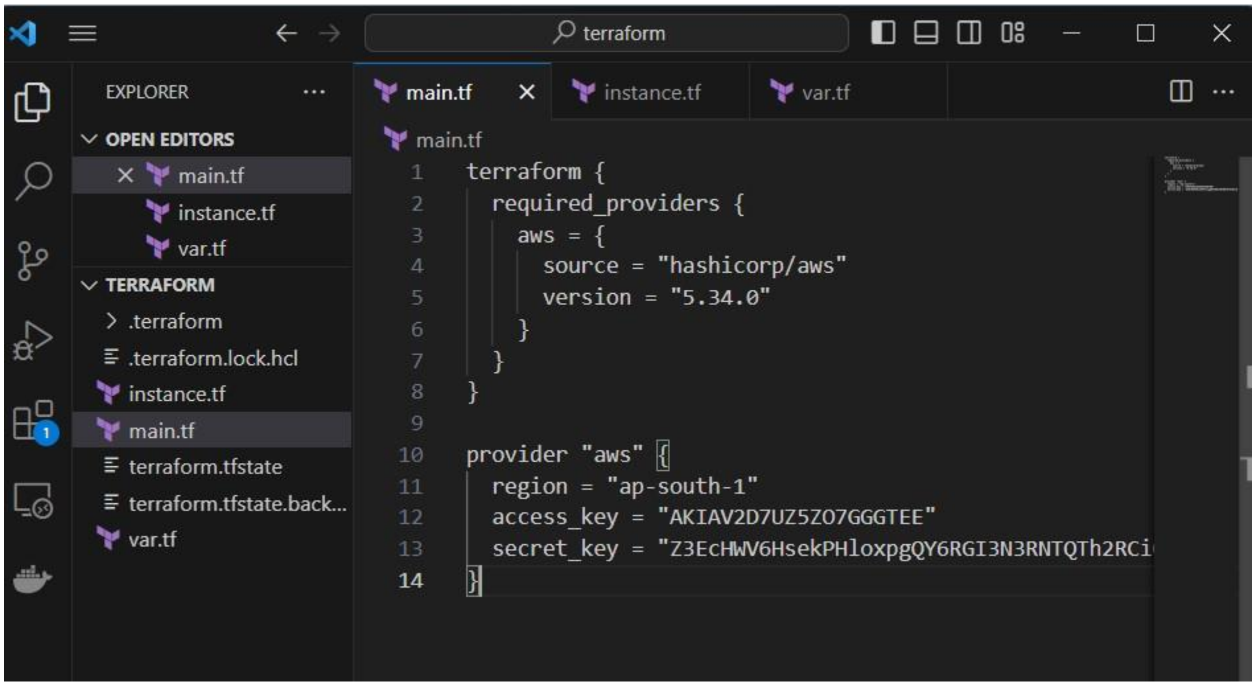
LAB-6

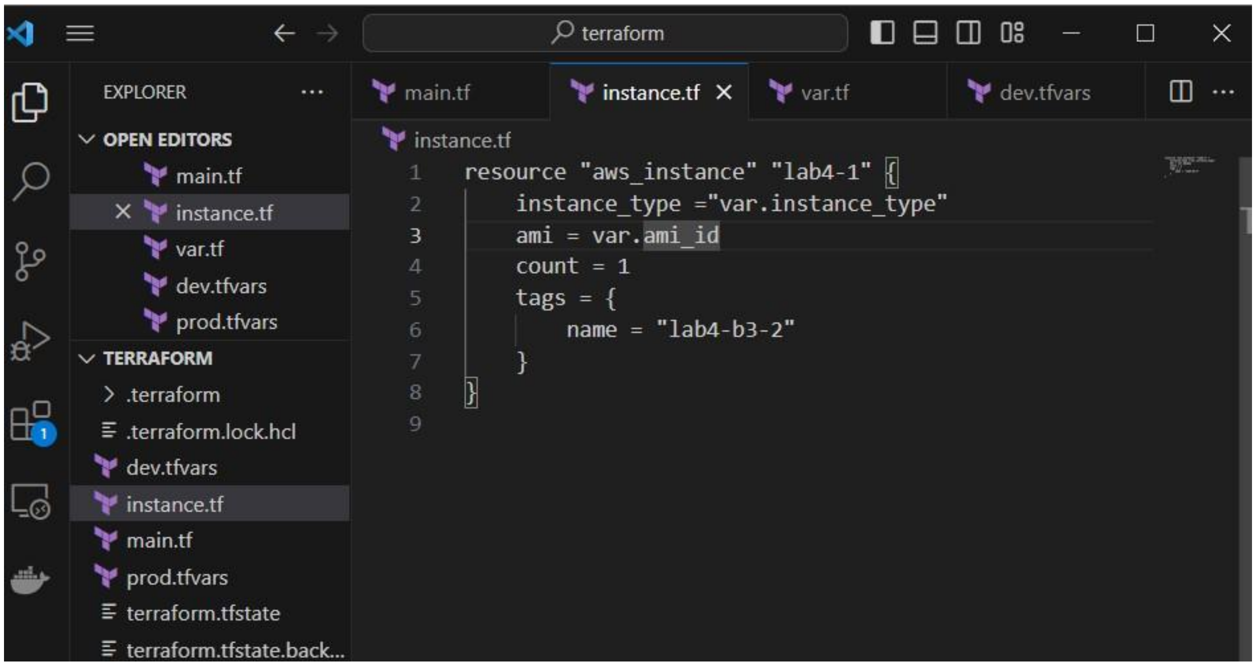
Terraform Multiple tfvars Files

Step 1: Create dev.tfvars and prod.tfvars



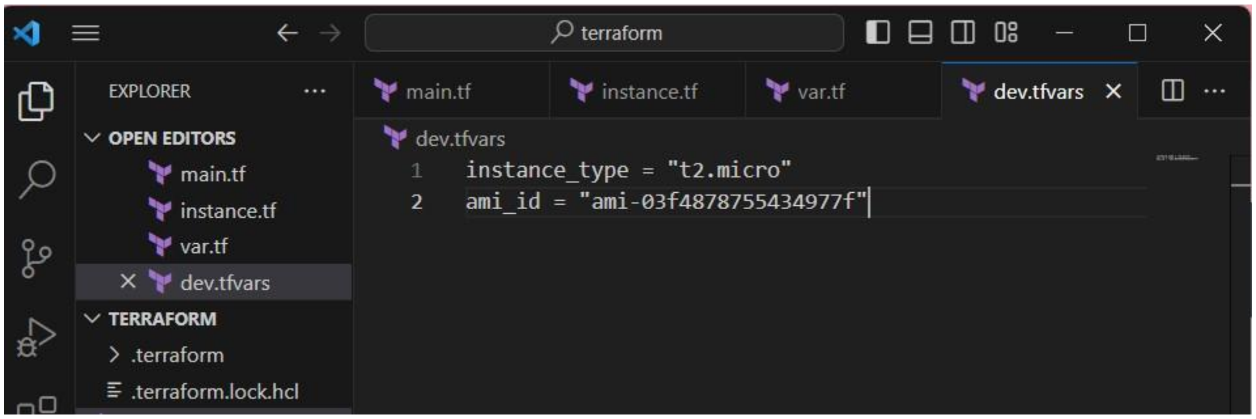
main.tf

