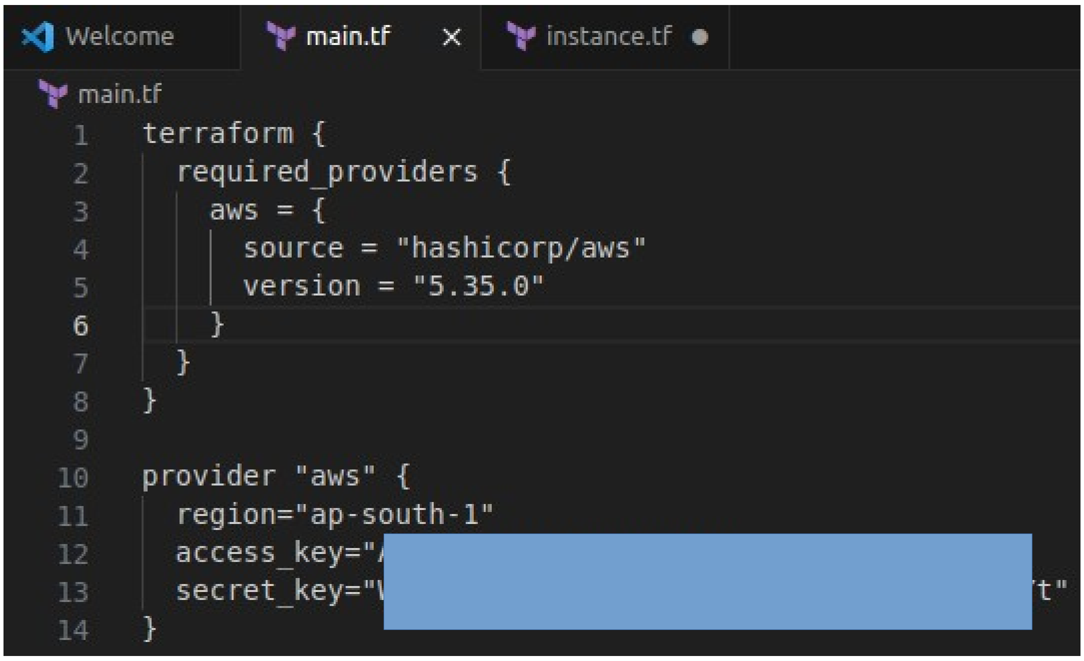
LAB-4
Terraform Variable

We will see different ways to declare variable in terraform

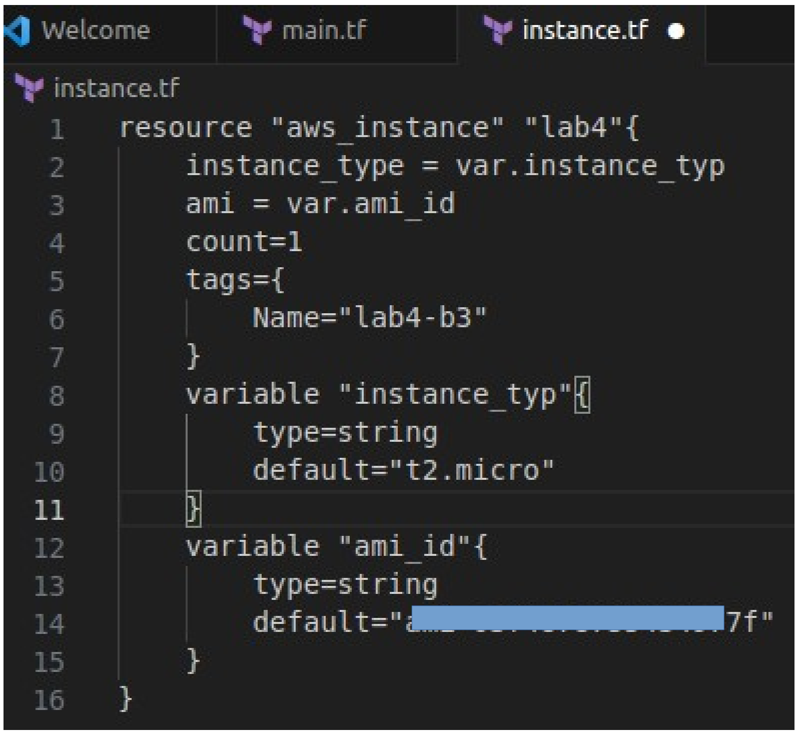
Step 1: First we will see declaring variable in instance.tf file



