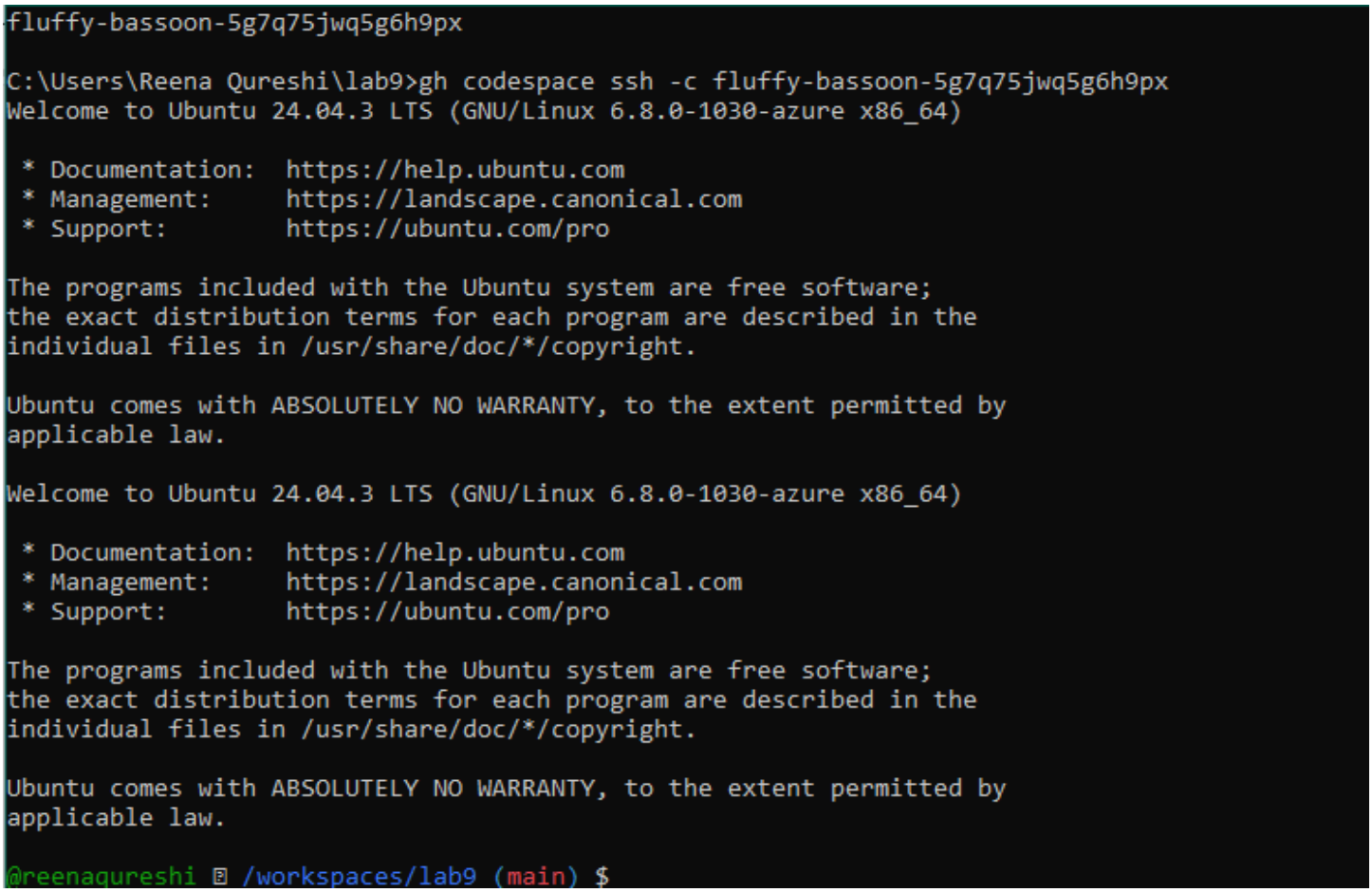
Name : Reena Qureshi

Reg No : 2023-BSE-052

Section : V-B

LAB 10

Task 1 — GitHub CLI, Codespace setup and authentication



Task 2 — Install AWS CLI inside the Codespace and configure it

Aws installed.