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.34.0"
    }
  }
}

provider "aws" {
  region = "ap-south-1"
  access_key = "AKIAV2D7UZ5ZO7GGGTEE"
  secret_key = "Z3EcHWV6HsekPHloxpgQY6RGI3N3RNTQTh2RCi
}
```



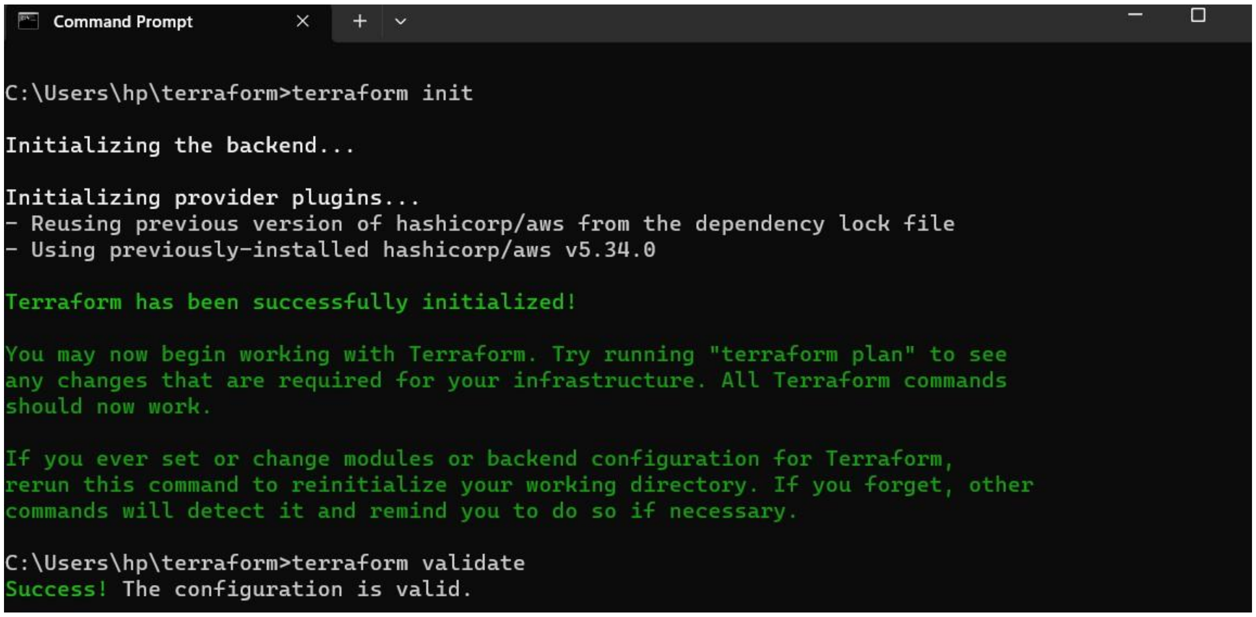
instance.tf