Welcome | main.tf | instance.tf

main.tf

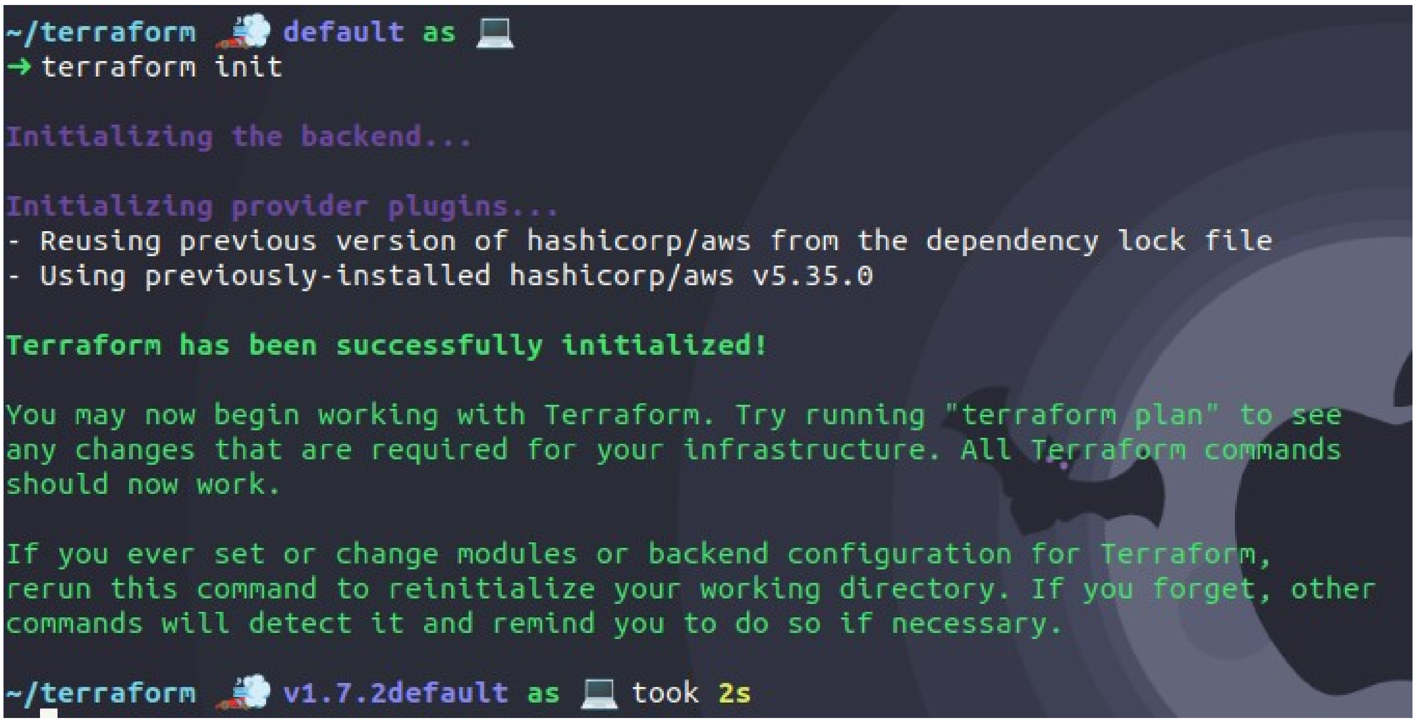
```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.35.0"
    }
  }
}

provider "aws" {
  region="ap-south-1"
  access_key="[redacted]
  secret_key="[redacted]t"
}
```