Install Terraform CLI

```
Get:26 http://archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1950 kB]
Get:27 http://archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [34.6 kB]
Get:28 http://archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [49.5 kB]
Fetched 35.7 MB in 6s (6145 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
@reenaqureshi /workspaces/lab9 (main) $ sudo apt install terraform
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  terraform
0 upgraded, 1 newly installed, 0 to remove and 51 not upgraded.
Need to get 30.6 MB of archives.
After this operation, 101 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com noble/main amd64 terraform amd64 1.14.3-1 [30.6 MB]
Fetched 30.6 MB in 0s (148 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) ...
@reenaqureshi /workspaces/lab9 (main) $ which terraform
/usr/bin/terraform
@reenaqureshi /workspaces/lab9 (main) $ terraform --version
Terraform v1.14.3
on linux_amd64
@reenaqureshi /workspaces/lab9 (main) $
```

C. Provider Configuration (main.tf)

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

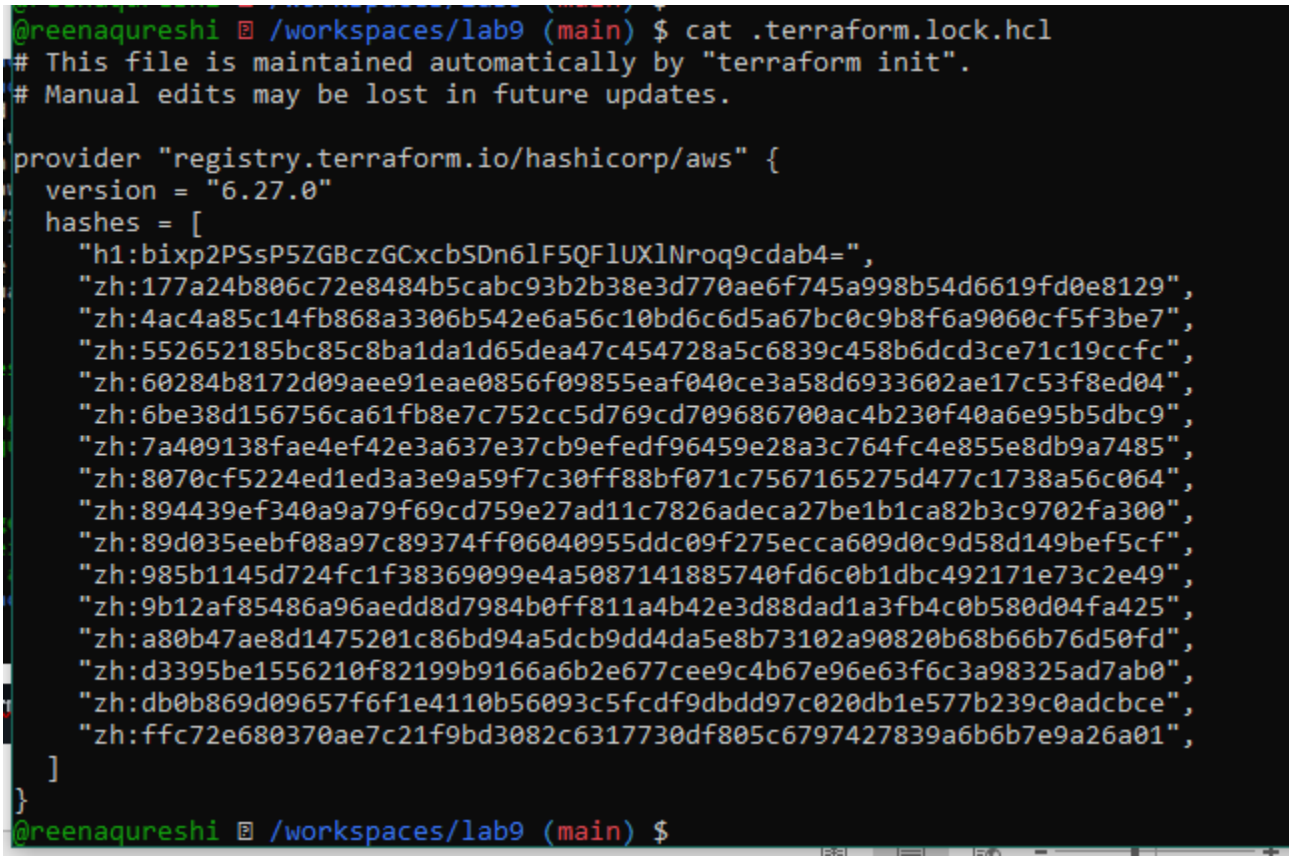
Initialize Terraform:

terraform init

```
@reenaqureshi /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.
@reenaqureshi /workspaces/lab9 (main) $
```

Show contents of .terraform.lock.hcl:

cat .terraform.lock.hcl



Show contents of .**terraform/** directory:

ls .terraform/

```
}
@reenaqureshi ➜ /workspaces/lab9 (main) $ ls .terraform/
providers
```

Task 3 — VPC/Subnet Creation, Initialization, Verification

Edit **main.tf** to add:

```
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"
}
```

Run:

terraform apply

```
@reenaqureshi ➜ /workspaces/lab9 (main) $ terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions a
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)
```

```
aws_vpc.development_vpc: Creating...
aws_vpc.development_vpc: Creation complete after 1s [id=vpc-0d5727099d23ec491]
aws_subnet.dev_subnet_1: Creating...
aws_subnet.dev_subnet_1: Creation complete after 1s [id=subnet-0c3f71551a015daf6]

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

Verify resources using AWS CLI:

```
@reenaqureshi /workspaces/lab9 (main) $ aws ec2 describe-subnets --filter "Name=subnet-id,
Values=<subnet-id>"
{
    "Subnets": []
}
@reenaqureshi /workspaces/lab9 (main) $ aws ec2 describe-vpcs --filter "Name=vpc-id,Values
=<vpc-id>"
{
    "Vpcs": []
}
@reenaqureshi /workspaces/lab9 (main) $
```

Task 4 — Data Source, Targeted Destroy, Tags

Data Source & Resource Creation

```
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"
}
~
~
~
~
~
~
~
~
```

Apply configuration:

```
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-04a44d49b9440a0d6"
    }

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-08103b01f7632e3c1]

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

Targeted Destroy & Refresh

Destroy only one resource:

```
      terraform plan

  Note that the -target option is not suitable for
  exceptional situations such as recovering from e
  specifically suggests to use it as part of an er


Destroy complete! Resources: 1 destroyed.
@reenaqureshi /workspaces/lab9 (main) $
```

Refresh state:

```
@reenaqureshi ⊡ /workspaces/lab9 (main) $ terraform refresh
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0d5727099d23ec491]
data.aws_vpc.existing_vpc: Read complete after 1s [id=vpc-04a44d49b9440a0d6]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-0c3f71551a015daf6]
```

Re-apply configuration:

terraform apply

```
  Enter a value: yes

aws_subnet.dev_subnet_1_existing: Creating...
aws_subnet.dev_subnet_1_existing: Creation complete after 1s [id=subnet-06a0a5d9fffaf37

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@reenaqureshi ⊡ /workspaces/lab9 (main) $
```