```
resource "aws_instance" "lab4-1" {
    instance_type ="var.instance_type"
    ami = var.ami_id
    count = 1
    tags = {
        name = "lab4-b3-2"
    }
}
```



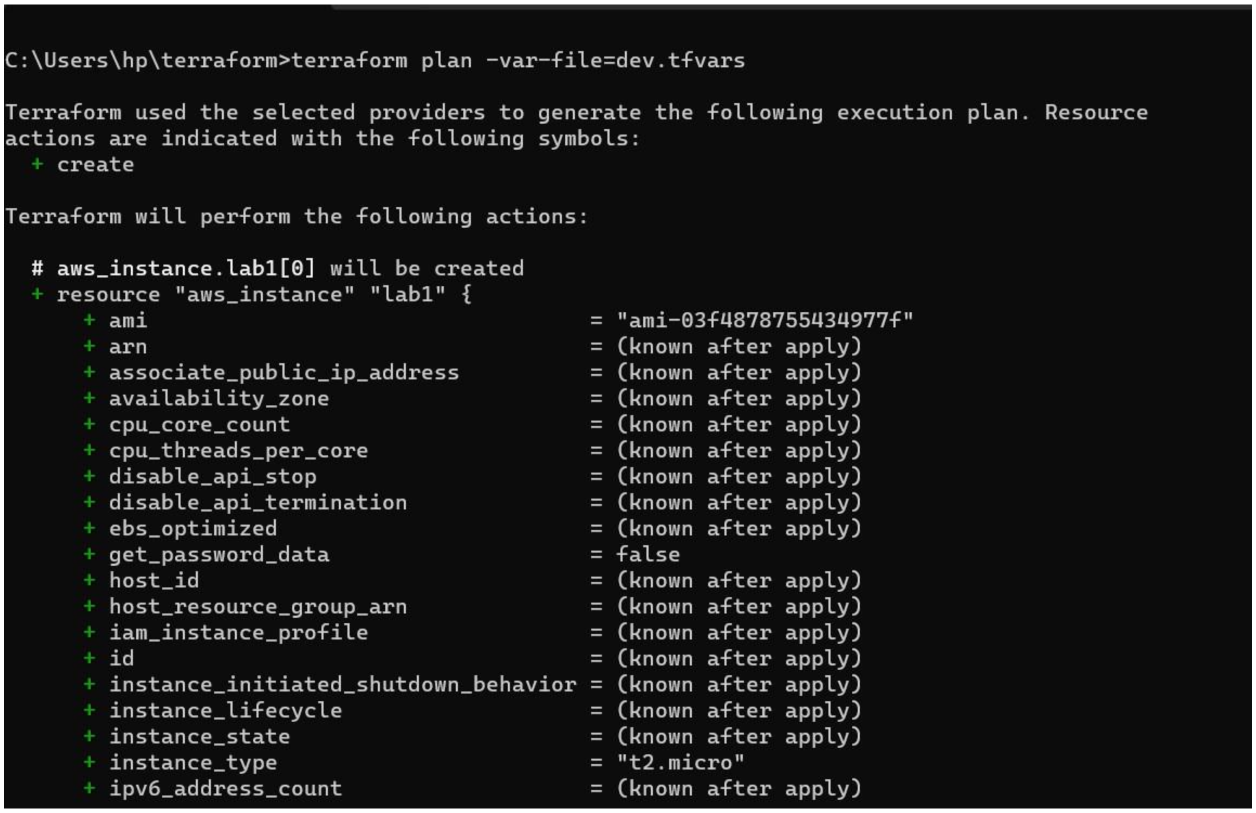
```
instance_type = "t2.micro"
ami_id = "ami-03f4878755434977f"
```

Step 2: Now run terraform cycle

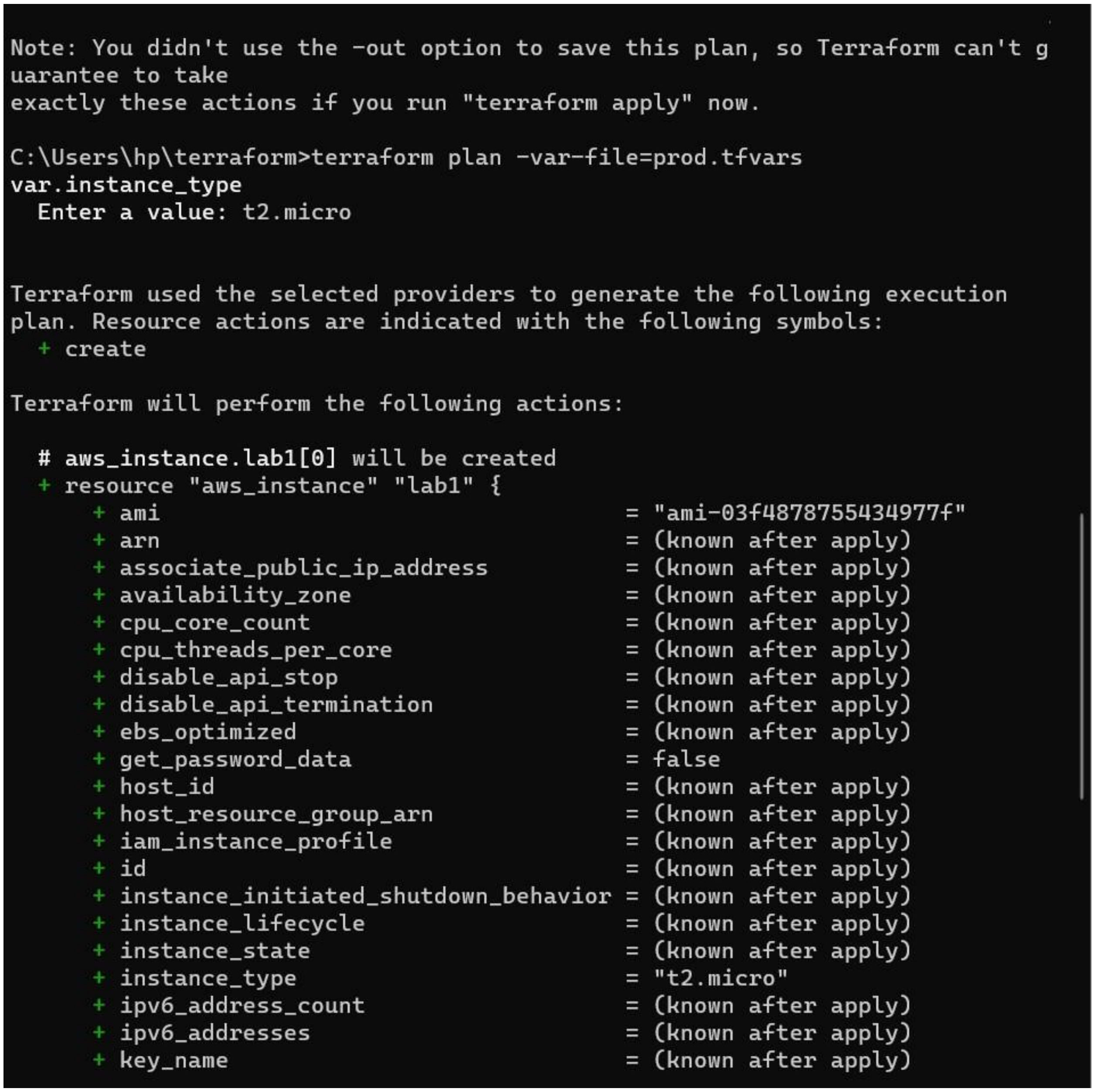