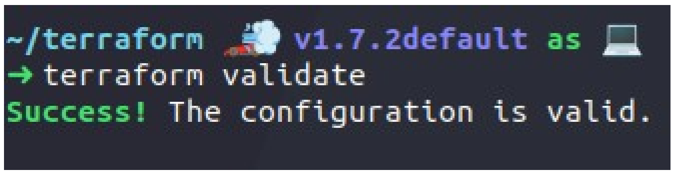
Welcome | main.tf | instance.tf

instance.tf

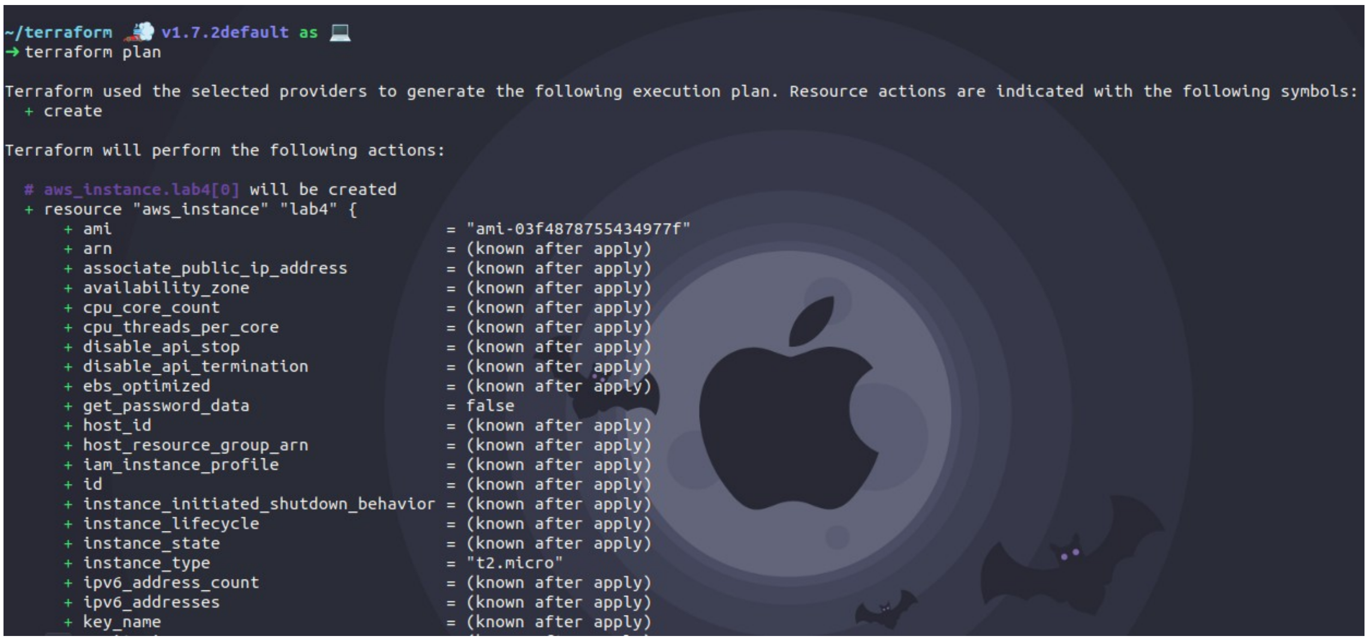
```
resource "aws_instance" "lab4"{
    instance_type = var.instance_typ
    ami = var.ami_id
    count=1
    tags={
        Name="lab4-b3"
    }
    variable "instance_typ"{
        type=string
        default="t2.micro"
    }
    variable "ami_id"{
        type=string
        default="a[redacted]7f"
    }
}
```