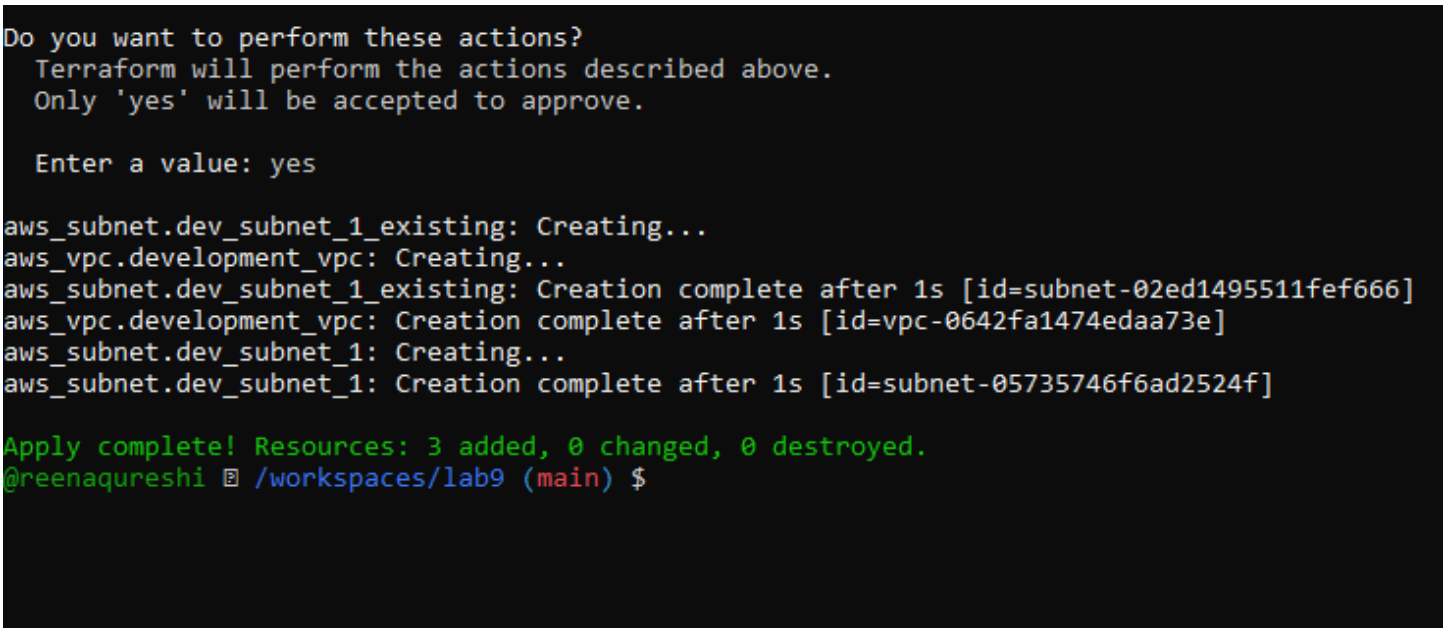
Destroy all resources:

terraform destroy

```
  # aws_vpc.development_vpc will be destroyed
  - resource "aws_vpc" "development_vpc" {
      - arn                                  = "arn:aws:ec2:me-central-1:6237051
:-0d5727099d23ec491" -> null
      - assign_generated_ipv6_cidr_block     = false -> null
      - cidr_block                           = "10.0.0.0/16" -> null
      - default_network_acl_id               = "acl-014d1c0230c4fb42a" -> null
      - default_route_table_id               = "rtb-031f210b1fc429dc9" -> null
      - default_security_group_id            = "sg-02283d585f15c294b" -> null
      - dhcp_options_id                      = "dopt-06a9132ca1eaa7ef9" -> null
      - enable_dns_hostnames                 = false -> null
      - enable_dns_support                   = true -> null
      - enable_network_address_usage_metrics = false -> null
      - id                                   = "vpc-0d5727099d23ec491" -> null
      - instance_tenancy                     = "default" -> null
      - ipv6_netmask_length                  = 0 -> null
      - main_route_table_id                  = "rtb-031f210b1fc429dc9" -> null
      - owner_id                             = "623705168110" -> 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: Destroying... [id=subnet-0c3f71551a015daf6]
```