```
C:\Users\hp\terraform>terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.34.0

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.

C:\Users\hp\terraform>terraform validate
Success! The configuration is valid.
```

Step 3: To run terraform plan we need to use -var-file=dev.tfvars or -var-file=prod.tfvars

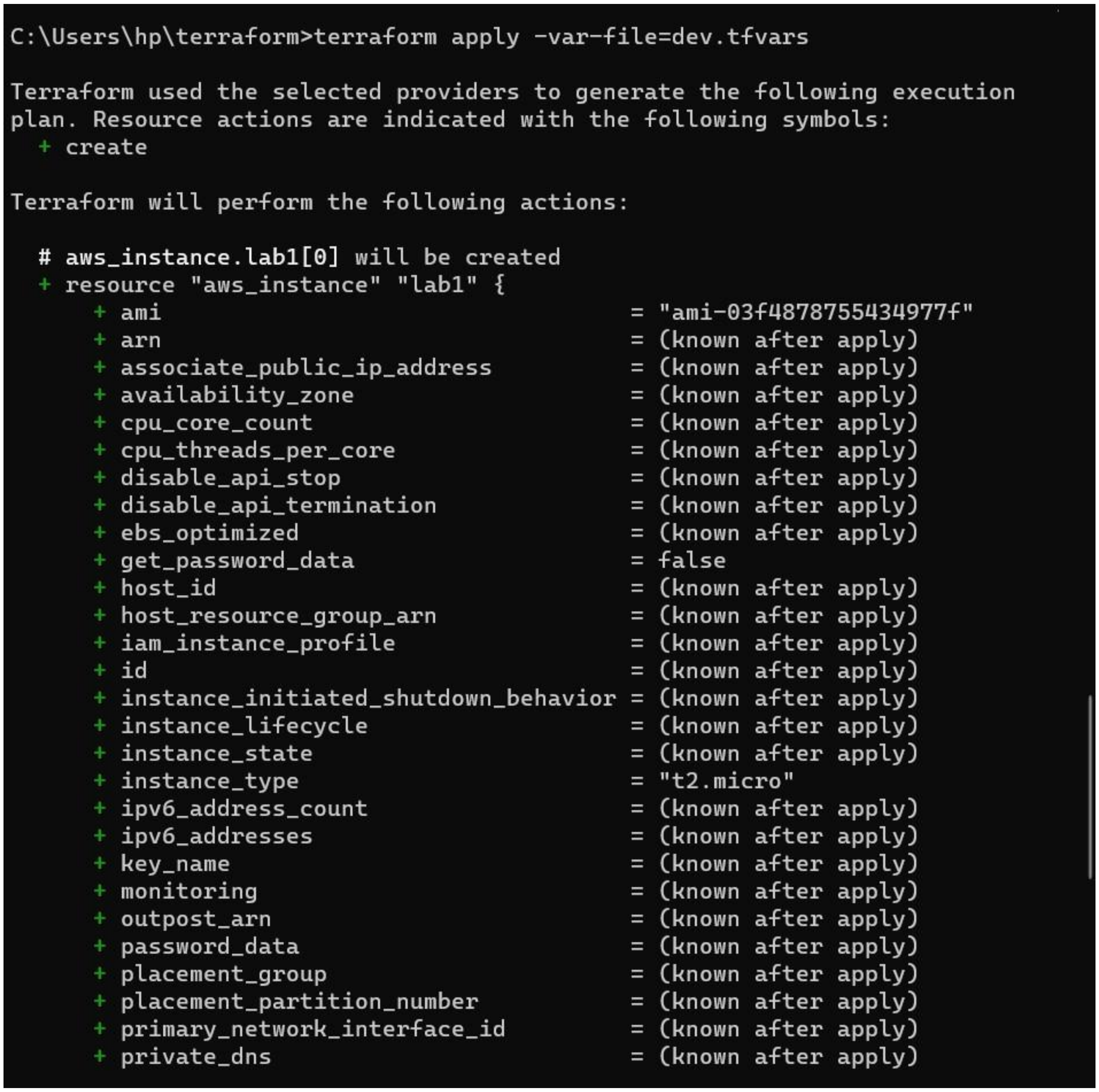