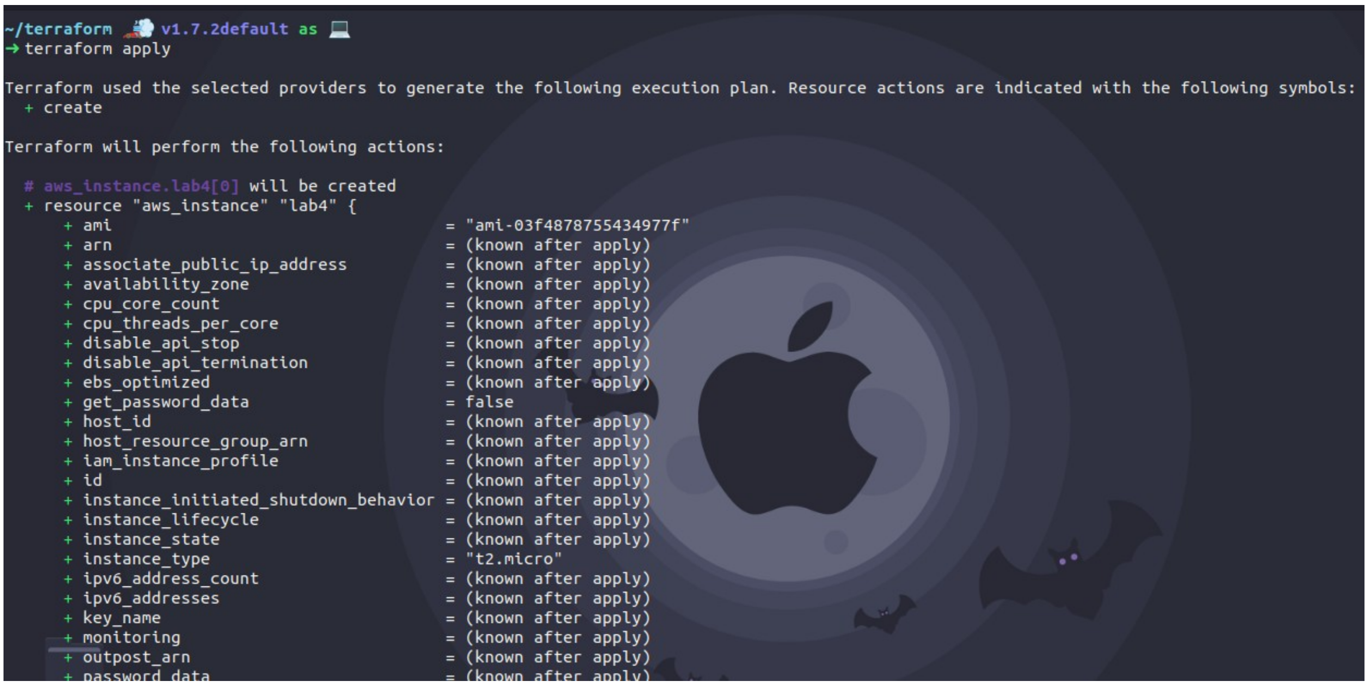
```
~/terraform 🚂 default as 💻
➜ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.35.0

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.

~/terraform 🚂 v1.7.2default as 💻 took 2s
```