Apply again:

terraform apply

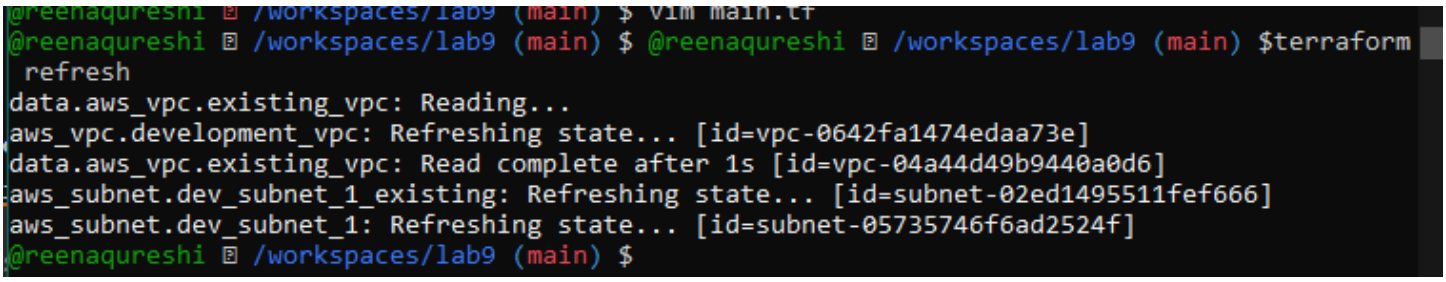


C. Tagging Resources

Modify **main.tf** to add tags:

```
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"
  }
}

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: "subnet-1-default"
  }
}
```

Use vim to update tags.



Recreate resources:

terraform apply

View state files:

Run:

List resources:

terraform state list

```
  + dev-vpc-tags_name     = "development"

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

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

  Enter a value: yes


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

Outputs:

dev-subnet-arn = "arn:aws:ec2:me-central-1:623705168110:subnet/subnet-05735746f6ad2524f"
dev-subnet-cidr_block = "10.0.10.0/24"
dev-subnet-id = "subnet-05735746f6ad2524f"
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-1:623705168110:vpc/vpc-0642fa1474edaa73e"
dev-vpc-cidr_block = "10.0.0.0/16"
dev-vpc-id = "vpc-0642fa1474edaa73e"
dev-vpc-region = "me-central-1"
dev-vpc-tags_all = tomap({
  "Name" = "development"
  "vpc_env" = "dev"
})
dev-vpc-tags_name = "development"
@reenagureshi /workspaces/lab9 (main) $
```

```
@reenaqureshi /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
@reenaqureshi /workspaces/lab9 (main) $ terraform state show aws_vpc.development_vpc
# aws_vpc.development_vpc:
resource "aws_vpc" "development_vpc" {
    arn                                  = "arn:aws:ec2:me-central-1:623705168110:vpc/vpc-06
42fa1474edaa73e"
    assign_generated_ipv6_cidr_block     = false
    cidr_block                           = "10.0.0.0/16"
    default_network_acl_id               = "acl-0a725df42d1cb0afa"
    default_route_table_id               = "rtb-0ad05dbc93057360d"
    default_security_group_id            = "sg-01f194951280a20c7"
    dhcp_options_id                      = "dopt-06a9132ca1eaa7ef9"
    enable_dns_hostnames                 = false
    enable_dns_support                   = true
    enable_network_address_usage_metrics = false
    id                                   = "vpc-0642fa1474edaa73e"
    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-0ad05dbc93057360d"
    owner_id                             = "623705168110"
    region                               = "me-central-1"
    tags                                 = {
        "Name"    = "development"
        "vpc_env" = "dev"
    }
    tags_all                             = {
        "Name"    = "development"
        "vpc_env" = "dev"
    }
```

