7 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease [3600
 B]
```



```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
50 packages can be upgraded. Run 'apt list --upgradable' to see them.
@Anara-hayat ➜ /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 50 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 (61.8 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) ...
@Anara-hayat ➜ /workspaces/lab9 (main) $ which terraform
/usr/bin/terraform
@Anara-hayat ➜ /workspaces/lab9 (main) $ terraform --version
Terraform v1.14.3
on linux_amd64
```

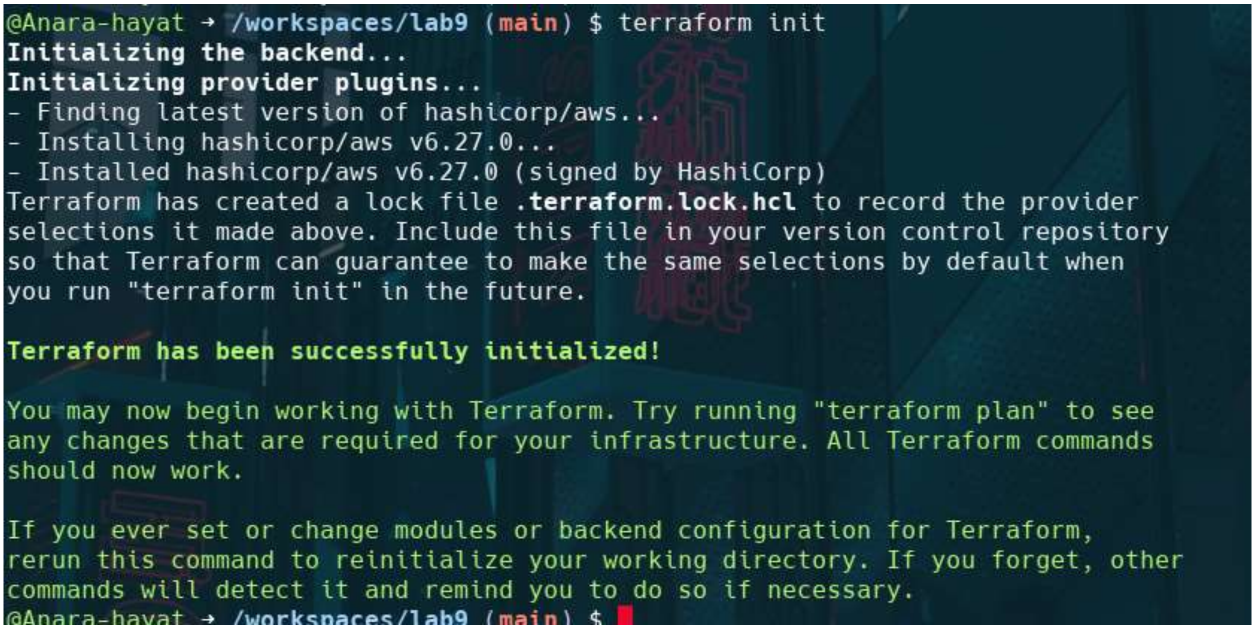
C. Provider Configuration (main.tf)

- **Create main.tf using Vim**



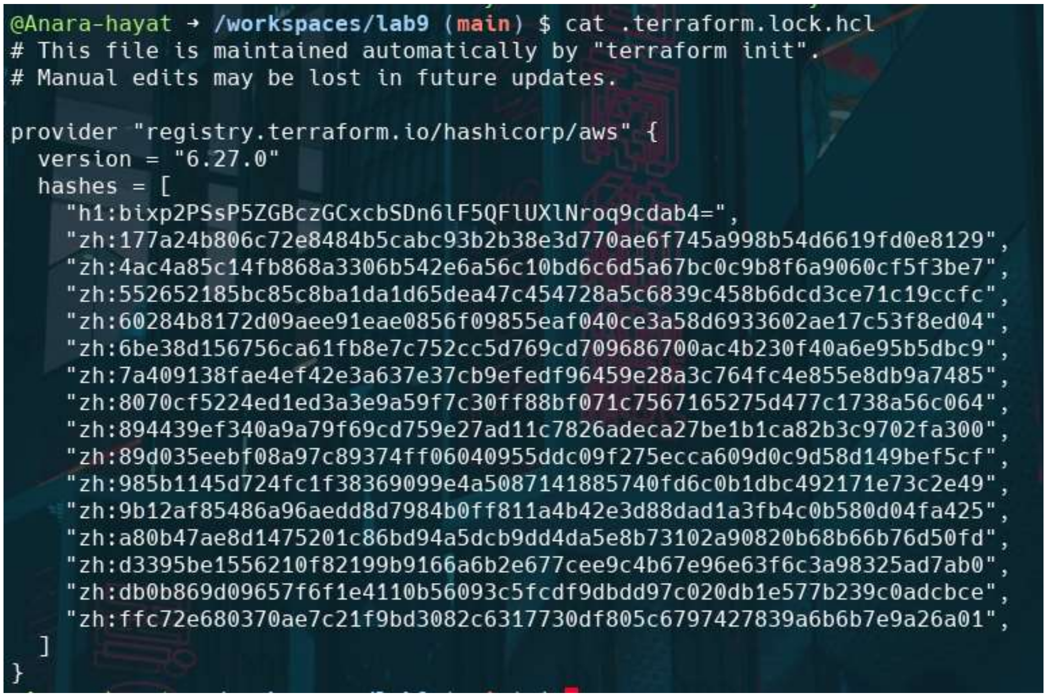
```
gh codespace ssh -c obscure-happiness-r45

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

Initialize Terraform:



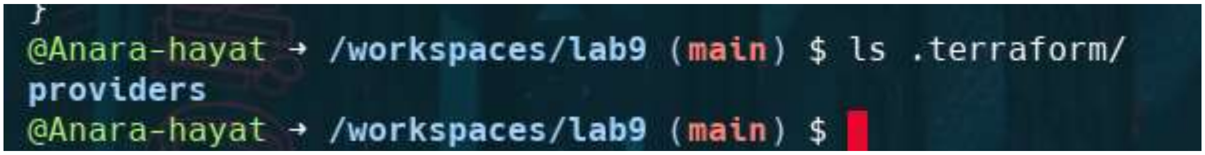
```
@Anara-hayat → /workspaces/lab9 (main) $ terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v6.27.0...
- Installed hashicorp/aws v6.27.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl 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.
@Anara-hayat → /workspaces/lab9 (main) $
```

Show contents of .terraform.lock.hcl:



```
@Anara-hayat → /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",
  ]
}
```

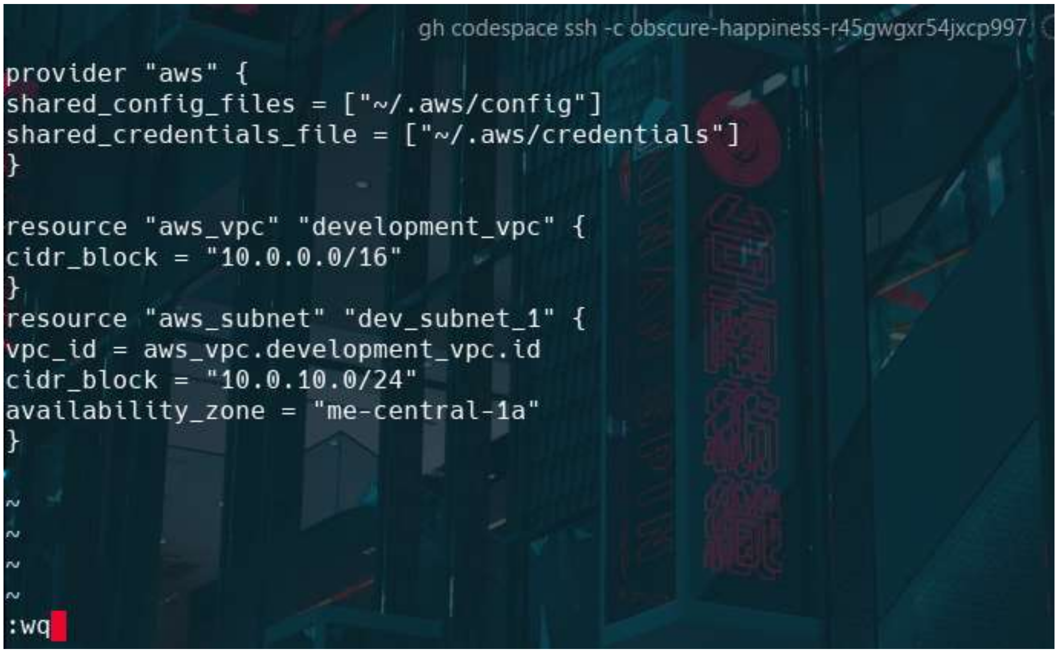
Show contents of .terraform/ directory:



```
}
@Anara-hayat → /workspaces/lab9 (main) $ ls .terraform/
providers
@Anara-hayat → /workspaces/lab9 (main) $
```

Task 3 — VPC/Subnet Creation, Initialization, Verification

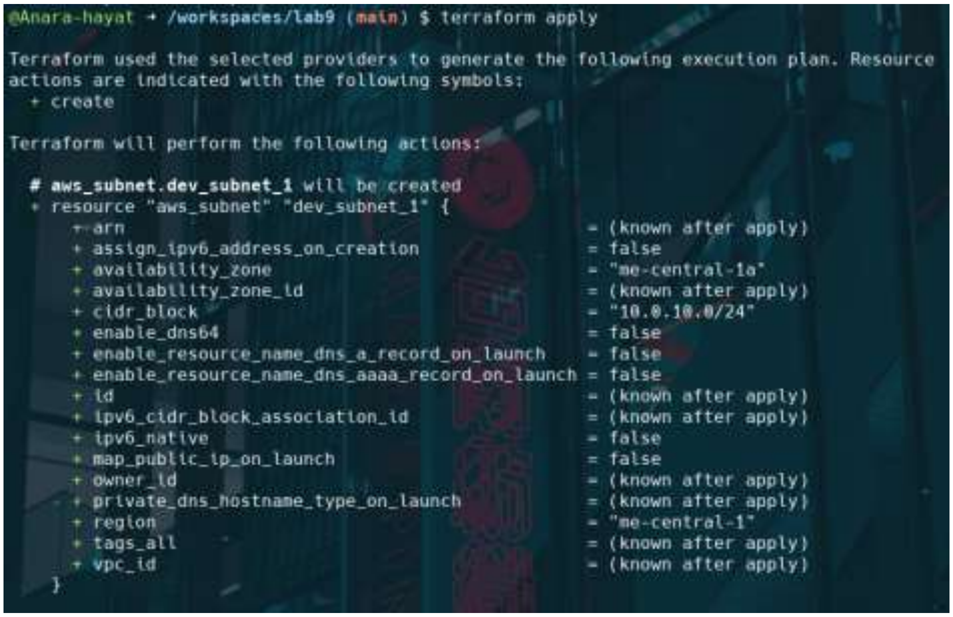
1. **Edit main.tf to add:**



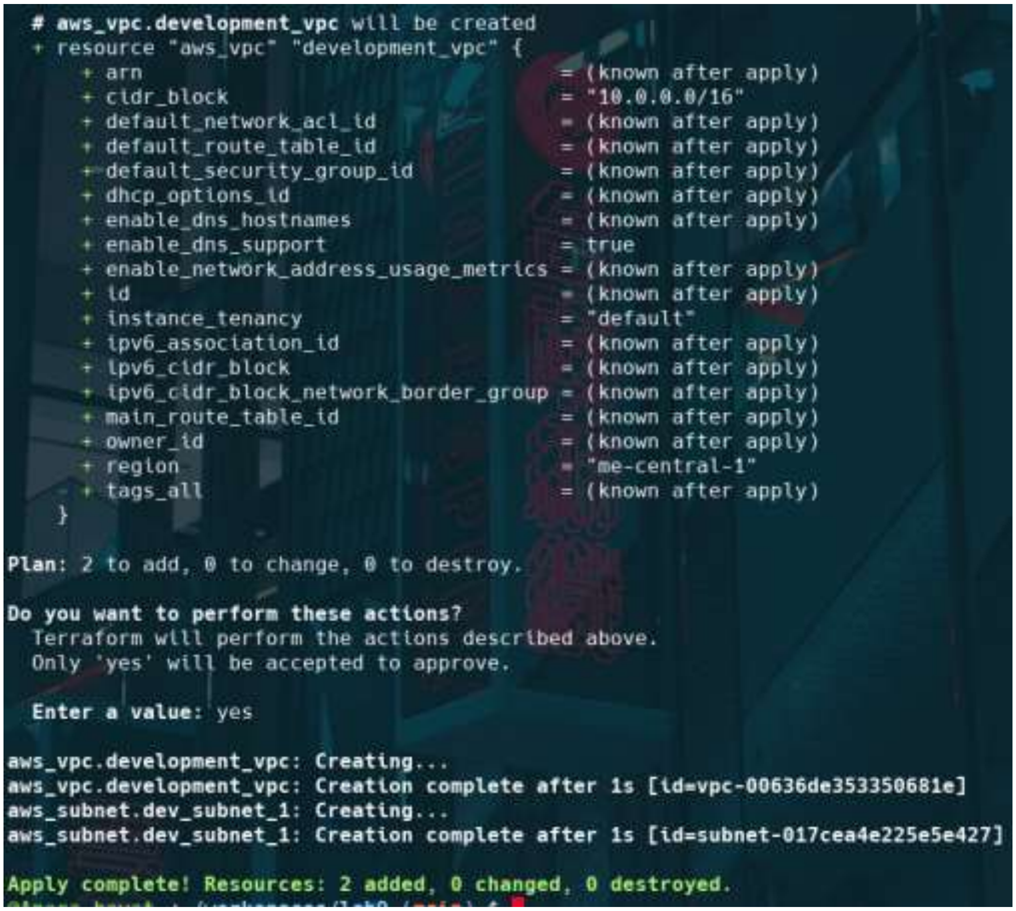
gh codespace ssh -c obscure-happiness-r45gwgxr54jxcp997

```
provider "aws" {
shared_config_files = ["~/.aws/config"]
shared_credentials_file = ["~/.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"
}

~
~
~
~
:wq
```

2. **Run:**

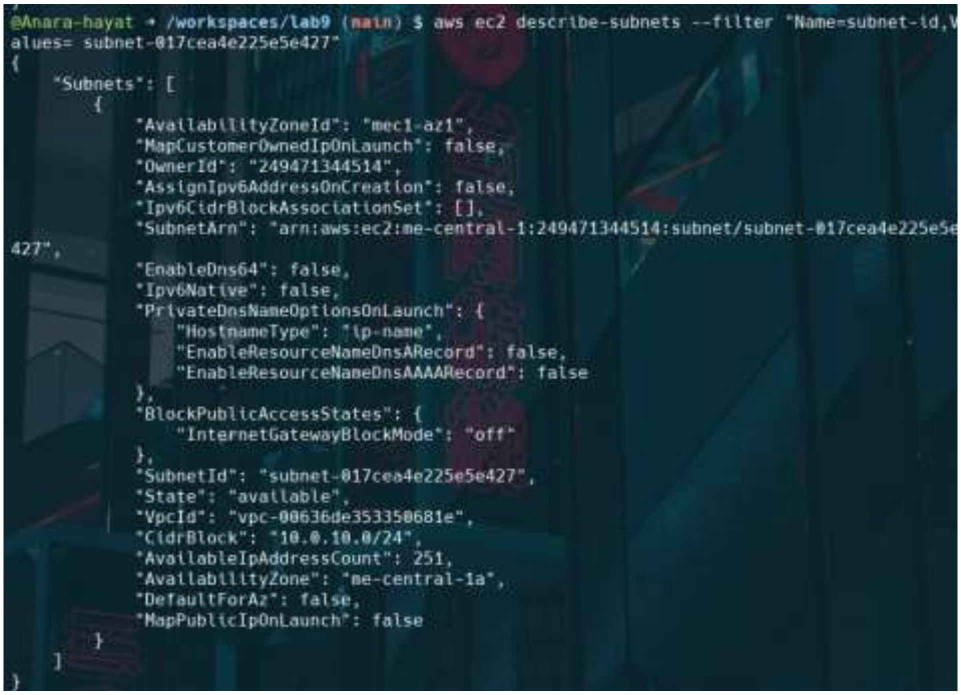


```
@Anara-hayat → /workspaces/lab9 (main) $ terraform apply

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)
    }
```