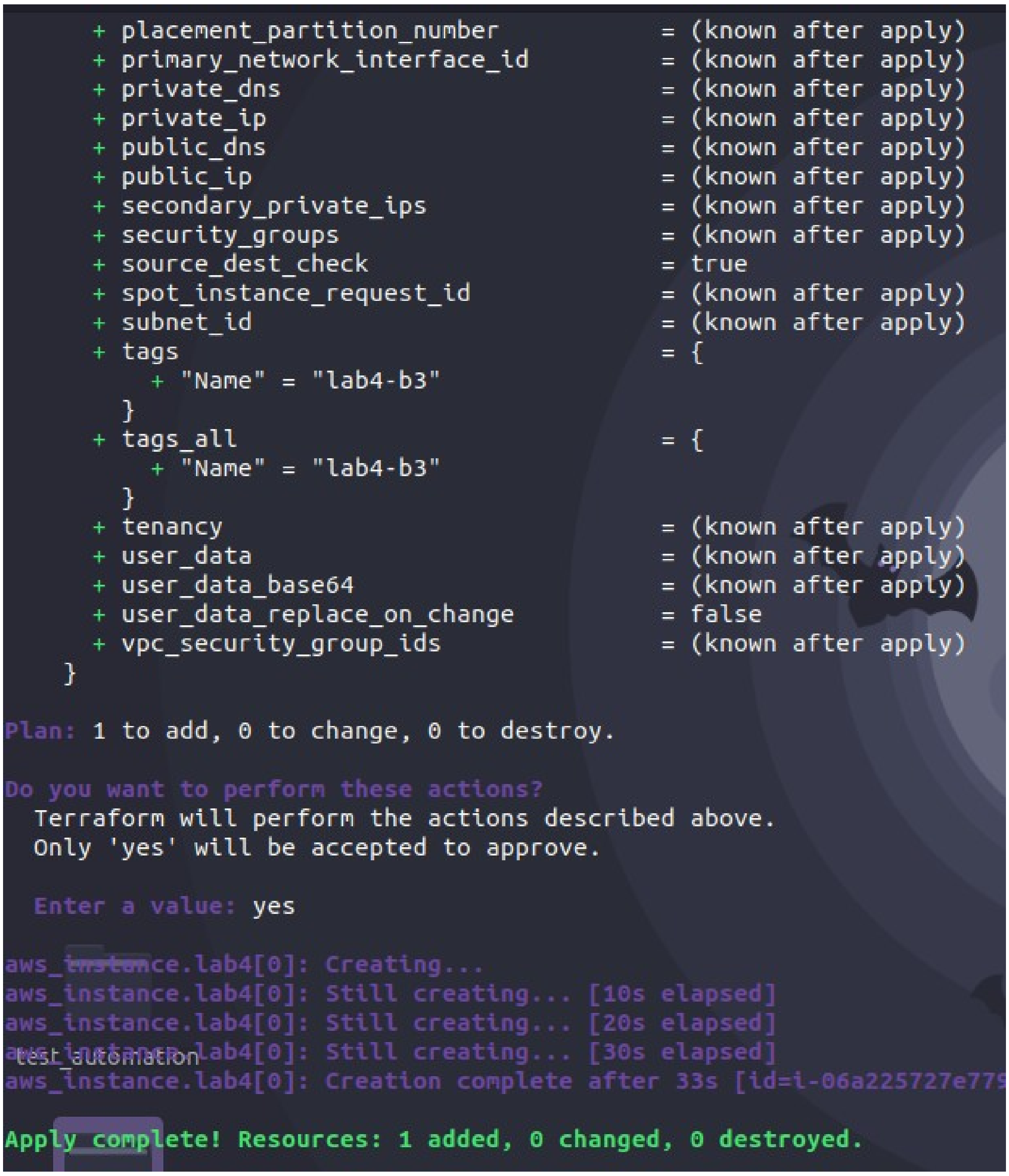
```
~/terraform 🚂 v1.7.2default as 💻
➜ terraform validate
Success! The configuration is valid.
```



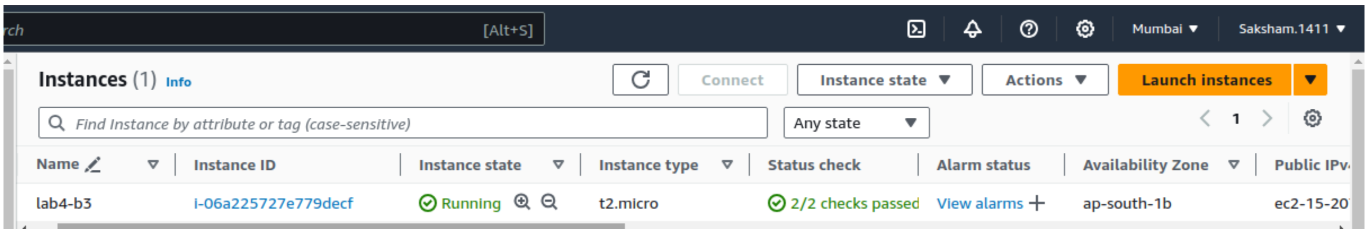
```
~/terraform 🚂 v1.7.2default as 💻
➜ terraform plan

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.lab4[0] will be created
  + resource "aws_instance" "lab4" {
      + 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)
```