```
C:\Users\hp\terraform>terraform plan -var-file=dev.tfvars

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_instance.lab1[0] will be created
  + resource "aws_instance" "lab1" {
      + ami                                  = "ami-03f4878755434977f"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
```

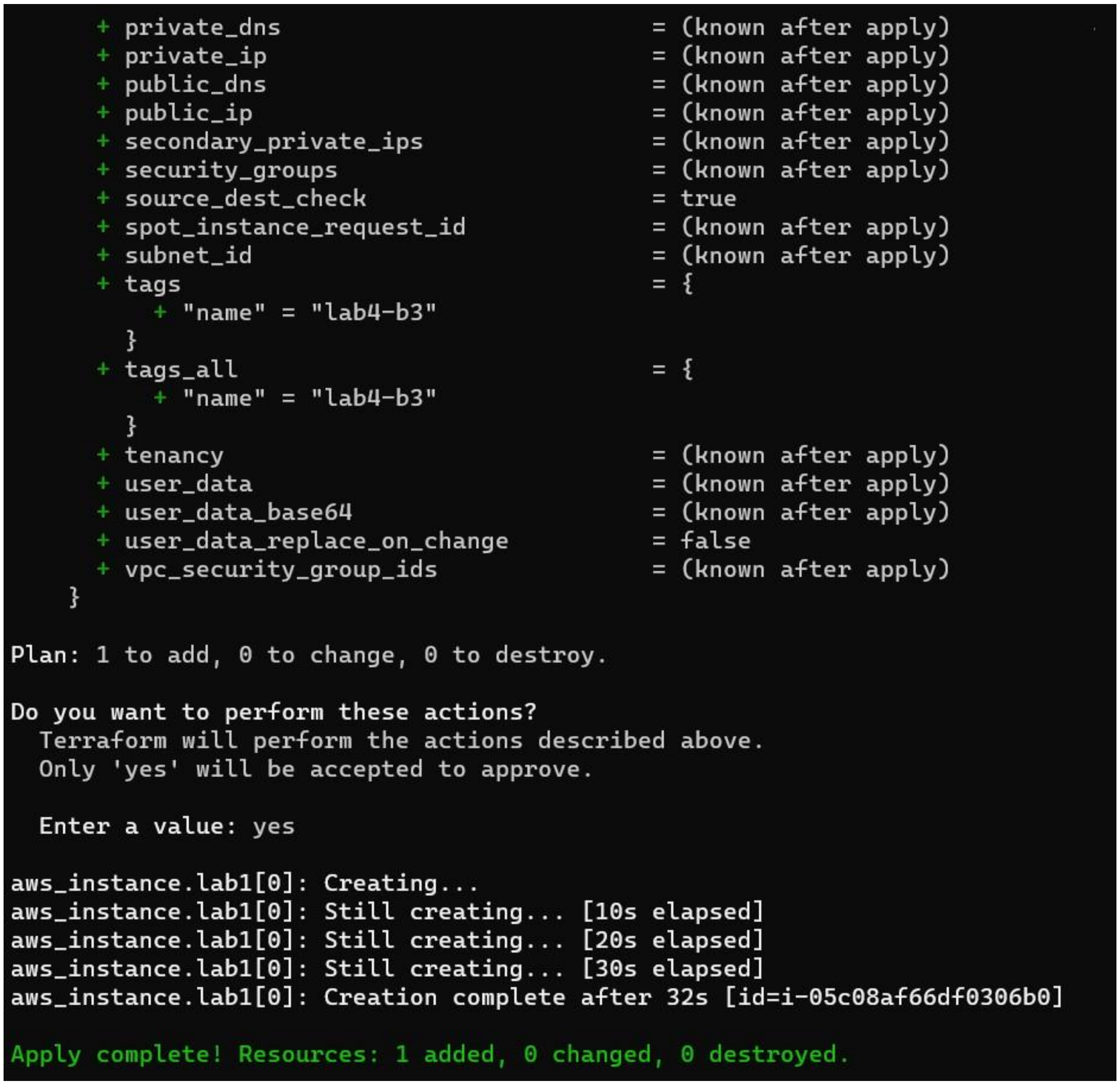


```
Note: You didn't use the -out option to save this plan, so Terraform can't g
uarantee to take
exactly these actions if you run "terraform apply" now.

C:\Users\hp\terraform>terraform plan -var-file=prod.tfvars
var.instance_type
  Enter a value: t2.micro


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_instance.lab1[0] will be created
  + resource "aws_instance" "lab1" {
      + ami                                  = "ami-03f4878755434977f"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
```

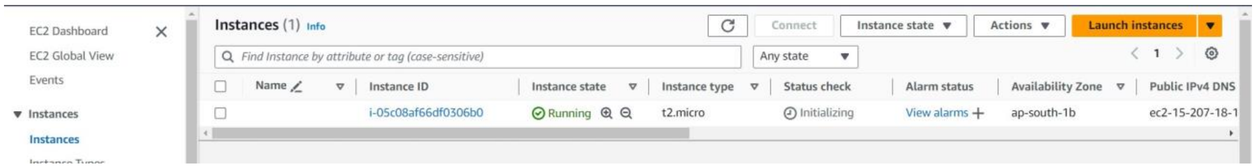
Step 4: To run terraform apply and destroy we need to use -var-file=dev.tfvars or -var-file=prod.tfvars