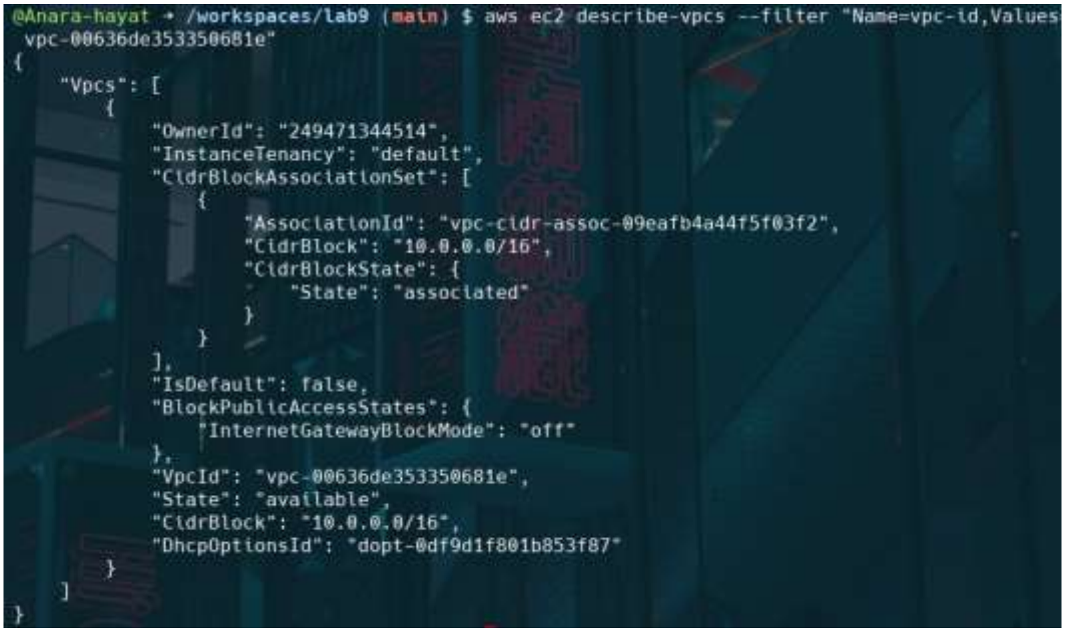
```
  # aws_vpc.development_vpc will be created
  + resource "aws_vpc" "development_vpc" {
      + arn                                  = (known after apply)
      + cidr_block                           = "10.0.0.0/16"
      + default_network_acl_id               = (known after apply)
      + default_route_table_id               = (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-00636de353350681e]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-017cea4e225e5e427]

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

3. **Verify resources using AWS CLI:**



```
@Anara-hayat → /workspaces/lab9 (main) $ aws ec2 describe-subnets --filter "Name=subnet-id,V
alues= subnet-017cea4e225e5e427"
{
    "Subnets": [
        {
            "AvailabilityZoneId": "mec1-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "249471344514",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [],
            "SubnetArn": "arn:aws:ec2:me-central-1:249471344514:subnet/subnet-017cea4e225e5e
427",
            "EnableDns64": false,
            "Ipv6Native": false,
            "PrivateDnsNameOptionsOnLaunch": {
                "HostnameType": "ip-name",
                "EnableResourceNameDnsARecord": false,
                "EnableResourceNameDnsAAAARecord": false
            },
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "SubnetId": "subnet-017cea4e225e5e427",
            "State": "available",
            "VpcId": "vpc-00636de353350681e",
            "CidrBlock": "10.0.10.0/24",
            "AvailableIpAddressCount": 251,
            "AvailabilityZone": "me-central-1a",
            "DefaultForAz": false,
            "MapPublicIpOnLaunch": false
        }
    ]
}
```

Run:

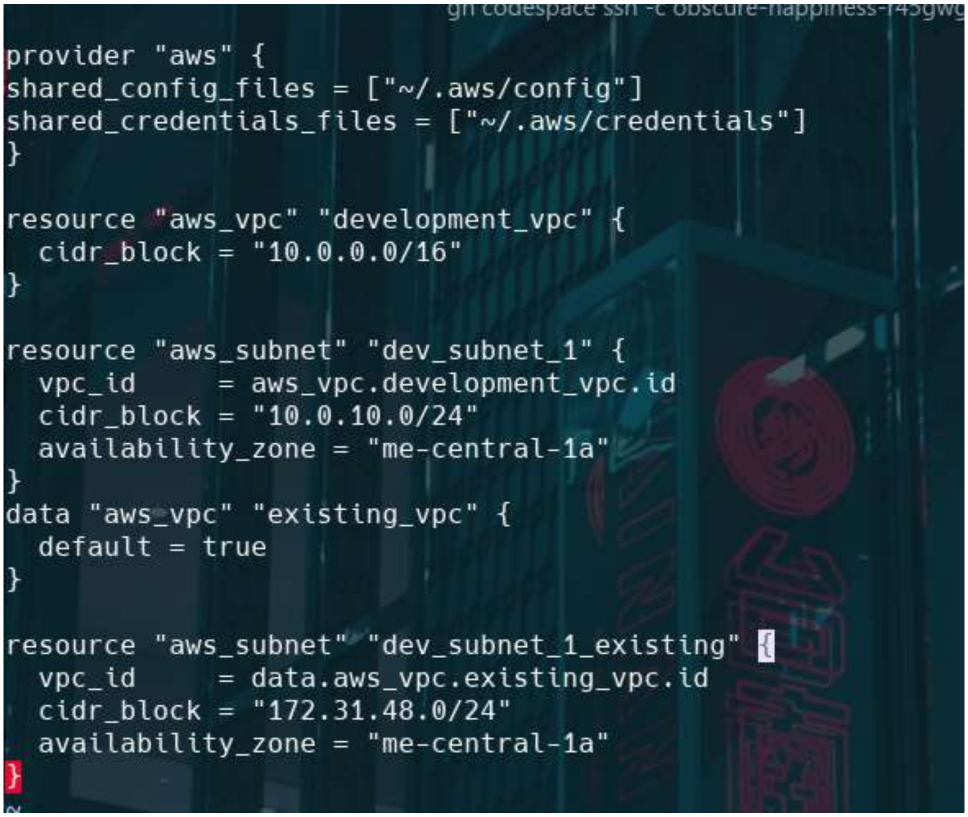


```
@Anara-hayat → /workspaces/lab9 (main) $ aws ec2 describe-vpcs --filter "Name=vpc-id,Values
 vpc-00636de353350681e"
{
    "Vpcs": [
        {
            "OwnerId": "249471344514",
            "InstanceTenancy": "default",
            "CidrBlockAssociationSet": [
                {
                    "AssociationId": "vpc-cidr-assoc-09eafb4a44f5f03f2",
                    "CidrBlock": "10.0.0.0/16",
                    "CidrBlockState": {
                        "State": "associated"
                    }
                }
            ],
            "IsDefault": false,
            "BlockPublicAccessStates": {
                "InternetGatewayBlockMode": "off"
            },
            "VpcId": "vpc-00636de353350681e",
            "State": "available",
            "CidrBlock": "10.0.0.0/16",
            "DhcpOptionsId": "dopt-0df9d1f801b853f87"
        }
    ]
}
```

Task 4 — Data Source, Targeted Destroy, Tags

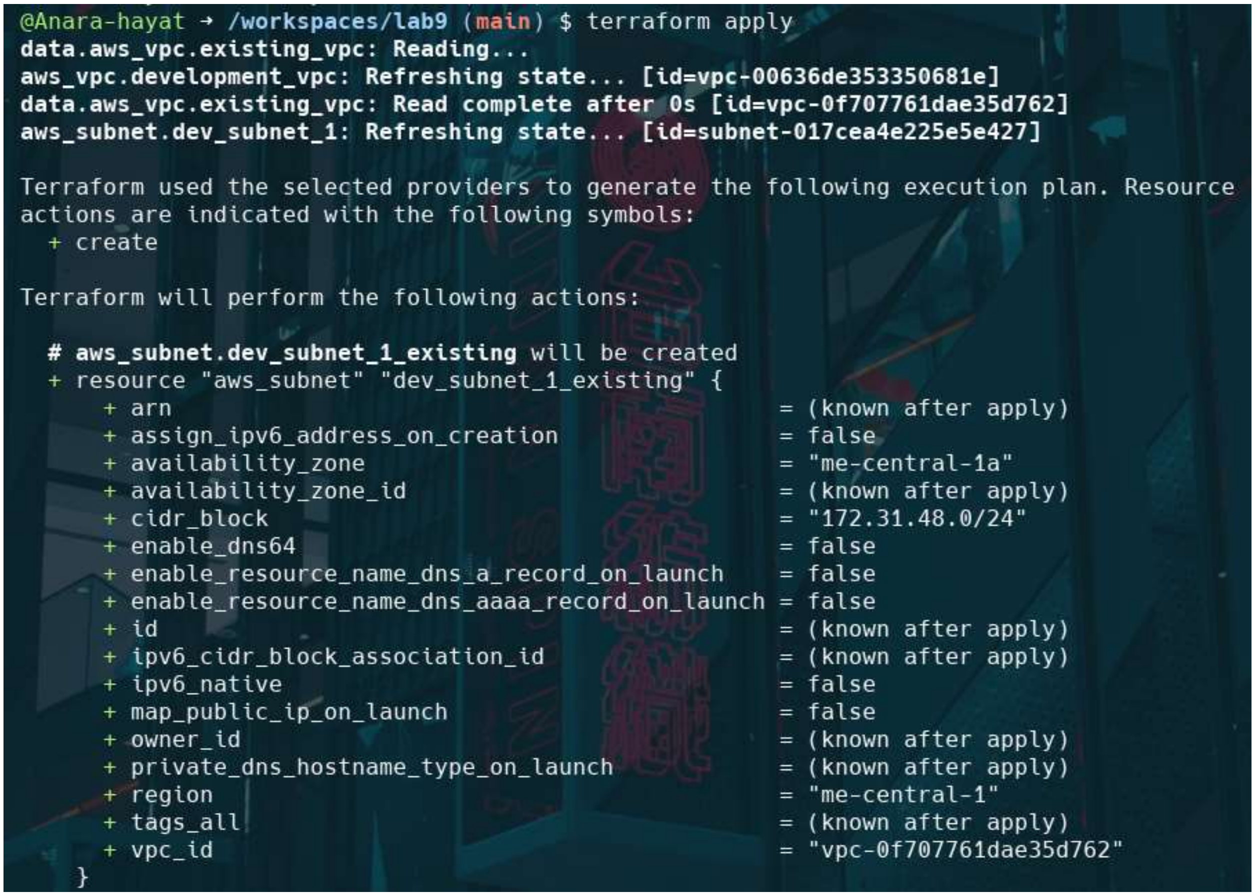
A. Data Source & Resource Creation

1. **Add to main.tf:**

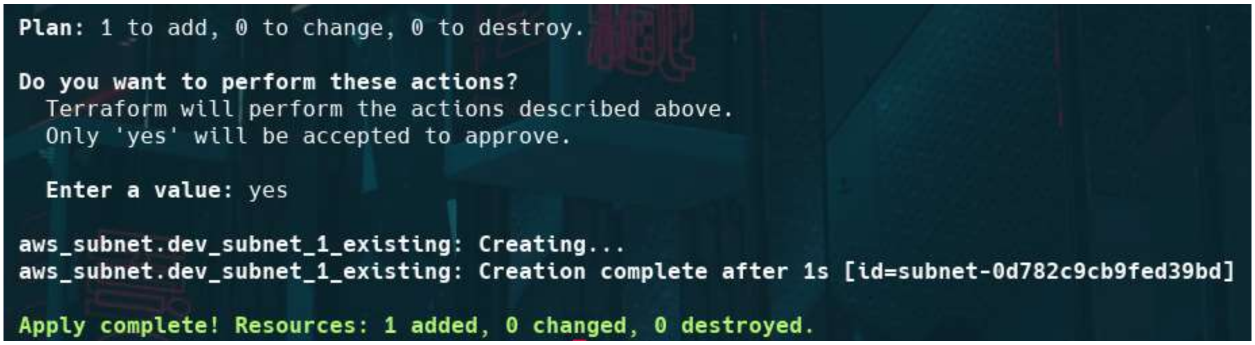


```
gh codespace ssh -c obscure-happiness-r45gwg

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"
}
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 = "me-central-1a"
}
```

2. Apply configuration:



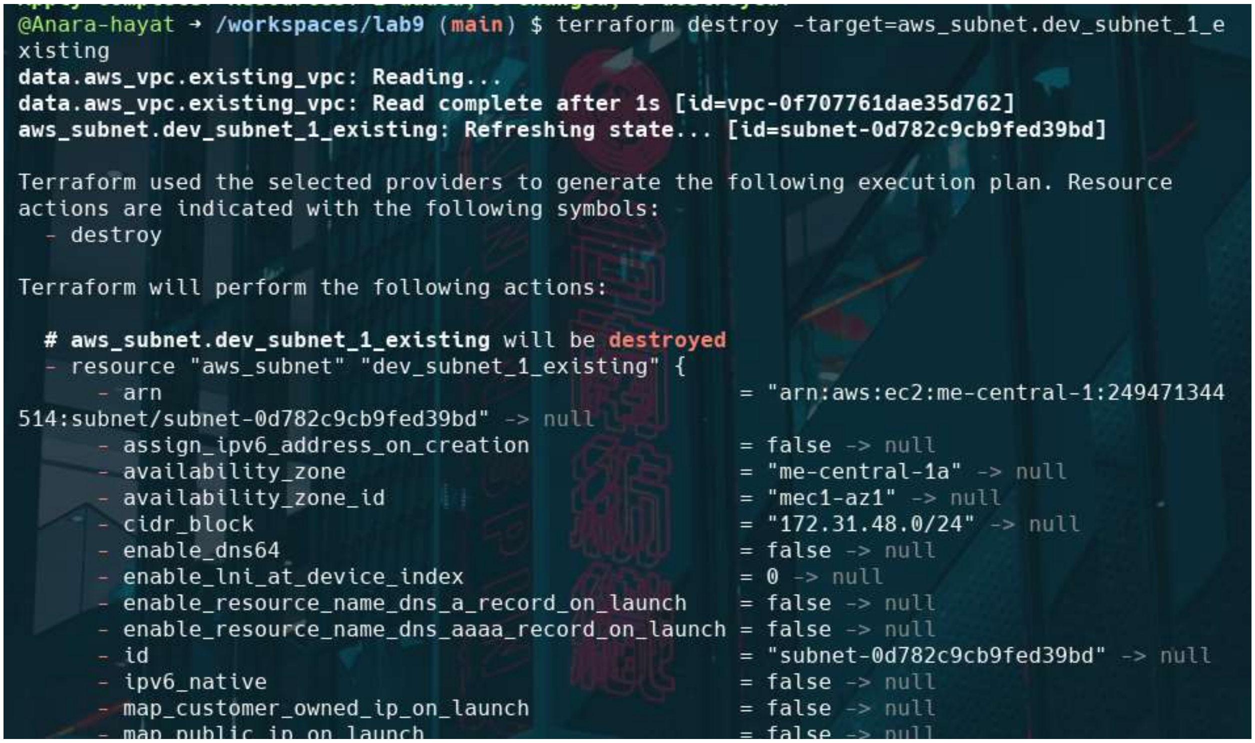
```
@Anara-hayat → /workspaces/lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00636de353350681e]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-017cea4e225e5e427]

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                              = "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_all                                       = (known after apply)
      + vpc_id                                         = "vpc-0f707761dae35d762"
    }
```



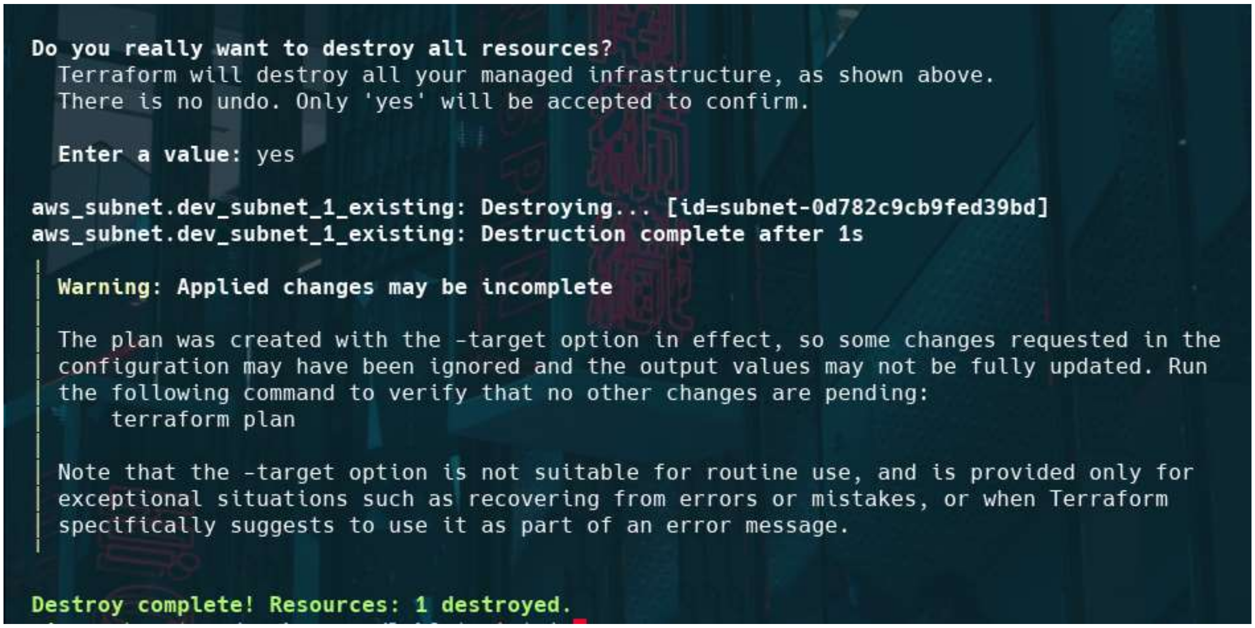
```
Plan: 1 to add, 0 to change, 0 to destroy.