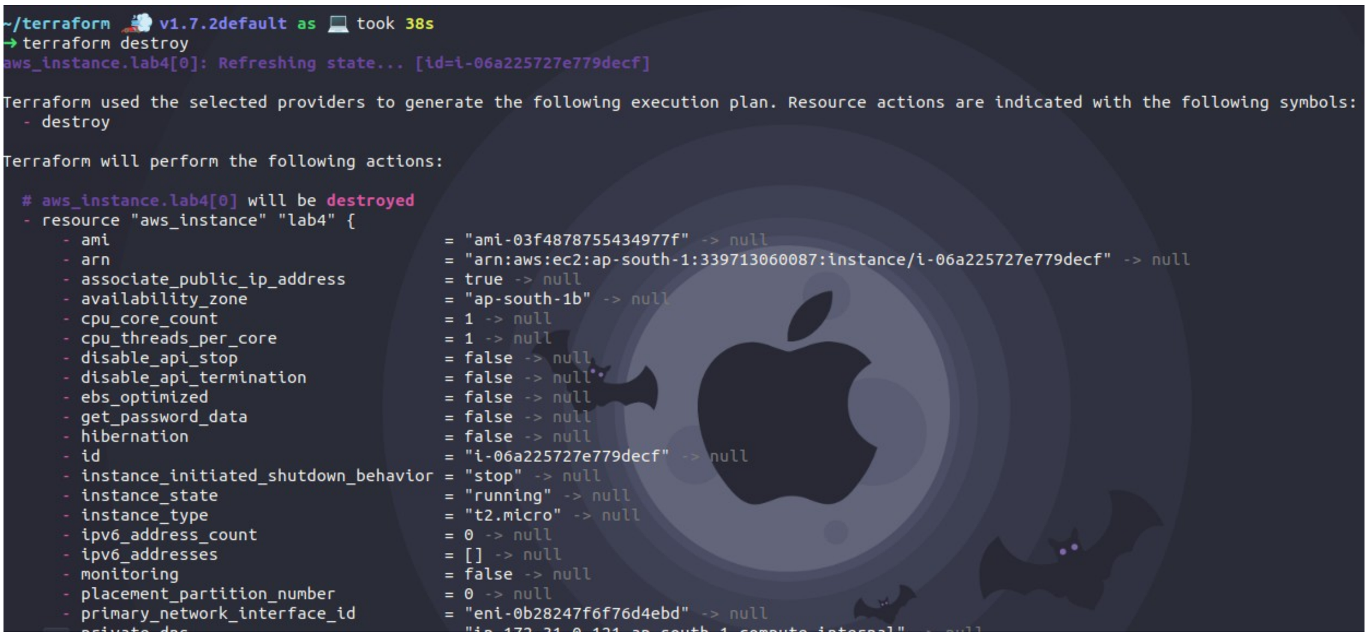
```
~/terraform  v1.7.2default as
→ terraform apply

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_instance.lab4[0] will be created
  + resource "aws_instance" "lab4" {
      + 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)
```