```
C:\Users\hp\terraform>terraform apply -var-file=dev.tfvars

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_instance.lab1[0] will be created
  + resource "aws_instance" "lab1" {
      + ami                                  = "ami-03f4878755434977f"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.micro"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = (known after apply)
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
```

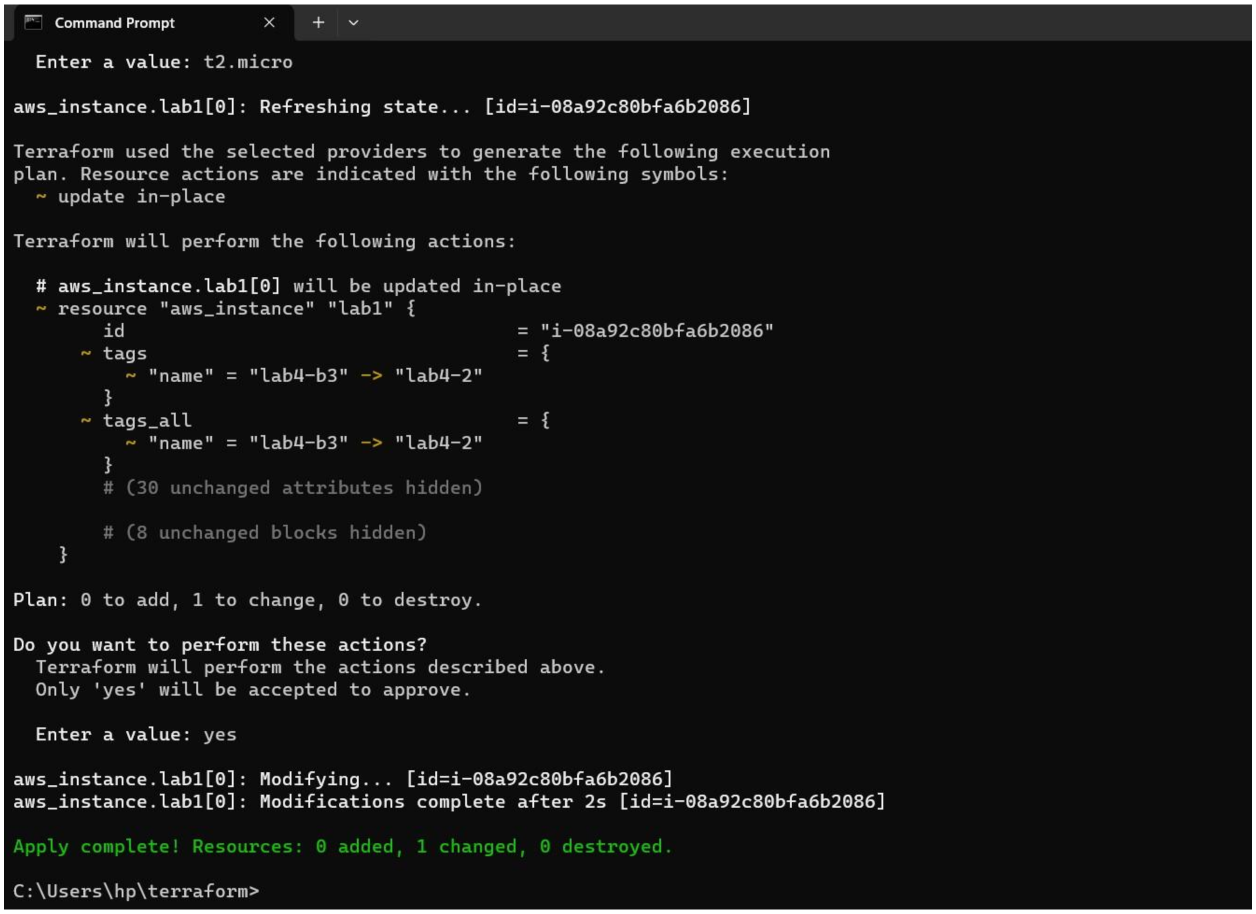


```
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
      + tags                                 = {
          + "name" = "lab4-b3"
        }
      + tags_all                             = {
          + "name" = "lab4-b3"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)
    }

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_instance.lab1[0]: Creating...
aws_instance.lab1[0]: Still creating... [10s elapsed]
aws_instance.lab1[0]: Still creating... [20s elapsed]
aws_instance.lab1[0]: Still creating... [30s elapsed]
aws_instance.lab1[0]: Creation complete after 32s [id=i-05c08af66df0306b0]

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

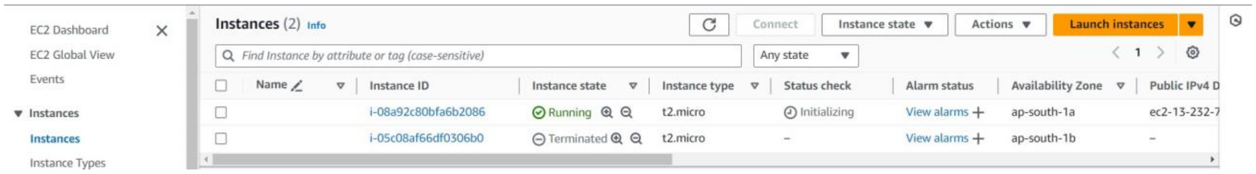


Instances (1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	i-05c08af66df0306b0	Running	t2.micro	Initializing	View alarms	ap-south-1b	ec2-15-207-18-1