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-0d782c9cb9fed39bd]

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

B. Targeted Destroy & Refresh

1. Destroy only one resource:

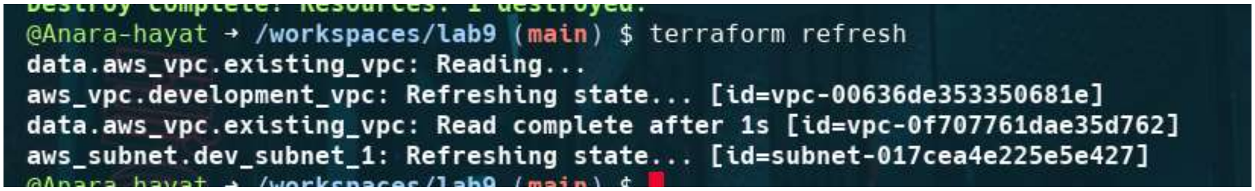


```
@Anara-hayat → /workspaces/lab9 (main) $ terraform destroy -target=aws_subnet.dev_subnet_1_e
xisting
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0d782c9cb9fed39bd]

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:me-central-1:249471344
514:subnet/subnet-0d782c9cb9fed39bd" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "me-central-1a" -> null
      - availability_zone_id                           = "mec1-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-0d782c9cb9fed39bd" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
```



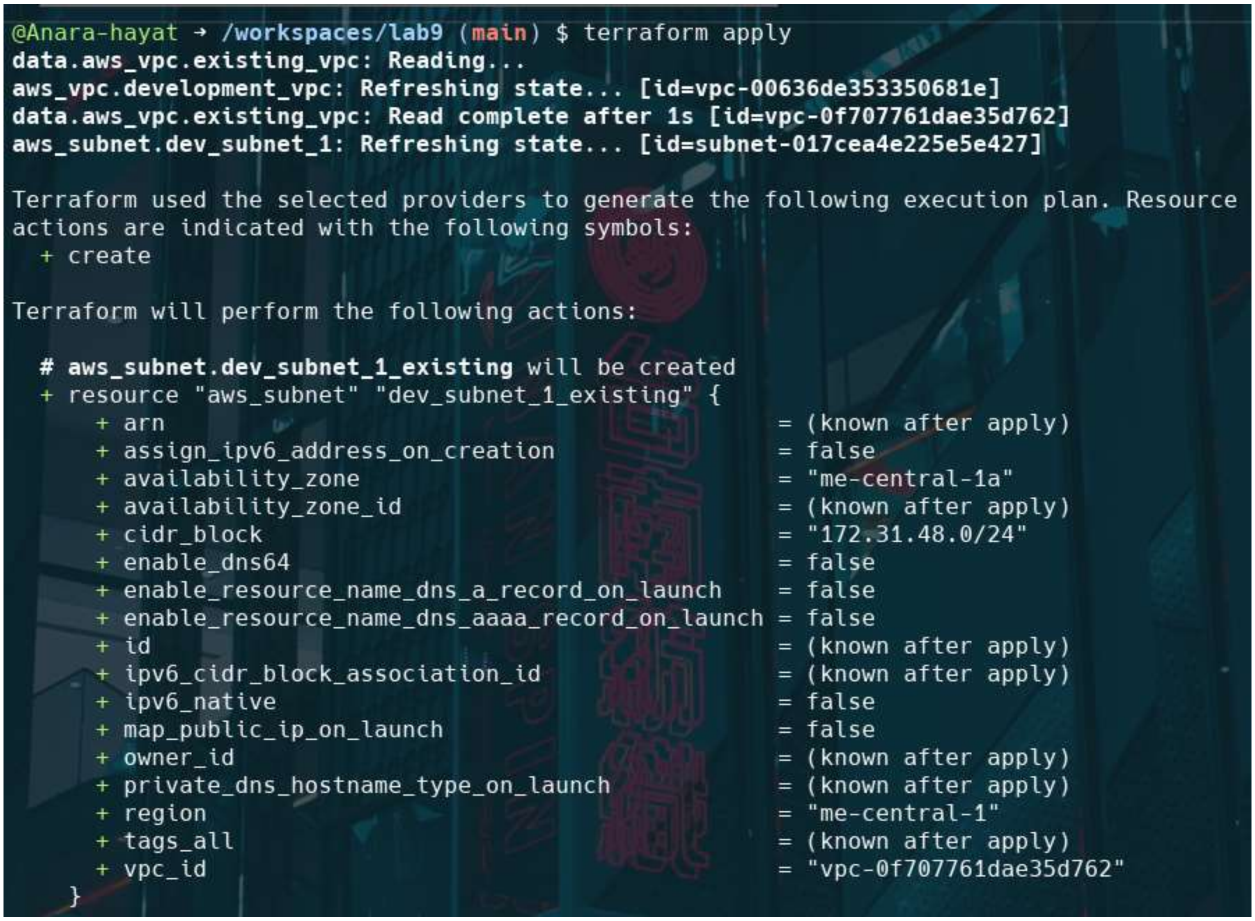
```
 Do you really want to destroy all resources?
   Terraform will destroy all your managed infrastructure, as shown above.
   There is no undo. Only 'yes' will be accepted to confirm.

   Enter a value: yes

 aws_subnet.dev_subnet_1_existing: Destroying... [id=subnet-0d782c9cb9fed39bd]
 aws_subnet.dev_subnet_1_existing: Destruction complete after 1s
│
│ Warning: Applied changes may be incomplete
│
│ The plan was created with the -target option in effect, so some changes requested in the
│ configuration may have been ignored and the output values may not be fully updated. Run
│ the following command to verify that no other changes are pending:
│     terraform plan
│
│ Note that the -target option is not suitable for routine use, and is provided only for
│ exceptional situations such as recovering from errors or mistakes, or when Terraform
│ specifically suggests to use it as part of an error message.
╵

 Destroy complete! Resources: 1 destroyed.
```

2. Refresh state:

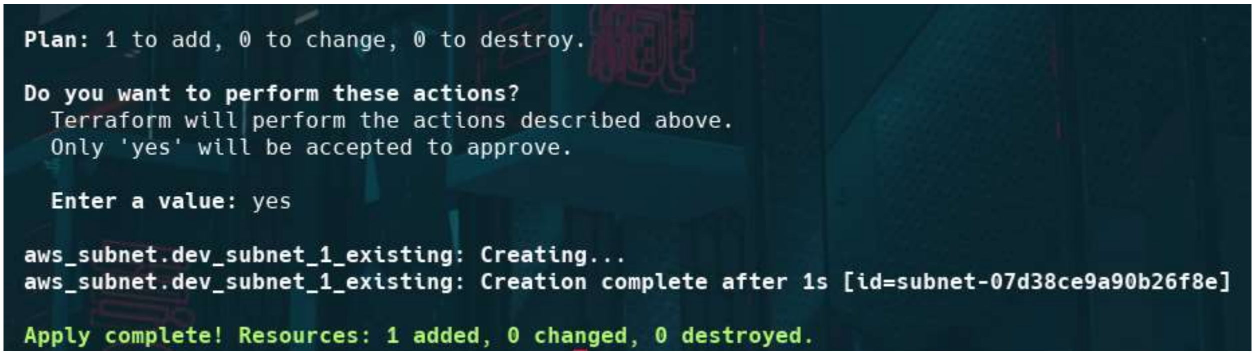


```
Destroy complete! Resources: 1 destroyed.
@Anara-hayat → /workspaces/lab9 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00636de353350681e]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-017cea4e225e5e427]
@Anara-hayat → /workspaces/lab9 (main) $
```

3. Re-apply configuration:

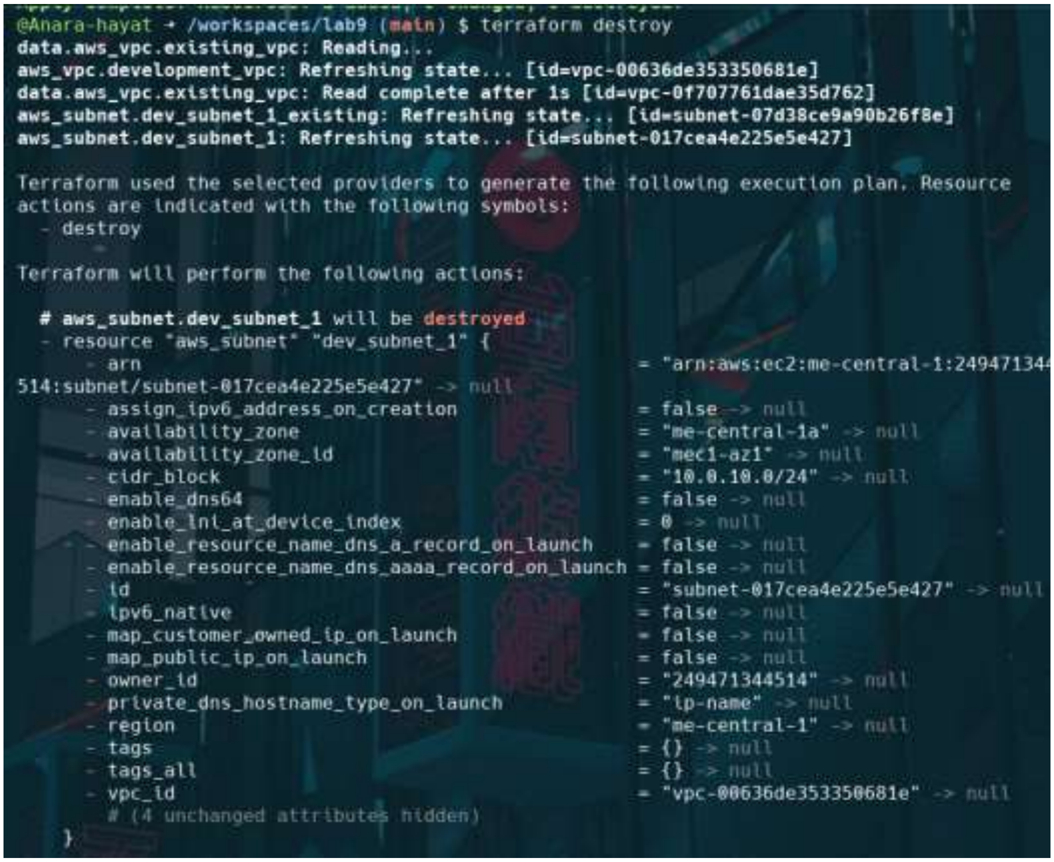


```
@Anara-hayat → /workspaces/lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00636de353350681e]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-017cea4e225e5e427]

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                              = "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_all                                       = (known after apply)
      + vpc_id                                         = "vpc-0f707761dae35d762"
    }
```

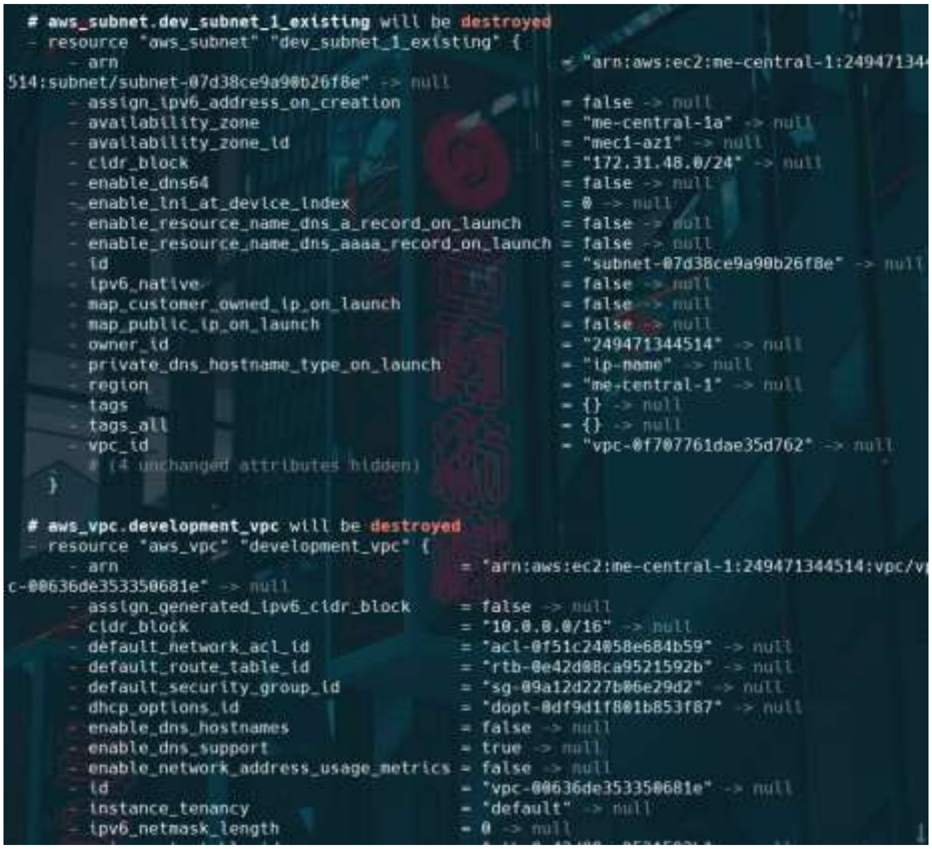


```
Plan: 1 to add, 0 to change, 0 to destroy.

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-07d38ce9a90b26f8e]

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

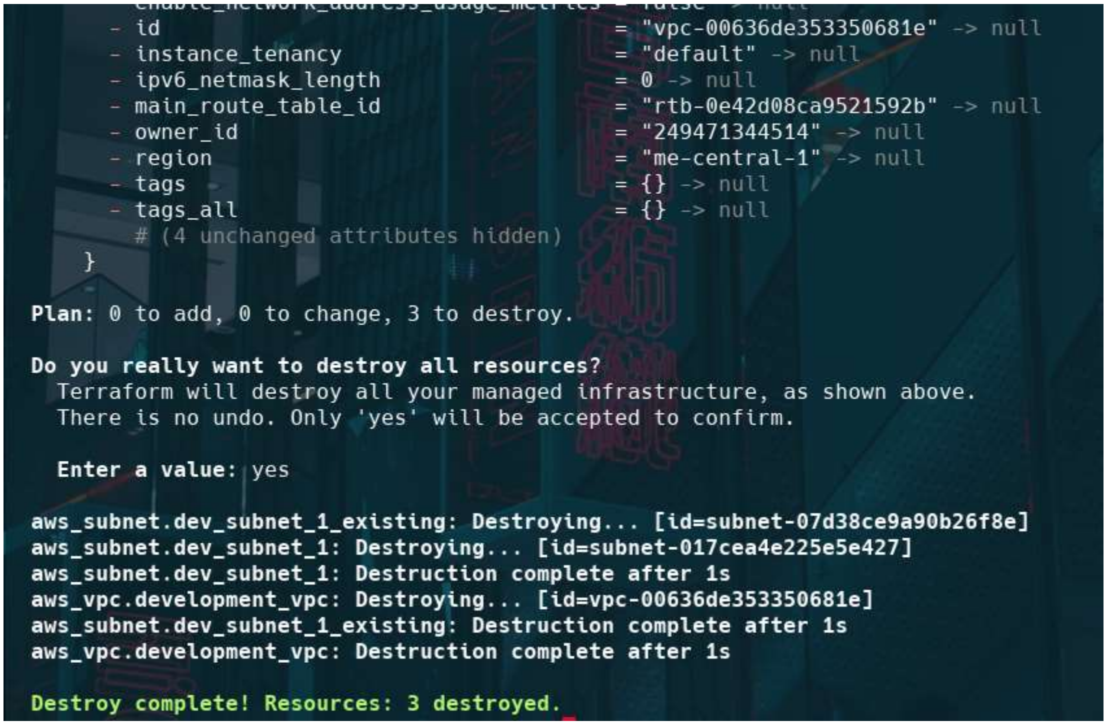
4. Destroy all resources:



```
@Anara-hayat → /workspaces/lab9 (main) $ terraform destroy
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-00636de353350681e]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-07d38ce9a90b26f8e]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-017cea4e225e5e427]

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:me-central-1:24947134
514:subnet/subnet-017cea4e225e5e427" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "me-central-1a" -> null
      - availability_zone_id                           = "mec1-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-017cea4e225e5e427" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "249471344514" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - region                                         = "me-central-1" -> null
      - tags                                           = {} -> null
      - tags_all                                       = {} -> null
      - vpc_id                                         = "vpc-00636de353350681e" -> null
        # (4 unchanged attributes hidden)
    }
```

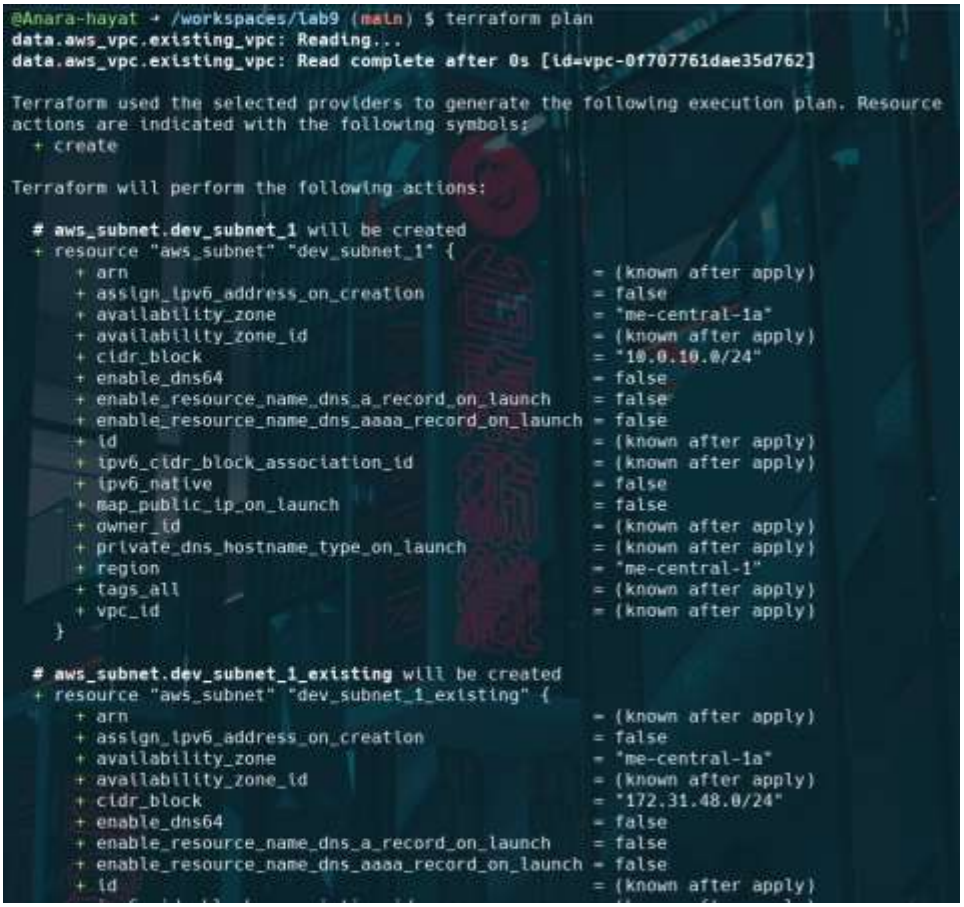


```
  # aws_subnet.dev_subnet_1_existing will be destroyed
  - resource "aws_subnet" "dev_subnet_1_existing" {
      - arn                                            = "arn:aws:ec2:me-central-1:24947134
514:subnet/subnet-07d38ce9a90b26f8e" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "me-central-1a" -> null
      - availability_zone_id                           = "mec1-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-07d38ce9a90b26f8e" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "249471344514" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
      - region                                         = "me-central-1" -> null
      - tags                                           = {} -> null
      - tags_all                                       = {} -> null
      - vpc_id                                         = "vpc-0f707761dae35d762" -> null
        # (4 unchanged attributes hidden)
    }

  # aws_vpc.development_vpc will be destroyed
  - resource "aws_vpc" "development_vpc" {
      - arn                                  = "arn:aws:ec2:me-central-1:249471344514:vpc/vp
c-00636de353350681e" -> null
      - assign_generated_ipv6_cidr_block     = false -> null
      - cidr_block                           = "10.0.0.0/16" -> null
      - default_network_acl_id               = "acl-0f51c24058e684b59" -> null
      - default_route_table_id               = "rtb-0e42d08ca9521592b" -> null
      - default_security_group_id            = "sg-09a12d227b06e29d2" -> null
      - dhcp_options_id                      = "dopt-0df9d1f801b853f87" -> null
      - enable_dns_hostnames                 = false -> null
      - enable_dns_support                   = true -> null
      - enable_network_address_usage_metrics = false -> null
      - id                                   = "vpc-00636de353350681e" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
```



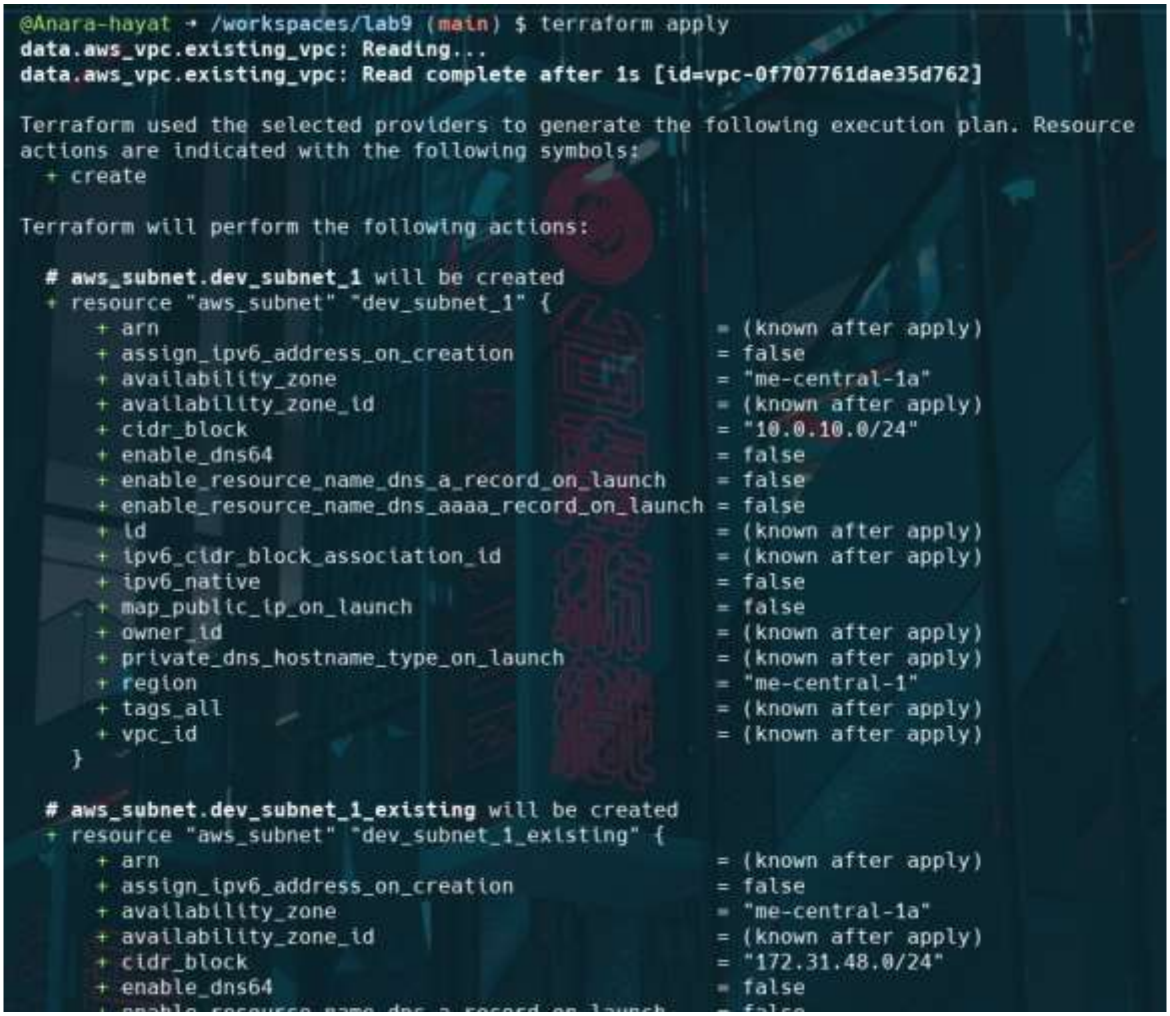
```
      enable_network_address_usage_metrics = false -> null
      - id                                  = "vpc-00636de353350681e" -> null
      - instance_tenancy                    = "default" -> null
      - ipv6_netmask_length                 = 0 -> null
      - main_route_table_id                 = "rtb-0e42d08ca9521592b" -> null
      - owner_id                            = "249471344514" -> 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-07d38ce9a90b26f8e]
aws_subnet.dev_subnet_1: Destroying... [id=subnet-017cea4e225e5e427]
aws_subnet.dev_subnet_1: Destruction complete after 1s
aws_vpc.development_vpc: Destroying... [id=vpc-00636de353350681e]
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s
aws_vpc.development_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
```

5. Plan configuration:

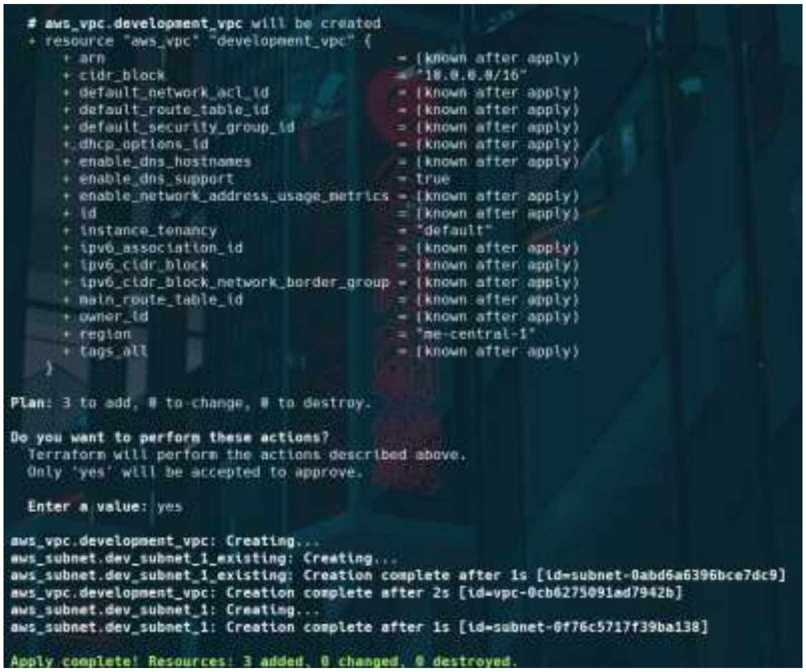


```
@Anara-hayat → /workspaces/lab9 (main) $ terraform plan
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-0f707761dae35d762]

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
  + 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)
```

6. Apply again:



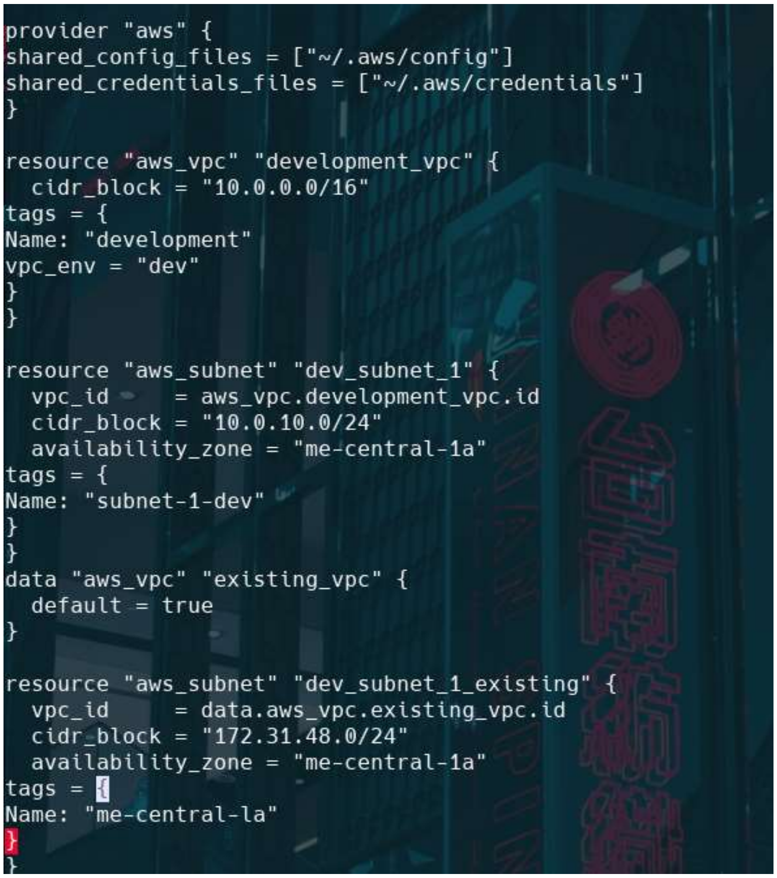
```
@Anara-hayat → /workspaces/lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]

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
  + 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
```



```
  # 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: 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-0abd6a6396bce7dc9]
aws_vpc.development_vpc: Creation complete after 2s [id=vpc-0cb6275091ad7942b]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-0f76c5717f39ba138]