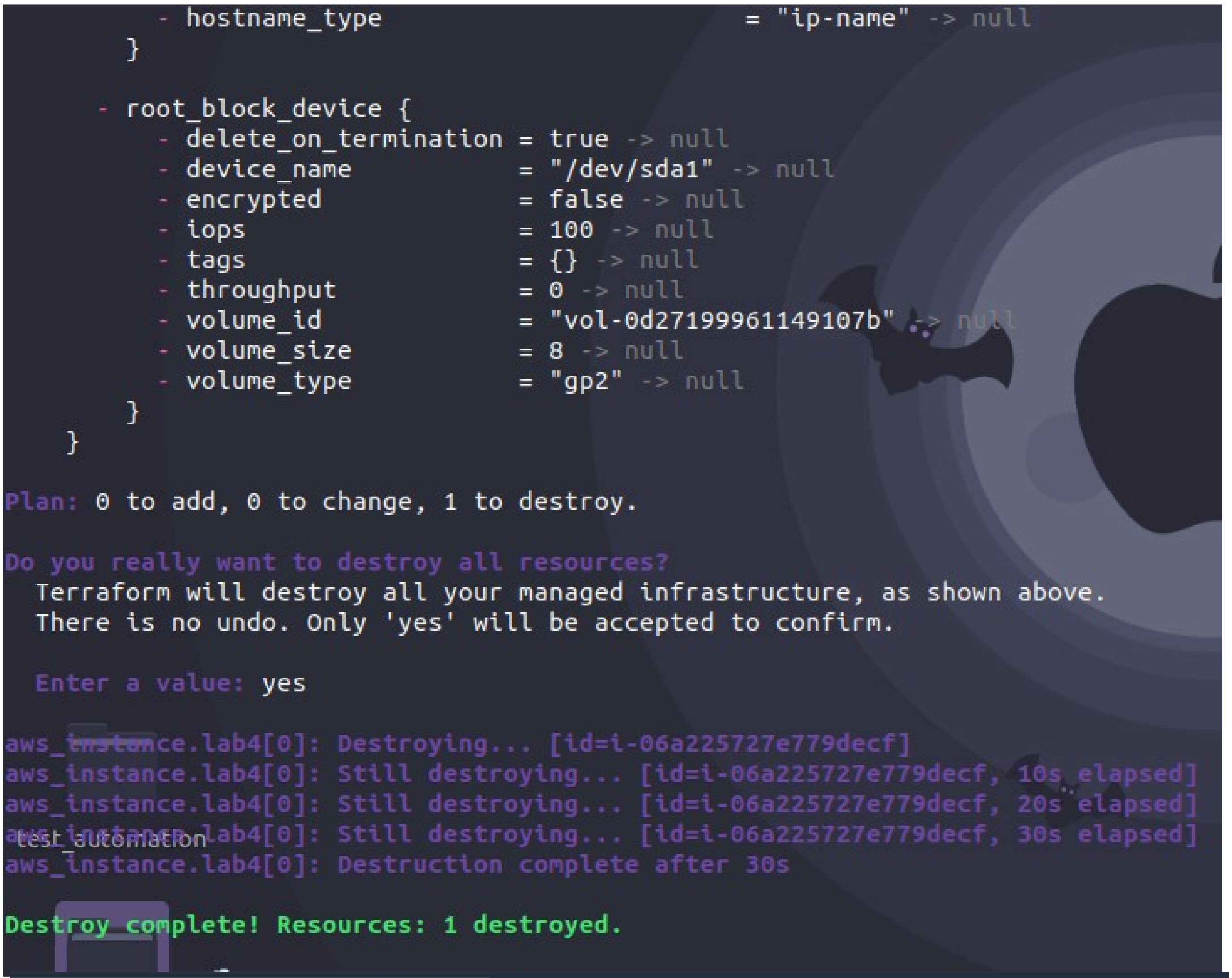
```
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (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.lab4[0]: Creating...
aws_instance.lab4[0]: Still creating... [10s elapsed]
aws_instance.lab4[0]: Still creating... [20s elapsed]
aws_instance.lab4[0]: Still creating... [30s elapsed]
aws_instance.lab4[0]: Creation complete after 33s [id=i-06a225727e779

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



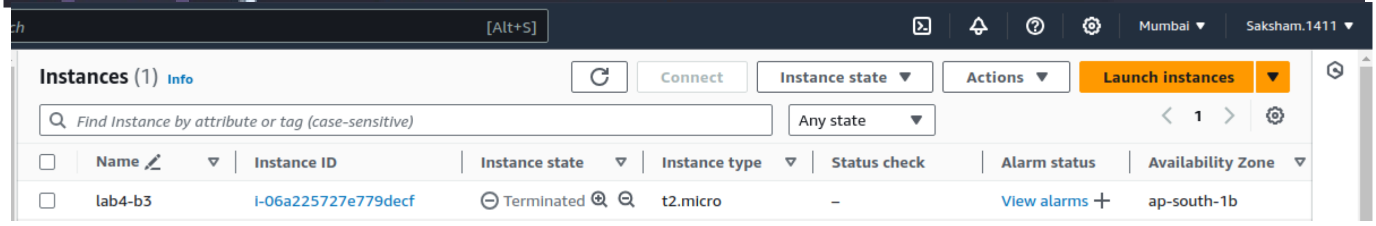
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
lab4-b3	i-06a225727e779decf	✓ Running	t2.micro	✓ 2/2 checks passed	View alarms +	ap-south-1b	ec2-15-20



```
~/terraform 💨 v1.7.2default as 💻 took 38s
→ terraform destroy
aws_instance.lab4[0]: Refreshing state... [id=i-06a225727e779decf]