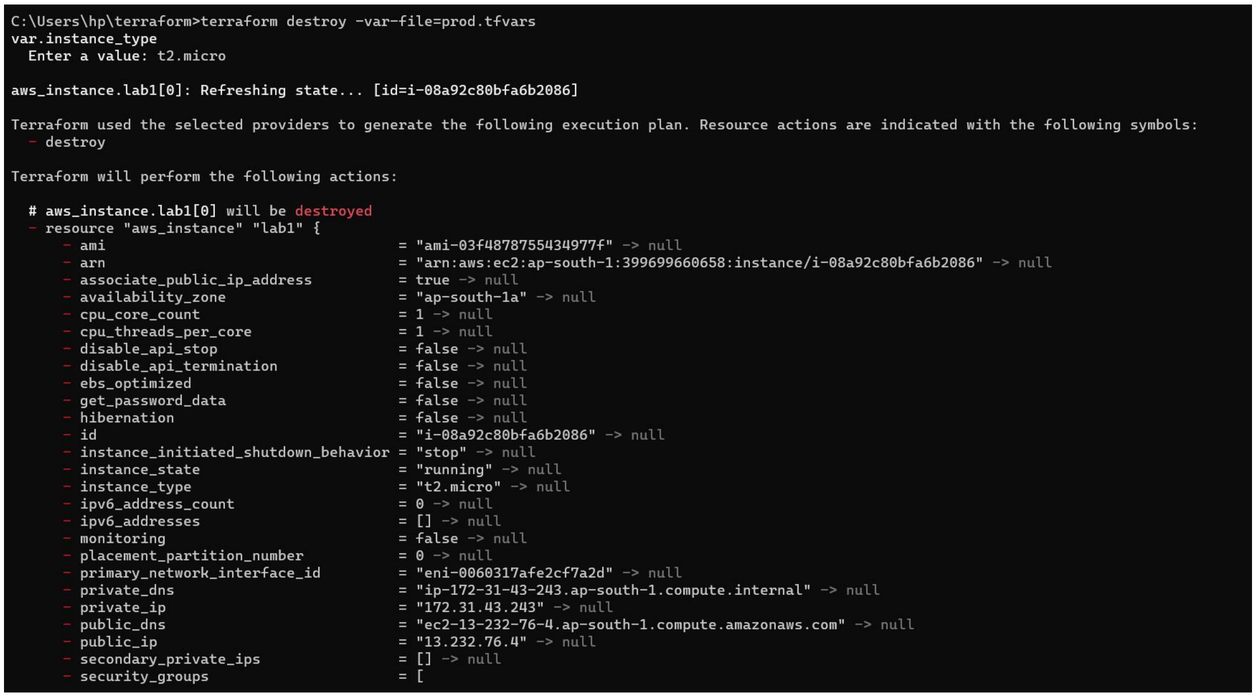


```
  Enter a value: t2.micro

aws_instance.lab1[0]: Refreshing state... [id=i-08a92c80bfa6b2086]

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_instance.lab1[0] will be updated in-place
  ~ resource "aws_instance" "lab1" {
        id                                   = "i-08a92c80bfa6b2086"
      ~ tags                                 = {
          ~ "name" = "lab4-b3" -> "lab4-2"
        }
      ~ tags_all                             = {
          ~ "name" = "lab4-b3" -> "lab4-2"
        }
        # (30 unchanged attributes hidden)

        # (8 unchanged blocks 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_instance.lab1[0]: Modifying... [id=i-08a92c80bfa6b2086]
aws_instance.lab1[0]: Modifications complete after 2s [id=i-08a92c80bfa6b2086]

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

C:\Users\hp\terraform>
```