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

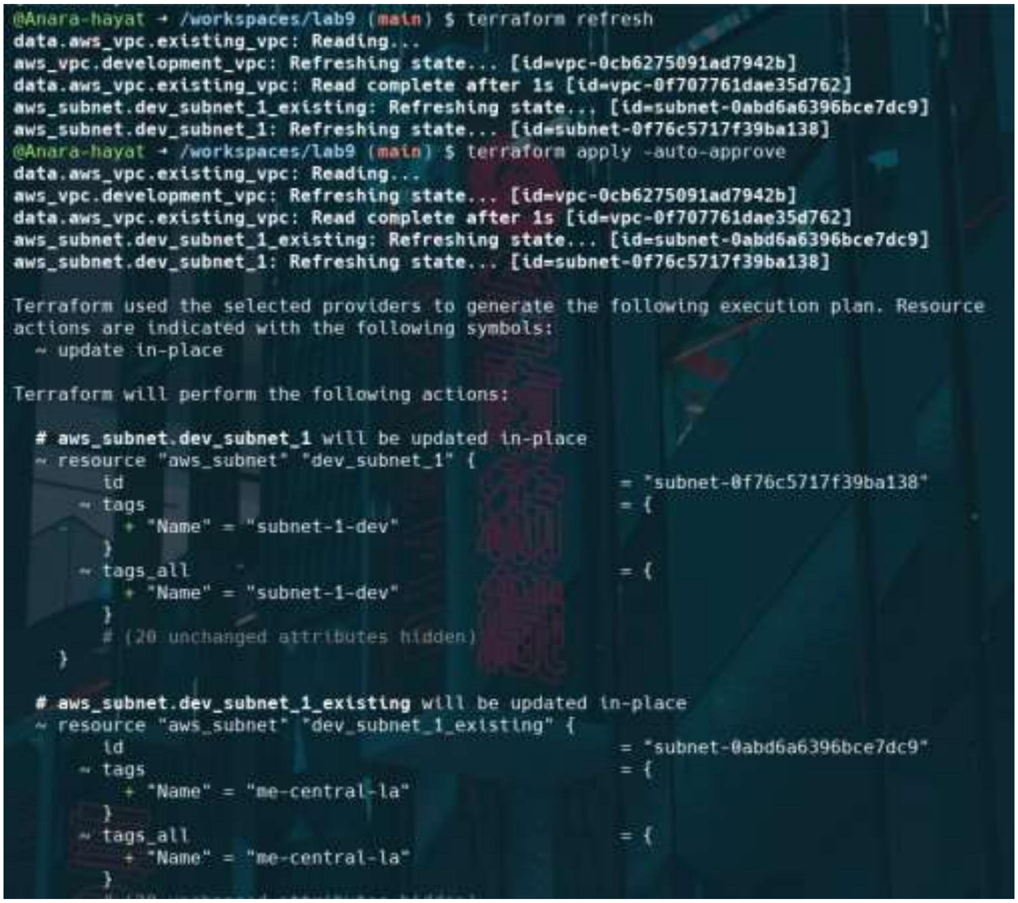
C. Tagging Resources

1. Modify main.tf to add tags:

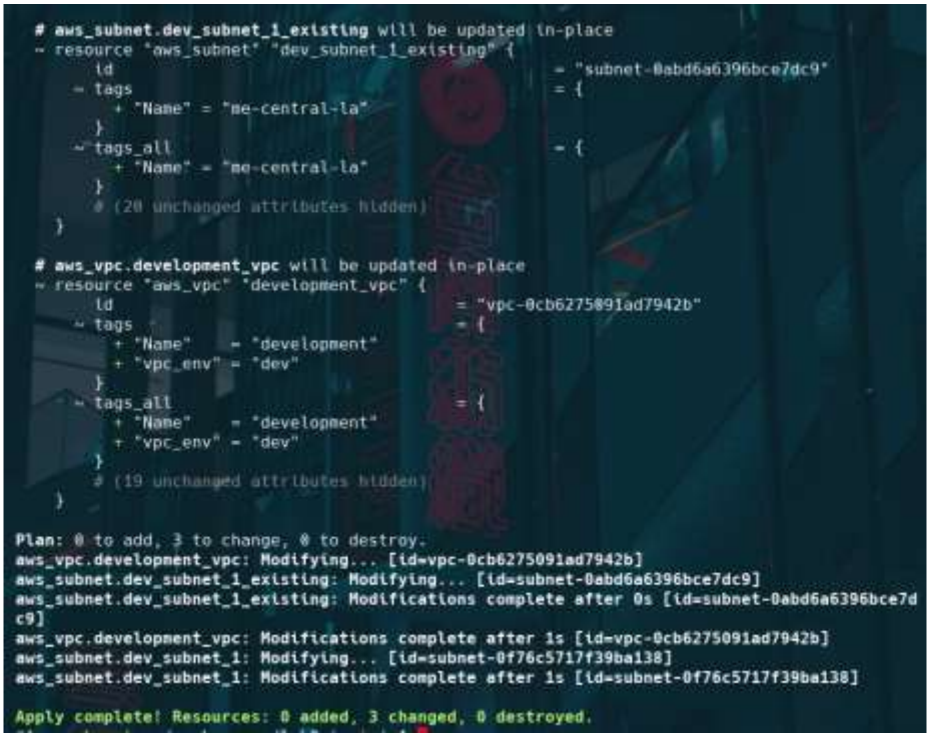


```
provider "aws" {
shared_config_files = ["~/.aws/config"]
shared_credentials_files = ["~/.aws/credentials"]
}

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"
}
}
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 = "me-central-1a"
tags = {
Name: "me-central-la"
}
}
```

2. Run:

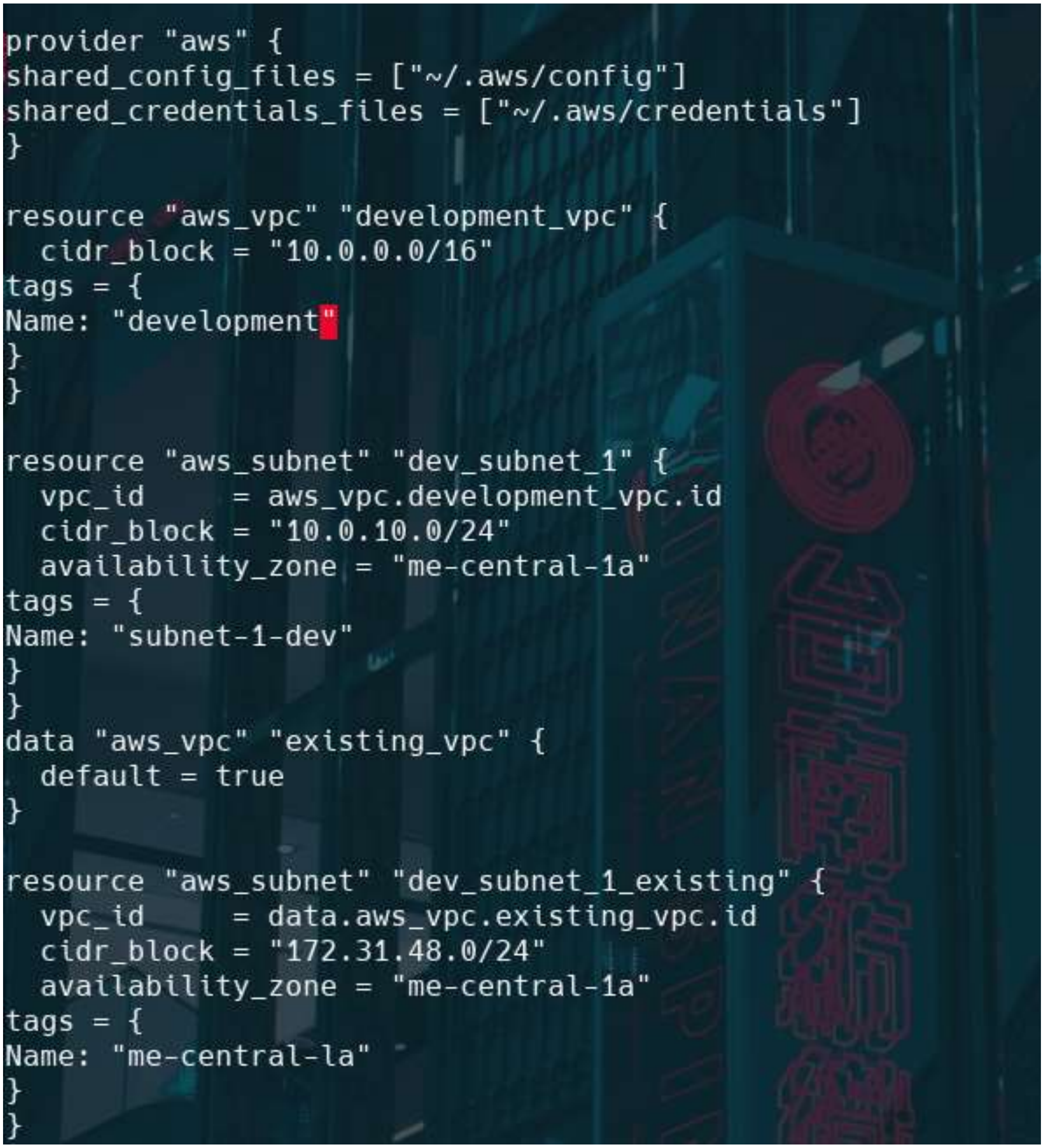


```
@Anara-hayat → /workspaces/lab9 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0cb6275091ad7942b]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0abd6a6396bce7dc9]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0f76c5717f39ba138]
@Anara-hayat → /workspaces/lab9 (main) $ terraform apply -auto-approve
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0cb6275091ad7942b]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0abd6a6396bce7dc9]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0f76c5717f39ba138]

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-0f76c5717f39ba138"
      ~ 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-0abd6a6396bce7dc9"
      ~ tags                                           = {
          + "Name" = "me-central-1a"
        }
      ~ tags_all                                       = {
          + "Name" = "me-central-1a"
        }
```

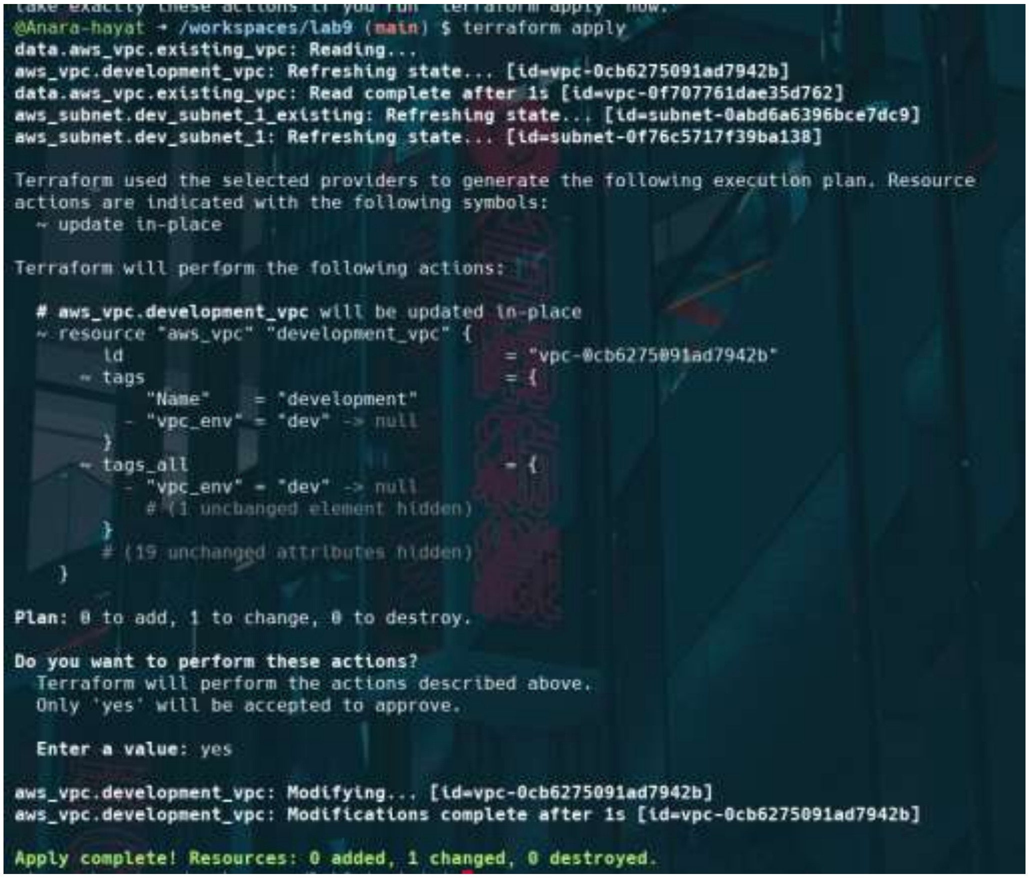


```
  # aws_subnet.dev_subnet_1_existing will be updated in-place
  ~ resource "aws_subnet" "dev_subnet_1_existing" {
        id                                             = "subnet-0abd6a6396bce7dc9"
      ~ tags                                           = {
          + "Name" = "me-central-1a"
        }
      ~ tags_all                                       = {
          + "Name" = "me-central-1a"
        }
        # (20 unchanged attributes hidden)
    }

  # aws_vpc.development_vpc will be updated in-place
  ~ resource "aws_vpc" "development_vpc" {
        id                                   = "vpc-0cb6275091ad7942b"
      ~ tags                                 = {
          + "Name"    = "development"
          + "vpc_env" = "dev"
        }
      ~ tags_all                             = {
          + "Name"    = "development"
          + "vpc_env" = "dev"
        }
        # (19 unchanged attributes hidden)
    }

Plan: 0 to add, 3 to change, 0 to destroy.
aws_vpc.development_vpc: Modifying... [id=vpc-0cb6275091ad7942b]
aws_subnet.dev_subnet_1_existing: Modifying... [id=subnet-0abd6a6396bce7dc9]
aws_subnet.dev_subnet_1_existing: Modifications complete after 0s [id=subnet-0abd6a6396bce7d
c9]
aws_vpc.development_vpc: Modifications complete after 1s [id=vpc-0cb6275091ad7942b]
aws_subnet.dev_subnet_1: Modifying... [id=subnet-0f76c5717f39ba138]
aws_subnet.dev_subnet_1: Modifications complete after 1s [id=subnet-0f76c5717f39ba138]

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

3. Remove vpc_env = "dev" tag from development_vpc resource, re-plan/apply:



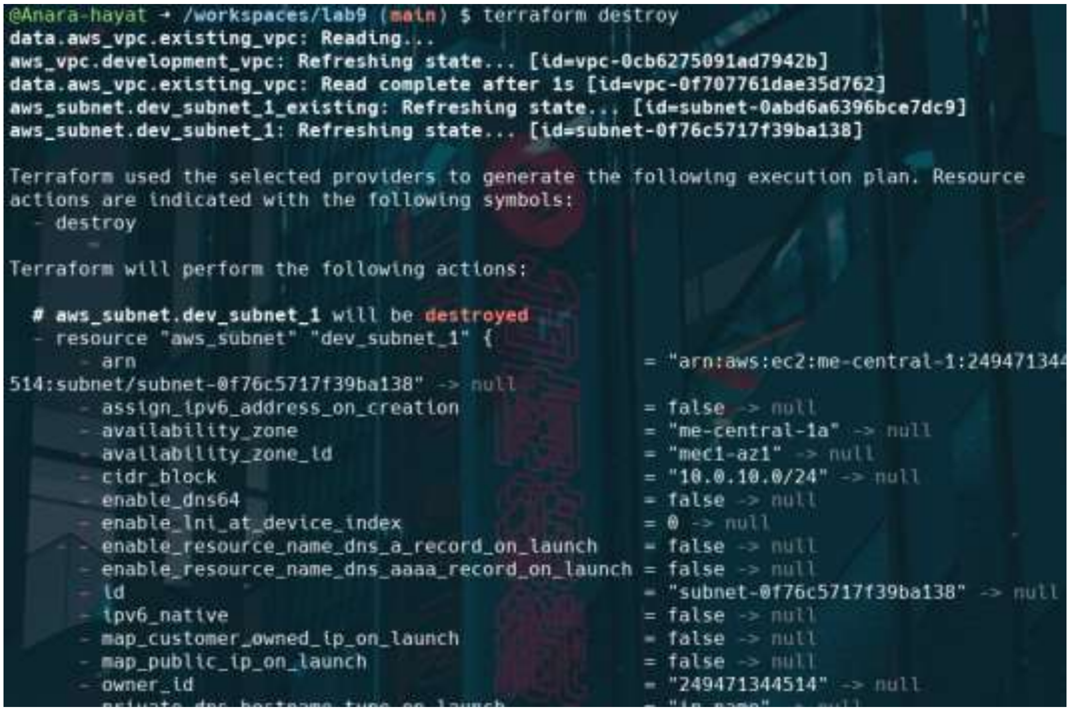
```
provider "aws" {
shared_config_files = ["~/.aws/config"]
shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_vpc" "development_vpc" {
  cidr_block = "10.0.0.0/16"
tags = {
Name: "development"
}
}

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"
}
}
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 = "me-central-1a"
tags = {
Name: "me-central-la"
}
}
```



```
take exactly these actions if you run "terraform apply" now.
@Anara-hayat → /workspaces/lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0cb6275091ad7942b]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0abd6a6396bce7dc9]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0f76c5717f39ba138]

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-0cb6275091ad7942b"
      ~ 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.