Task 6 — Terraform Outputs & Attributes Reporting

1. Add outputs in main.tf:

```
output "dev-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
}
```

Expand output section in main.tf accordingly

Run terraform apply and record outputs.

```
@reenaqureshi /workspaces/lab9 (main) $ terraform apply
data.aws_vpc.existing_vpc: Reading...
aws_vpc.development_vpc: Refreshing state... [id=vpc-0642fa1474edaa73e]
data.aws_vpc.existing_vpc: Read complete after 0s [id=vpc-04a44d49b9440a0d6]
aws_subnet.dev_subnet_1_existing: Refreshing state... [id=subnet-02ed1495511fef666]
aws_subnet.dev_subnet_1: Refreshing state... [id=subnet-05735746f6ad2524f]

Changes to Outputs:
  + dev-subnet-arn        = "arn:aws:ec2:me-central-1:623705168110:subnet/subnet-05735746f6a
d2524f"
  + dev-subnet-cidr_block = "10.0.10.0/24"
  + dev-subnet-id         = "subnet-05735746f6ad2524f"
  + dev-subnet-region     = "me-central-1a"
  + dev-subnet-tags_all   = {
      + Name = "subnet-1-dev"
    }
  + dev-subnet-tags_name  = "subnet-1-dev"
  + dev-vpc-arn           = "arn:aws:ec2:me-central-1:623705168110:vpc/vpc-0642fa1474edaa73e
"
  + dev-vpc-cidr_block    = "10.0.0.0/16"
  + dev-vpc-id            = "vpc-0642fa1474edaa73e"
  + 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 real infrastructure.

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

Cleanup — Delete Resources & State Verification

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

Changes to Outputs:
  - dev-subnet-arn        = "arn:aws:ec2:me-central-1:623705168110:subnet/subnet-05735746f
d2524f" -> null
  - dev-subnet-cidr_block = "10.0.10.0/24" -> null
  - dev-subnet-id         = "subnet-05735746f6ad2524f" -> 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-1:623705168110:vpc/vpc-0642fa1474edaa7
" -> null
  - dev-vpc-cidr_block    = "10.0.0.0/16" -> null
  - dev-vpc-id            = "vpc-0642fa1474edaa73e" -> 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: Destroying... [id=subnet-05735746f6ad2524f]
aws_subnet.dev_subnet_1_existing: Destroying... [id=subnet-02ed1495511fef666]
aws_subnet.dev_subnet_1: Destruction complete after 1s
aws_vpc.development_vpc: Destroying... [id=vpc-0642fa1474edaa73e]
aws_subnet.dev_subnet_1_existing: Destruction complete after 1s
aws_vpc.development_vpc: Destruction complete after 1s

Destroy complete! Resources: 3 destroyed.
```

Inspect state files:

cat terraform.tfstate

cat terraform.tfstate.backup

```
estroy complete! Resources: 3 destroyed.
reenaqureshi /workspaces/lab9 (main) $ cat terraform.tfstate
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 29,
  "lineage": "a604d006-44c8-1626-4a7e-671a0a8d749a",
  "outputs": {},
  "resources": [],
  "check_results": null
}
reenaqureshi /workspaces/lab9 (main) $ cat terraform.tfstate.backup
{
  "version": 4,
  "terraform_version": "1.14.3",
  "serial": 24,
  "lineage": "a604d006-44c8-1626-4a7e-671a0a8d749a",
  "outputs": {
    "dev-subnet-arn": {
      "value": "arn:aws:ec2:me-central-1:623705168110:subnet/subnet-05735746f6ad2524f",
      "type": "string"
    },
    "dev-subnet-cidr_block": {
      "value": "10.0.10.0/24",
      "type": "string"
    },
    "dev-subnet-id": {
      "value": "subnet-05735746f6ad2524f",
      "type": "string"
    },
    "dev-subnet-region": {
      "value": "me-central-1a",
      "type": "string"
```