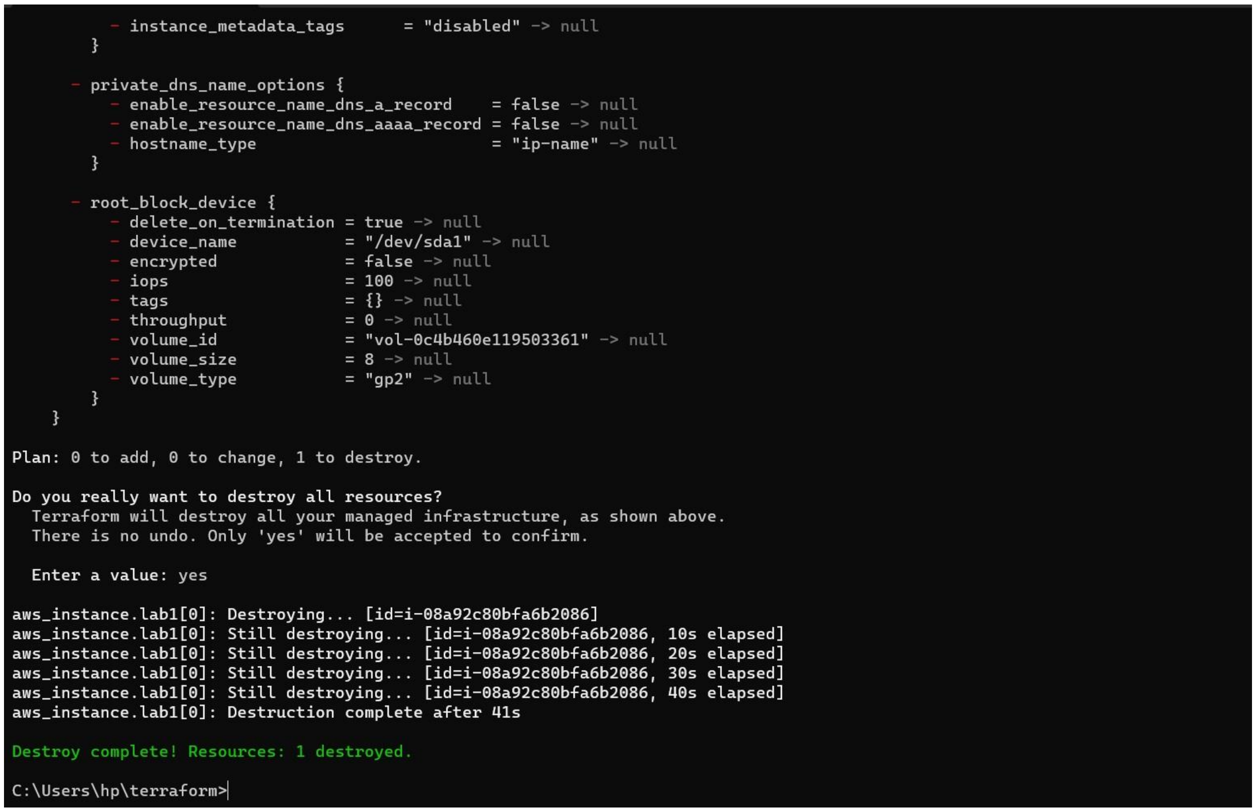


Instances (2) Info

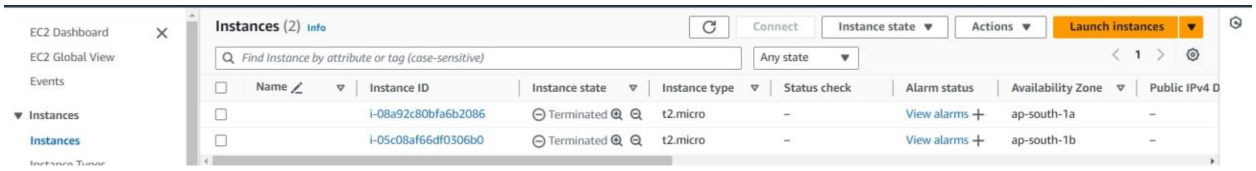
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
	i-08a92c80bfa6b2086	Running	t2.micro	Initializing	View alarms	ap-south-1a	ec2-13-232-7
	i-05c08af66df0306b0	Terminated	t2.micro	–	View alarms	ap-south-1b	–



```
C:\Users\hp\terraform>terraform destroy -var-file=prod.tfvars
var.instance_type
  Enter a value: t2.micro

aws_instance.lab1[0]: Refreshing state... [id=i-08a92c80bfa6b2086]

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_instance.lab1[0] will be destroyed
  - resource "aws_instance" "lab1" {
      - ami                                  = "ami-03f4878755434977f" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:399699660658:instance/i-08a92c80bfa6b2086" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1a" -> null
      - cpu_core_count                       = 1 -> null
      - cpu_threads_per_core                 = 1 -> null
      - disable_api_stop                     = false -> null
      - disable_api_termination              = false -> null
      - ebs_optimized                        = false -> null
      - get_password_data                    = false -> null
      - hibernation                          = false -> null
      - id                                   = "i-08a92c80bfa6b2086" -> null
      - instance_initiated_shutdown_behavior = "stop" -> null
      - instance_state                       = "running" -> null
      - instance_type                        = "t2.micro" -> null
      - ipv6_address_count                   = 0 -> null
      - ipv6_addresses                       = [] -> null
      - monitoring                           = false -> null
      - placement_partition_number           = 0 -> null
      - primary_network_interface_id         = "eni-0060317afe2cf7a2d" -> null
      - private_dns                          = "ip-172-31-43-243.ap-south-1.compute.internal" -> null
      - private_ip                           = "172.31.43.243" -> null
      - public_dns                           = "ec2-13-232-76-4.ap-south-1.compute.amazonaws.com" -> null
      - public_ip                            = "13.232.76.4" -> null
      - secondary_private_ips                = [] -> null
      - security_groups                      = [
```



```
          - instance_metadata_tags      = "disabled" -> null
        }

      - private_dns_name_options {
          - enable_resource_name_dns_a_record    = false -> null
          - enable_resource_name_dns_aaaa_record = false -> null
          - hostname_type                        = "ip-name" -> null
        }

      - root_block_device {
          - delete_on_termination = true -> null
          - device_name           = "/dev/sda1" -> null
          - encrypted             = false -> null
          - iops                  = 100 -> null
          - tags                  = {} -> null
          - throughput            = 0 -> null
          - volume_id             = "vol-0c4b460e119503361" -> null
          - volume_size           = 8 -> null
          - volume_type           = "gp2" -> null
        }
    }

Plan: 0 to add, 0 to change, 1 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_instance.lab1[0]: Destroying... [id=i-08a92c80bfa6b2086]
aws_instance.lab1[0]: Still destroying... [id=i-08a92c80bfa6b2086, 10s elapsed]
aws_instance.lab1[0]: Still destroying... [id=i-08a92c80bfa6b2086, 20s elapsed]
aws_instance.lab1[0]: Still destroying... [id=i-08a92c80bfa6b2086, 30s elapsed]
aws_instance.lab1[0]: Still destroying... [id=i-08a92c80bfa6b2086, 40s elapsed]
aws_instance.lab1[0]: Destruction complete after 41s

Destroy complete! Resources: 1 destroyed.

C:\Users\hp\terraform>
```



Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
	i-08a92c80bfa6b2086	Terminated	t2.micro	–	View alarms +	ap-south-1a	–
	i-05c08af66df0306b0	Terminated	t2.micro	–	View alarms +	ap-south-1b	–