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-0cb6275091ad7942b]
aws_vpc.development_vpc: Modifications complete after 1s [id=vpc-0cb6275091ad7942b]

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

Task 5 — State File Inspection & Terraform State Commands

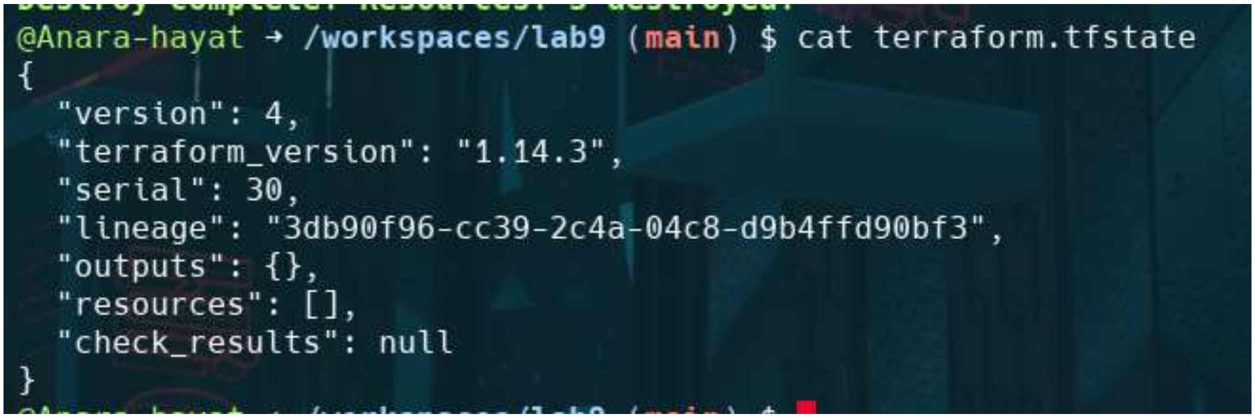
1. Destroy all resources:



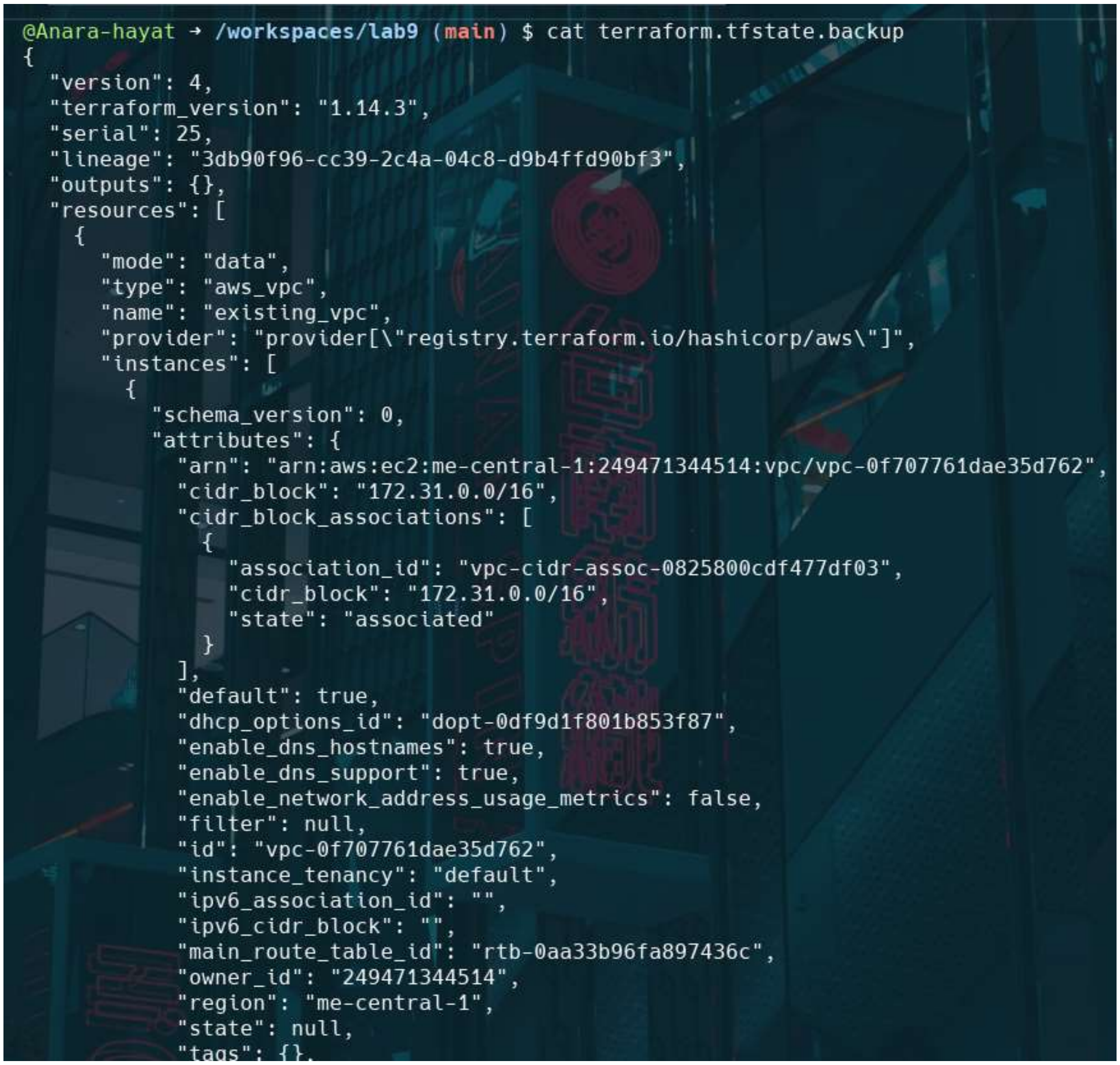
```
@Anara-hayat → /workspaces/lab9 (main) $ terraform destroy
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0cb6275091ad7942b]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0abd6a6396bce7dc9]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0f76c5717f39ba138]

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:me-central-1:249471344
514:subnet/subnet-0f76c5717f39ba138" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "me-central-1a" -> null
      - availability_zone_id                           = "mec1-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-0f76c5717f39ba138" -> null
      - ipv6_native                                    = false -> null
      - map_customer_owned_ip_on_launch                = false -> null
      - map_public_ip_on_launch                        = false -> null
      - owner_id                                       = "249471344514" -> null
      - private_dns_hostname_type_on_launch            = "ip-name" -> null
```

2. Inspect state files after destroy:

   Run:

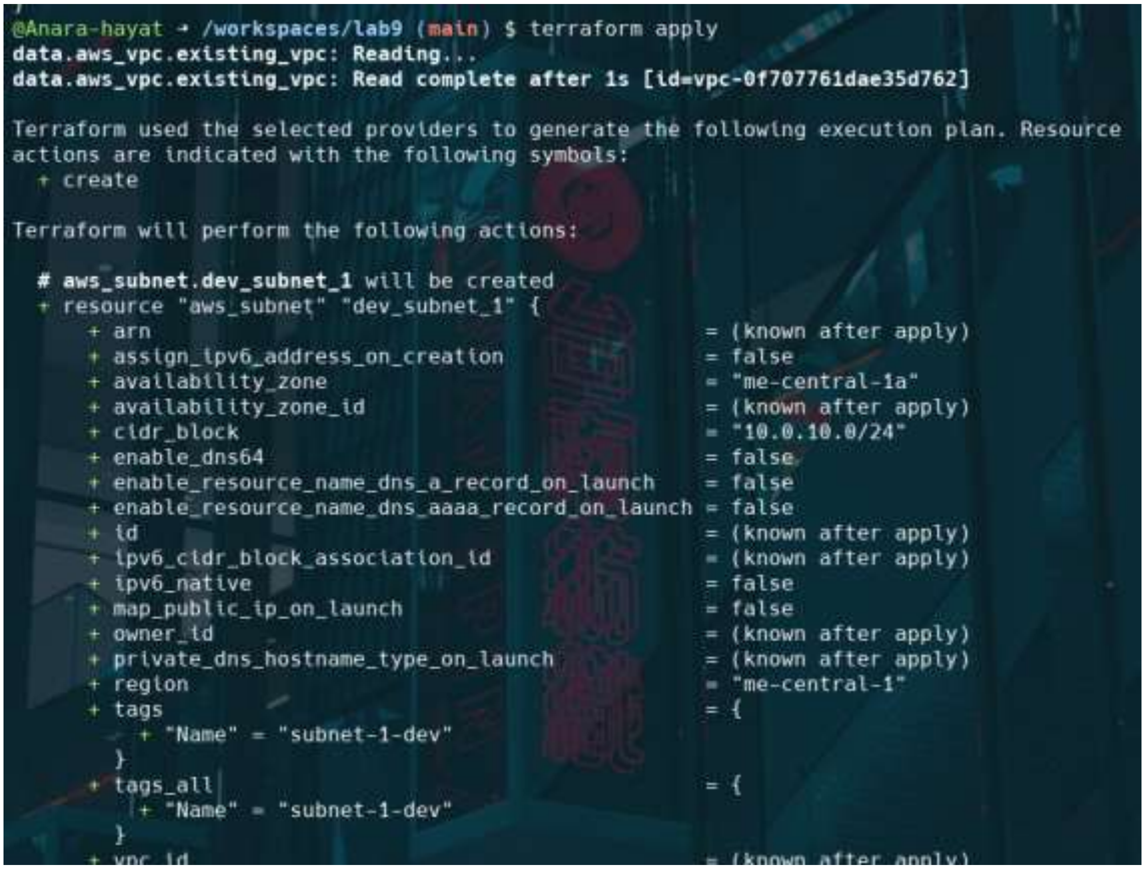


```
Destroy complete! Resources: 3 destroyed.
@Anara-hayat → /workspaces/lab9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 30,
  "lineage": "3db90f96-cc39-2c4a-04c8-d9b4ffd90bf3",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Anara-hayat → /workspaces/lab9 (main) $
```

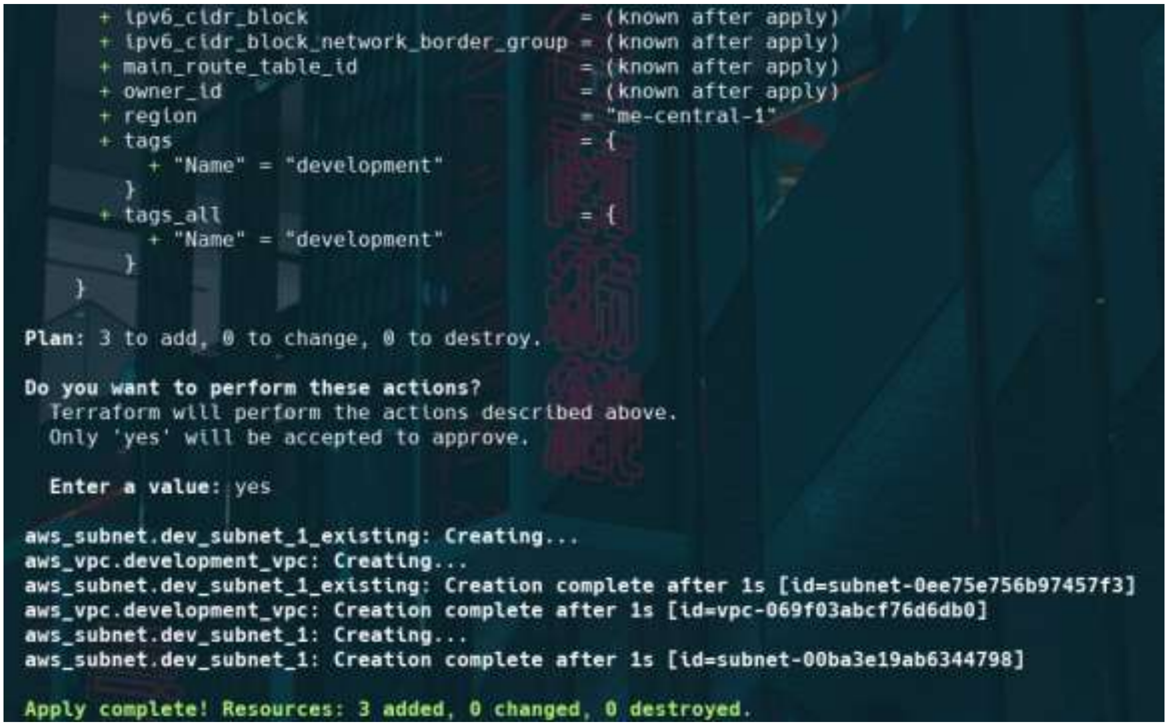


```
@Anara-hayat → /workspaces/lab9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 25,
  "lineage": "3db90f96-cc39-2c4a-04c8-d9b4ffd90bf3",
  "outputs": {},
  "resources": [
    {
      "mode": "data",
      "type": "aws_vpc",
      "name": "existing_vpc",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "arn": "arn:aws:ec2:me-central-1:249471344514:vpc/vpc-0f707761dae35d762",
            "cidr_block": "172.31.0.0/16",
            "cidr_block_associations": [
              {
                "association_id": "vpc-cidr-assoc-0825800cdf477df03",
                "cidr_block": "172.31.0.0/16",
                "state": "associated"
              }
            ],
            "default": true,
            "dhcp_options_id": "dopt-0df9d1f801b853f87",
            "enable_dns_hostnames": true,
            "enable_dns_support": true,
            "enable_network_address_usage_metrics": false,
            "filter": null,
            "id": "vpc-0f707761dae35d762",
            "instance_tenancy": "default",
            "ipv6_association_id": "",
            "ipv6_cidr_block": "",
            "main_route_table_id": "rtb-0aa33b96fa897436c",
            "owner_id": "249471344514",
            "region": "me-central-1",
            "state": null,
            "tags": {},
```

3. Recreate resources

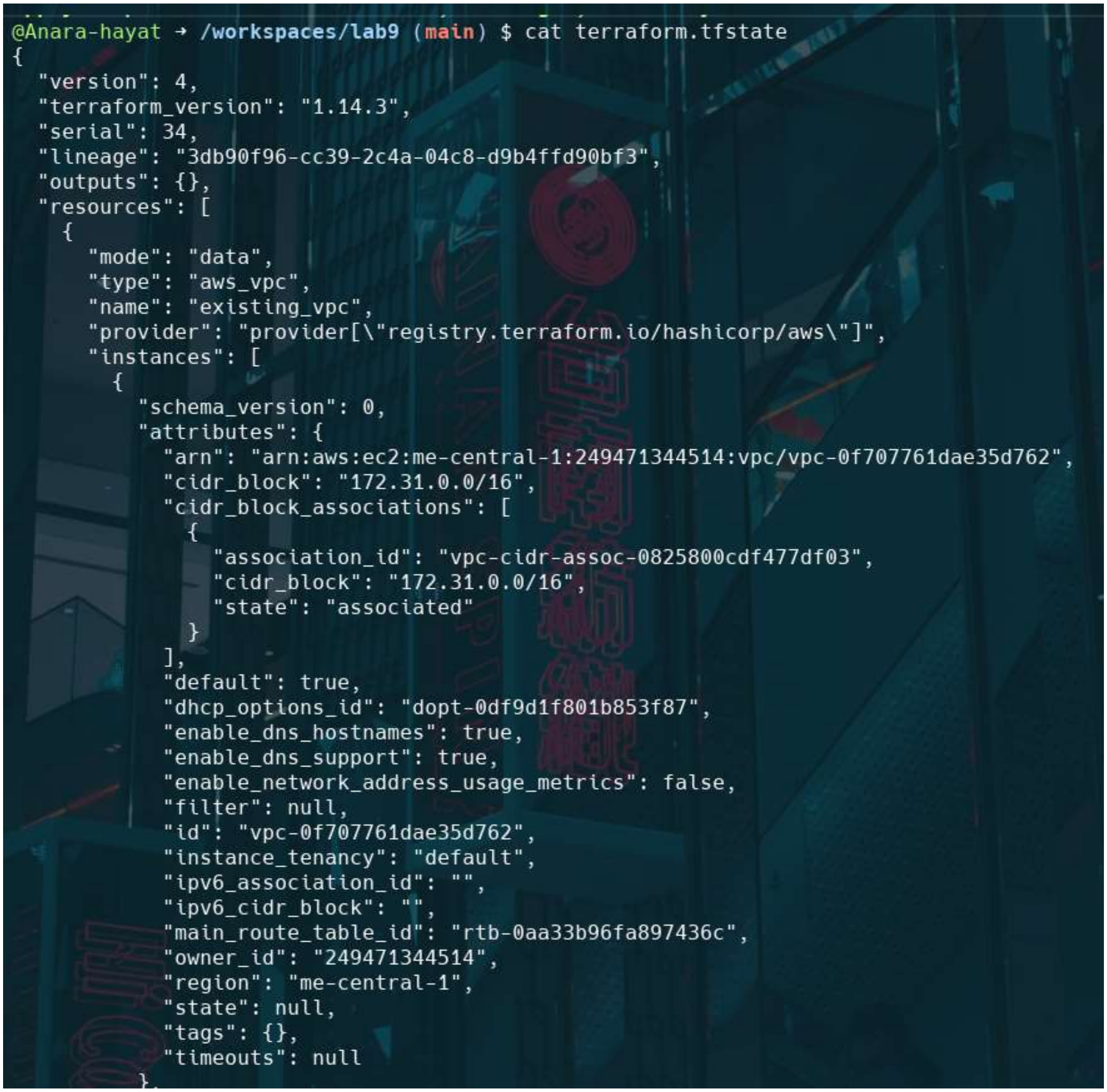


```
@Anara-hayat → /workspaces/lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]

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                                           = {
          + "Name" = "subnet-1-dev"
        }
      + tags_all                                       = {
          + "Name" = "subnet-1-dev"
        }
      + vpc_id                                         = (known after apply)
```

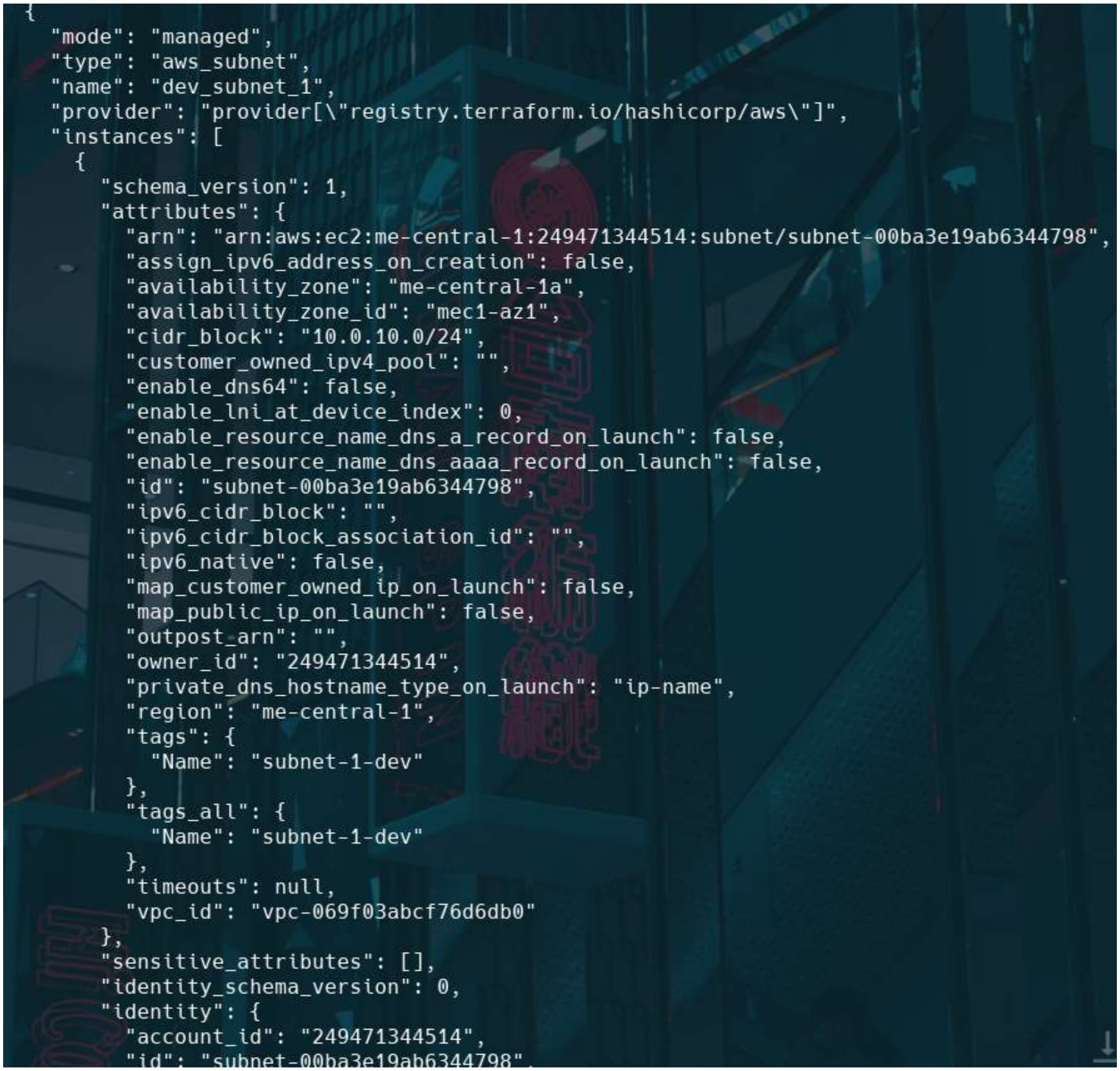


```
      + ipv6_cidr_block                      = (known after apply)
      + ipv6_cidr_block_network_border_group = (known after apply)
      + main_route_table_id                  = (known after apply)
      + owner_id                             = (known after apply)
      + region                               = "me-central-1"
      + tags                                 = {
          + "Name" = "development"
        }
      + tags_all                             = {
          + "Name" = "development"
        }
    }

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_subnet.dev_subnet_1_existing: Creating...
aws_vpc.development_vpc: Creating...
aws_subnet.dev_subnet_1_existing: Creation complete after 1s [id=subnet-0ee75e756b97457f3]
aws_vpc.development_vpc: Creation complete after 1s [id=vpc-069f03abcf76d6db0]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-00ba3e19ab6344798]

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

4. View state files:

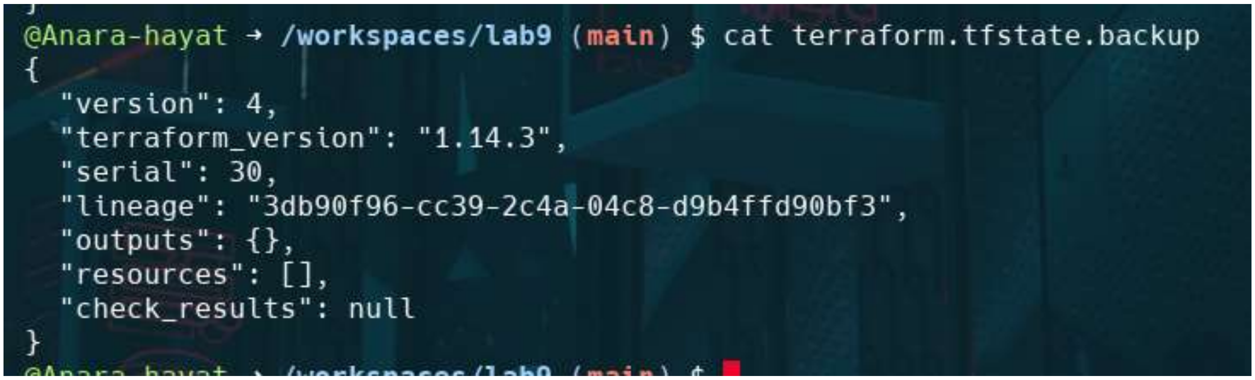


```
@Anara-hayat → /workspaces/lab9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 34,
  "lineage": "3db90f96-cc39-2c4a-04c8-d9b4ffd90bf3",
  "outputs": {},
  "resources": [
    {
      "mode": "data",
      "type": "aws_vpc",
      "name": "existing_vpc",
      "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "arn": "arn:aws:ec2:me-central-1:249471344514:vpc/vpc-0f707761dae35d762",
            "cidr_block": "172.31.0.0/16",
            "cidr_block_associations": [
              {
                "association_id": "vpc-cidr-assoc-0825800cdf477df03",
                "cidr_block": "172.31.0.0/16",
                "state": "associated"
              }
            ],
            "default": true,
            "dhcp_options_id": "dopt-0df9d1f801b853f87",
            "enable_dns_hostnames": true,
            "enable_dns_support": true,
            "enable_network_address_usage_metrics": false,
            "filter": null,
            "id": "vpc-0f707761dae35d762",
            "instance_tenancy": "default",
            "ipv6_association_id": "",
            "ipv6_cidr_block": "",
            "main_route_table_id": "rtb-0aa33b96fa897436c",
            "owner_id": "249471344514",
            "region": "me-central-1",
            "state": null,
            "tags": {},
            "timeouts": null
          },
```



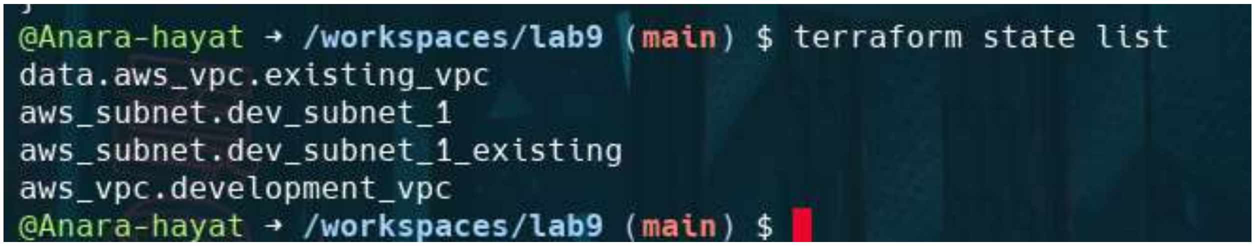
```
{
  "mode": "managed",
  "type": "aws_subnet",
  "name": "dev_subnet_1",
  "provider": "provider[\"registry.terraform.io/hashicorp/aws\"]",
  "instances": [
    {
      "schema_version": 1,
      "attributes": {
        "arn": "arn:aws:ec2:me-central-1:249471344514:subnet/subnet-00ba3e19ab6344798",
        "assign_ipv6_address_on_creation": false,
        "availability_zone": "me-central-1a",
        "availability_zone_id": "mec1-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-00ba3e19ab6344798",
        "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": "249471344514",
        "private_dns_hostname_type_on_launch": "ip-name",
        "region": "me-central-1",
        "tags": {
          "Name": "subnet-1-dev"
        },
        "tags_all": {
          "Name": "subnet-1-dev"
        },
        "timeouts": null,
        "vpc_id": "vpc-069f03abcf76d6db0"
      },
      "sensitive_attributes": [],
      "identity_schema_version": 0,
      "identity": {
        "account_id": "249471344514",
        "id": "subnet-00ba3e19ab6344798",
```

Run:



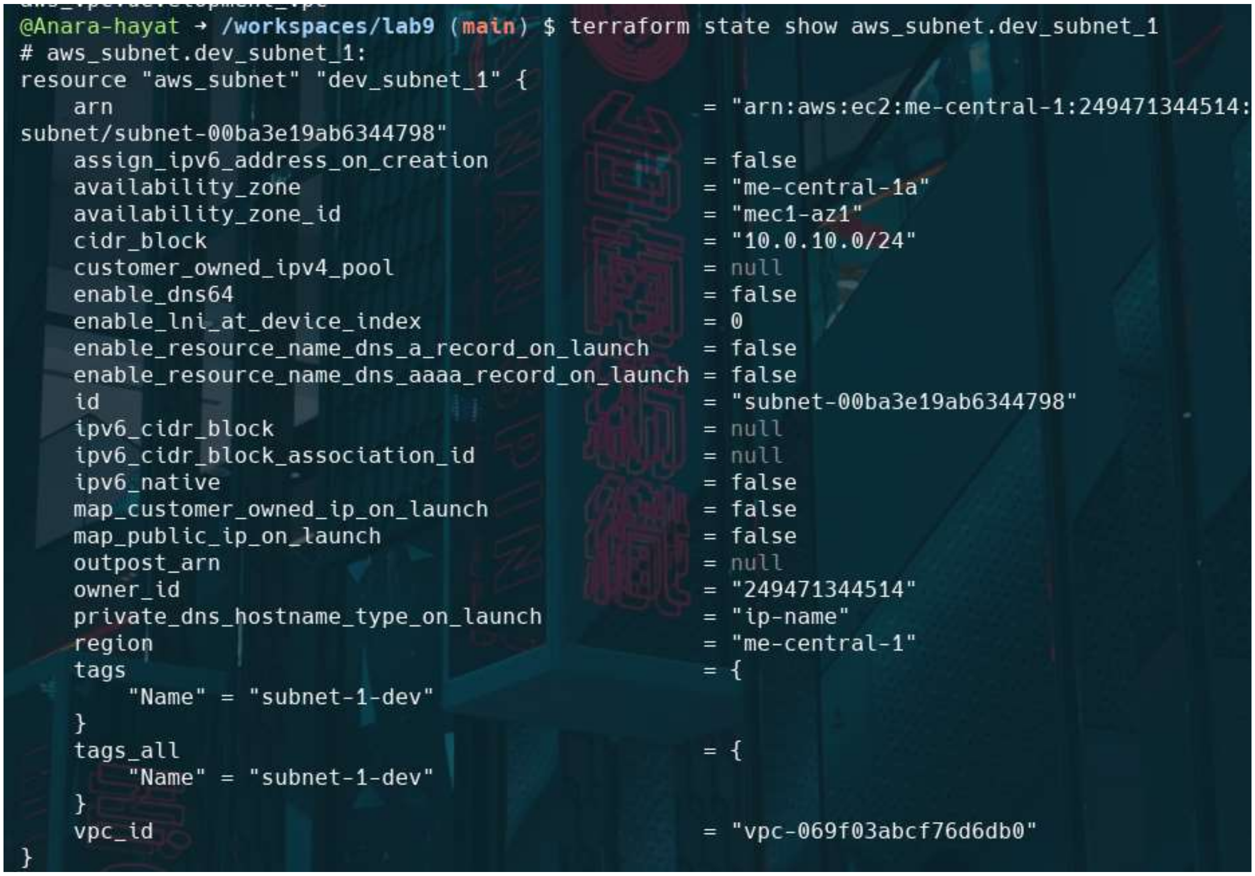
```
@Anara-hayat → /workspaces/lab9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 30,
  "lineage": "3db90f96-cc39-2c4a-04c8-d9b4ffd90bf3",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Anara-hayat → /workspaces/lab9 (main) $
```

5. List Resources:



```
@Anara-hayat → /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
@Anara-hayat → /workspaces/lab9 (main) $
```

6. Show full attributes:

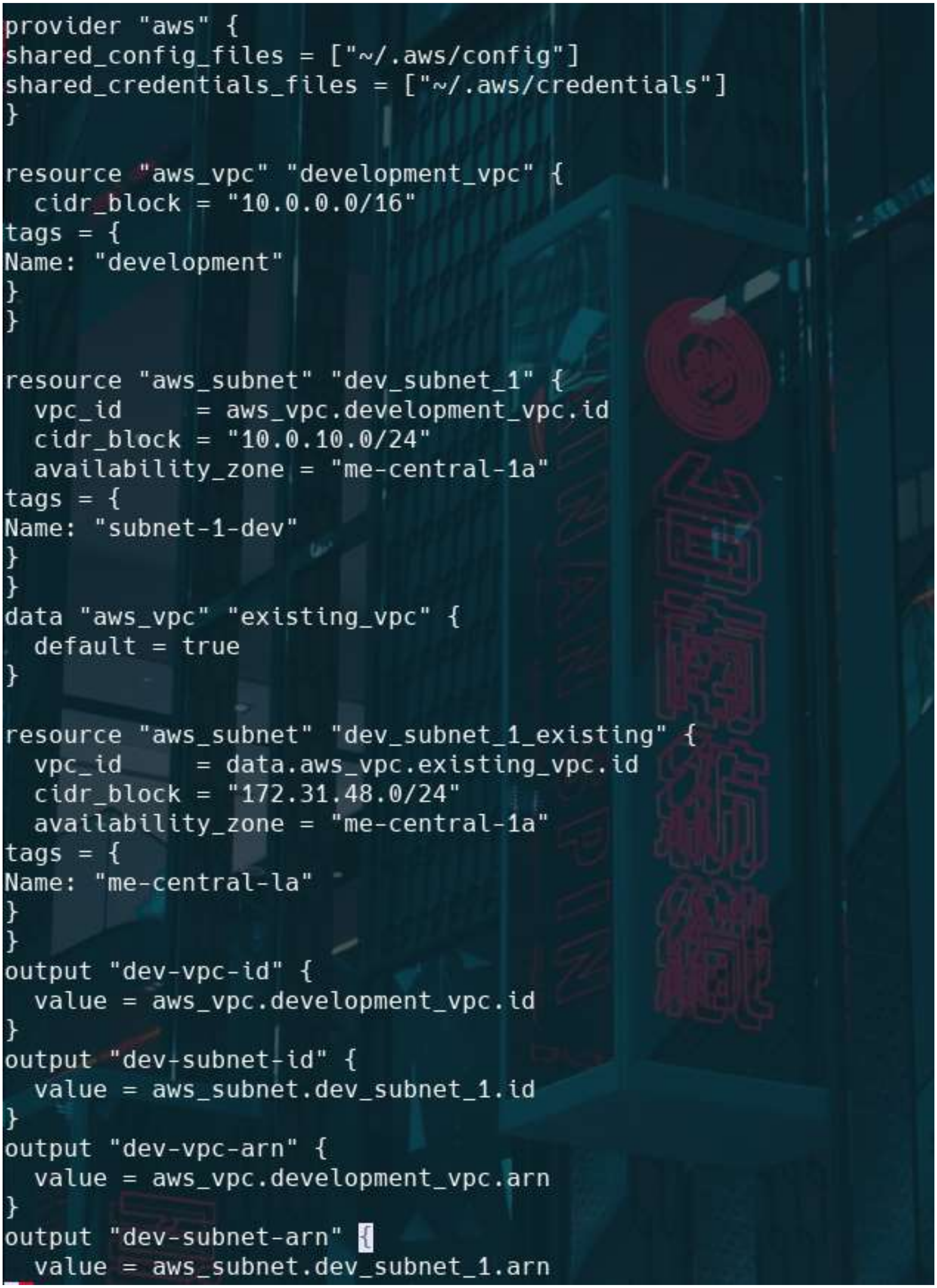


```
@Anara-hayat → /workspaces/lab9 (main) $ terraform state show aws_subnet.dev_subnet_1
# aws_subnet.dev_subnet_1:
resource "aws_subnet" "dev_subnet_1" {
    arn                                            = "arn:aws:ec2:me-central-1:249471344514:
subnet/subnet-00ba3e19ab6344798"
    assign_ipv6_address_on_creation                = false
    availability_zone                              = "me-central-1a"
    availability_zone_id                           = "mec1-az1"
    cidr_block                                     = "10.0.10.0/24"
    customer_owned_ipv4_pool                       = null
    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-00ba3e19ab6344798"
    ipv6_cidr_block                                = null
    ipv6_cidr_block_association_id                 = null
    ipv6_native                                    = false
    map_customer_owned_ip_on_launch                = false
    map_public_ip_on_launch                        = false
    outpost_arn                                    = null
    owner_id                                       = "249471344514"
    private_dns_hostname_type_on_launch            = "ip-name"
    region                                         = "me-central-1"
    tags                                           = {
        "Name" = "subnet-1-dev"
    }
    tags_all                                       = {
        "Name" = "subnet-1-dev"
    }
    vpc_id                                         = "vpc-069f03abcf76d6db0"
}
```

7. Note: Do **NOT** run the following (for information only):
   terraform state rm <resource-name>

Task 6 — Terraform Outputs & Attributes Reporting

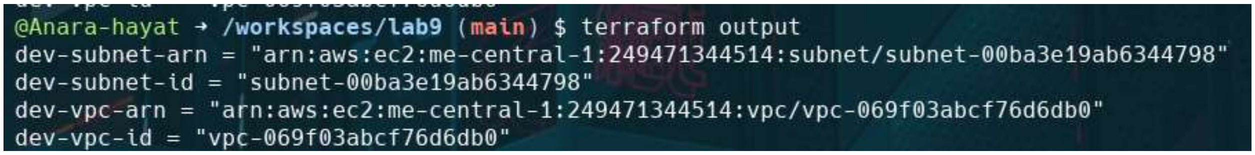
1. Add outputs in main.tf



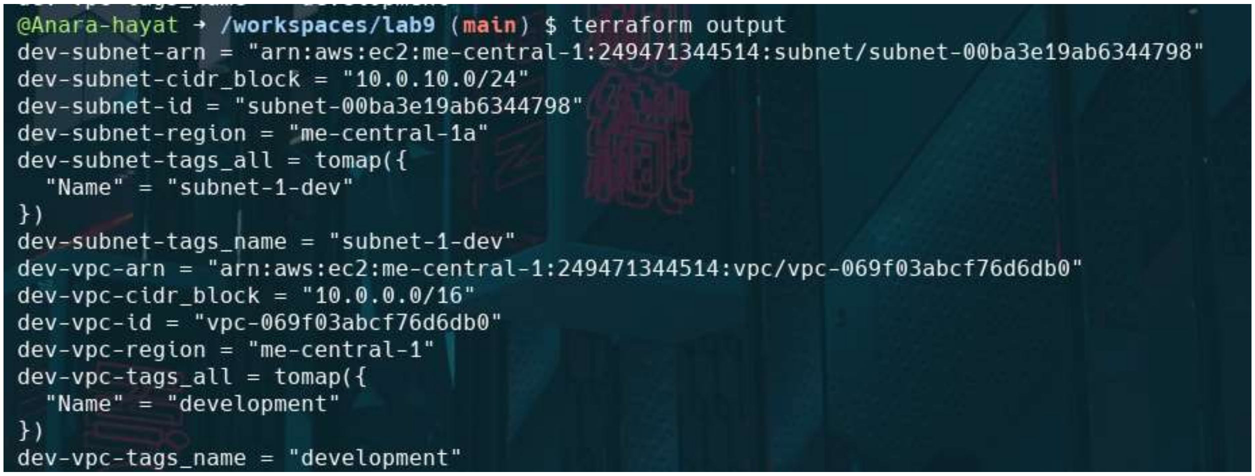
```
provider "aws" {
shared_config_files = ["~/.aws/config"]
shared_credentials_files = ["~/.aws/credentials"]
}

resource "aws_vpc" "development_vpc" {
  cidr_block = "10.0.0.0/16"
tags = {
Name: "development"
}
}

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"
}
}
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 = "me-central-1a"
tags = {
Name: "me-central-la"
}
}
output "dev-vpc-id" {
  value = aws_vpc.development_vpc.id
}
output "dev-subnet-id" {
  value = aws_subnet.dev_subnet_1.id
}
output "dev-vpc-arn" {
  value = aws_vpc.development_vpc.arn
}
output "dev-subnet-arn" {
  value = aws_subnet.dev_subnet_1.arn
```

Return ALL of the following output attributes from your VPC and Subnet resources:

- a) cidr_block
- b) region
- c) tags.Name
- d) tags_all



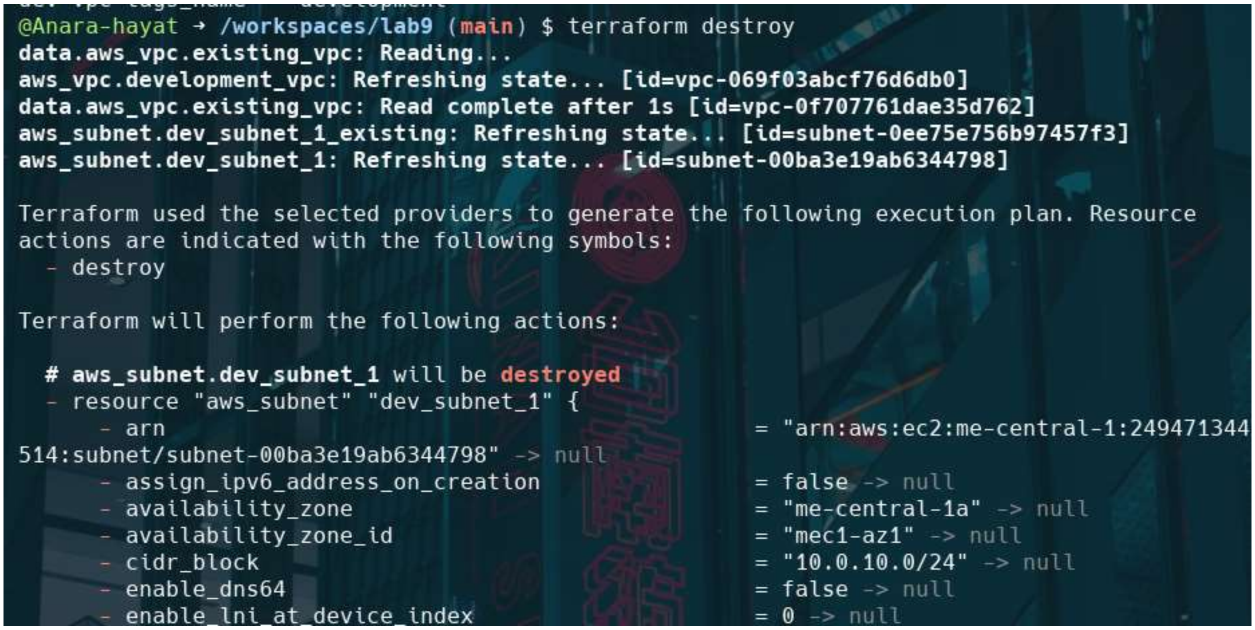
```
@Anara-hayat → /workspaces/lab9 (main) $ terraform output
dev-subnet-arn = "arn:aws:ec2:me-central-1:249471344514:subnet/subnet-00ba3e19ab6344798"
dev-subnet-id = "subnet-00ba3e19ab6344798"
dev-vpc-arn = "arn:aws:ec2:me-central-1:249471344514:vpc/vpc-069f03abcf76d6db0"
dev-vpc-id = "vpc-069f03abcf76d6db0"
```



```
@Anara-hayat → /workspaces/lab9 (main) $ terraform output
dev-subnet-arn = "arn:aws:ec2:me-central-1:249471344514:subnet/subnet-00ba3e19ab6344798"
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-id = "subnet-00ba3e19ab6344798"
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-1:249471344514:vpc/vpc-069f03abcf76d6db0"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-id = "vpc-069f03abcf76d6db0"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
})
dev-vpc-tags_name = "development"
```

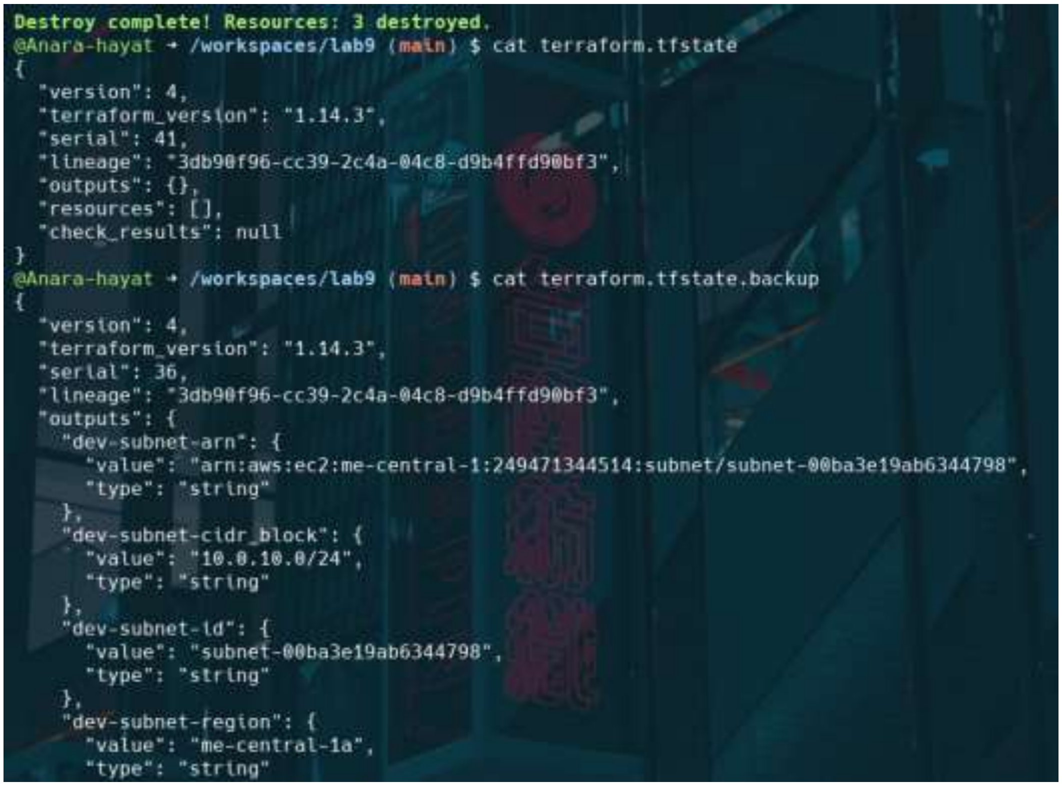
Cleanup — Delete Resources & State Verification

1. Destroy all resources:



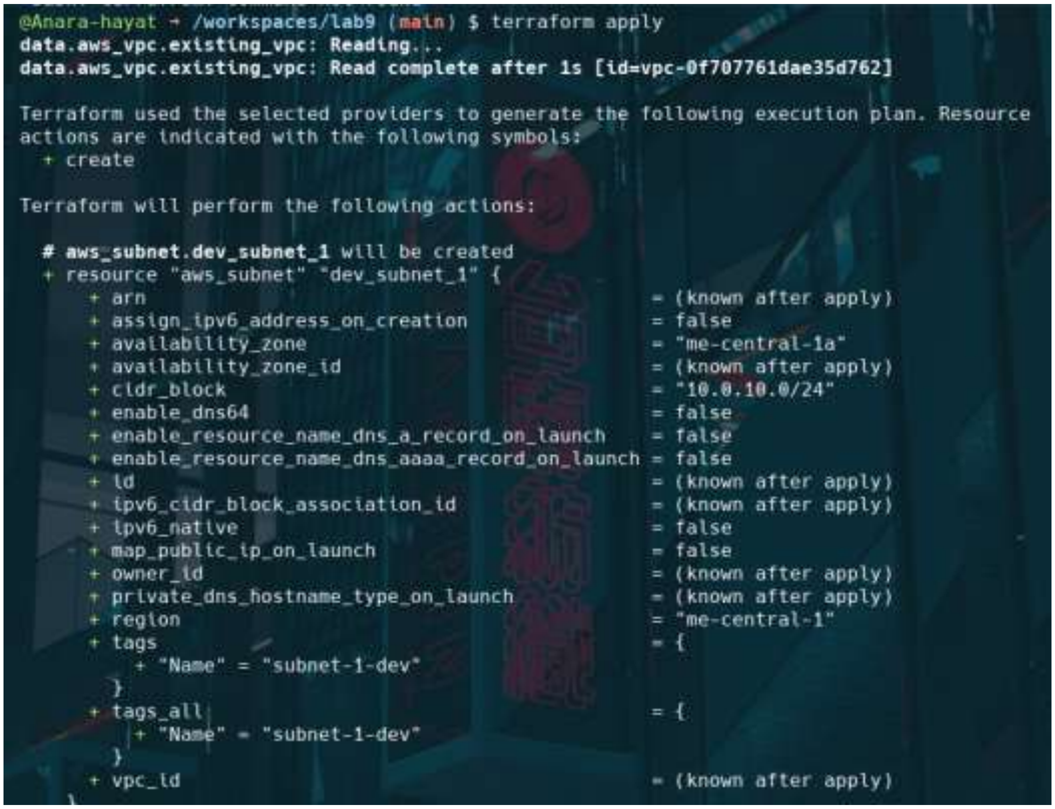
```
@Anara-hayat → /workspaces/lab9 (main) $ terraform destroy
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-069f03abcf76d6db0]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-0ee75e756b97457f3]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-00ba3e19ab6344798]

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:me-central-1:249471344
514:subnet/subnet-00ba3e19ab6344798" -> null
      - assign_ipv6_address_on_creation                = false -> null
      - availability_zone                              = "me-central-1a" -> null
      - availability_zone_id                           = "mec1-az1" -> null
      - cidr_block                                     = "10.0.10.0/24" -> null
      - enable_dns64                                   = false -> null
      - enable_lni_at_device_index                     = 0 -> null
```

2. Inspect state files:



```
Destroy complete! Resources: 3 destroyed.
@Anara-hayat → /workspaces/lab9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 41,
  "lineage": "3db90f96-cc39-2c4a-04c8-d9b4ffd90bf3",
  "outputs": {},
  "resources": [],
  "check_results": null
}
@Anara-hayat → /workspaces/lab9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 36,
  "lineage": "3db90f96-cc39-2c4a-04c8-d9b4ffd90bf3",
  "outputs": {
    "dev-subnet-arn": {
      "value": "arn:aws:ec2:me-central-1:249471344514:subnet/subnet-00ba3e19ab6344798",
      "type": "string"
    },
    "dev-subnet-cidr_block": {
      "value": "10.0.10.0/24",
      "type": "string"
    },
    "dev-subnet-id": {
      "value": "subnet-00ba3e19ab6344798",
      "type": "string"
    },
    "dev-subnet-region": {
      "value": "me-central-1a",
      "type": "string"
```

3. Reapply then compare state and backup files to see differences:



```
@Anara-hayat → /workspaces/lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-0f707761dae35d762]

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                                           = {
          + "Name" = "subnet-1-dev"
        }
      + tags_all                                       = {
          + "Name" = "subnet-1-dev"
        }
      + vpc_id                                         = (known after apply)
```