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.lab4[0] will be destroyed
  - resource "aws_instance" "lab4" {
      - ami                                  = "ami-03f4878755434977f" -> null
      - arn                                  = "arn:aws:ec2:ap-south-1:339713060087:instance/i-06a225727e779decf" -> null
      - associate_public_ip_address          = true -> null
      - availability_zone                    = "ap-south-1b" -> 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-06a225727e779decf" -> 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-0b28247f6f76d4ebd" -> null
```

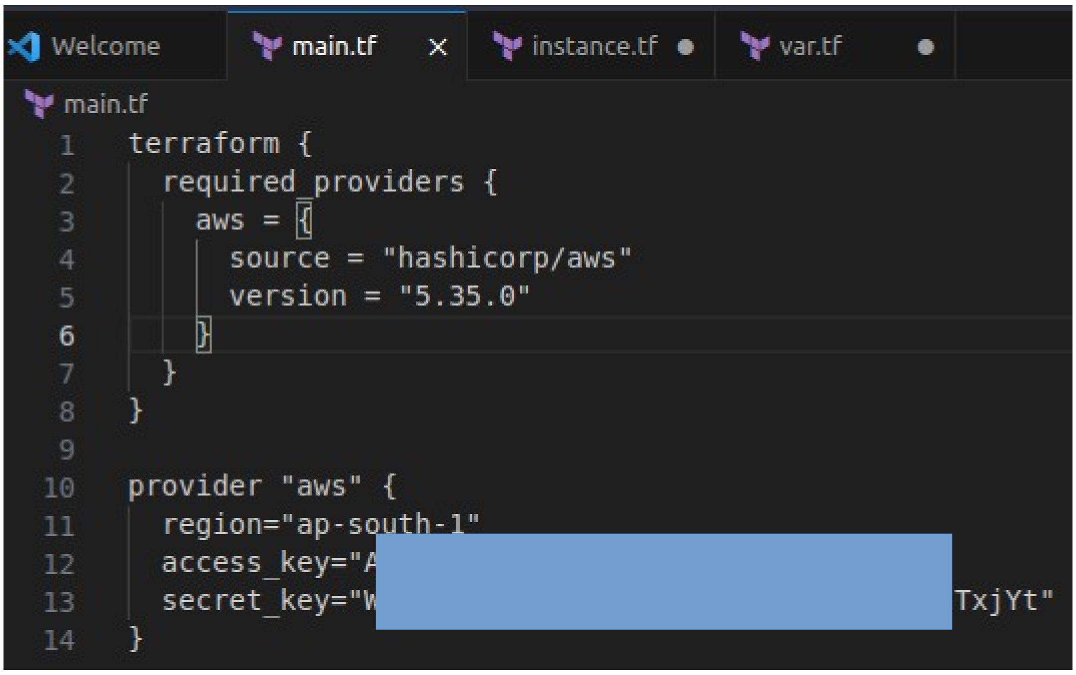


```
          - hostname_type                                = "ip-name" -> null
        }

      - 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-0d27199961149107b" -> 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.lab4[0]: Destroying... [id=i-06a225727e779decf]
aws_instance.lab4[0]: Still destroying... [id=i-06a225727e779decf, 10s elapsed]
aws_instance.lab4[0]: Still destroying... [id=i-06a225727e779decf, 20s elapsed]
aws_instance.lab4[0]: Still destroying... [id=i-06a225727e779decf, 30s elapsed]
aws_instance.lab4[0]: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.
```

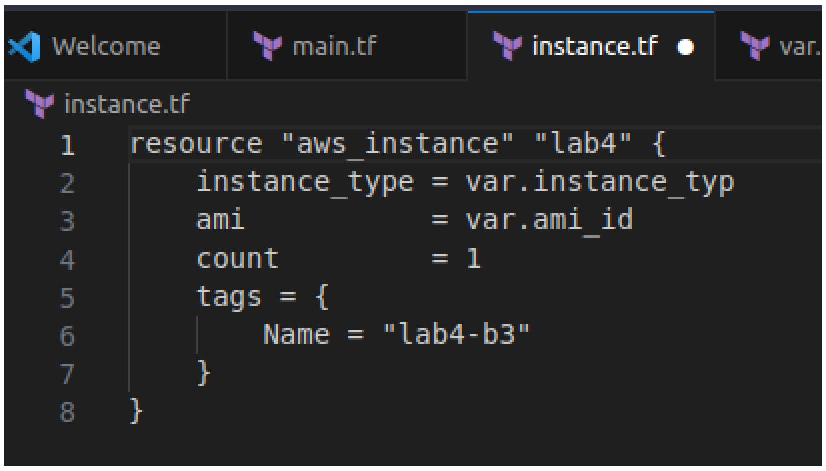


Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
lab4-b3	i-06a225727e779decf	Terminated	t2.micro	–	View alarms +	ap-south-1b

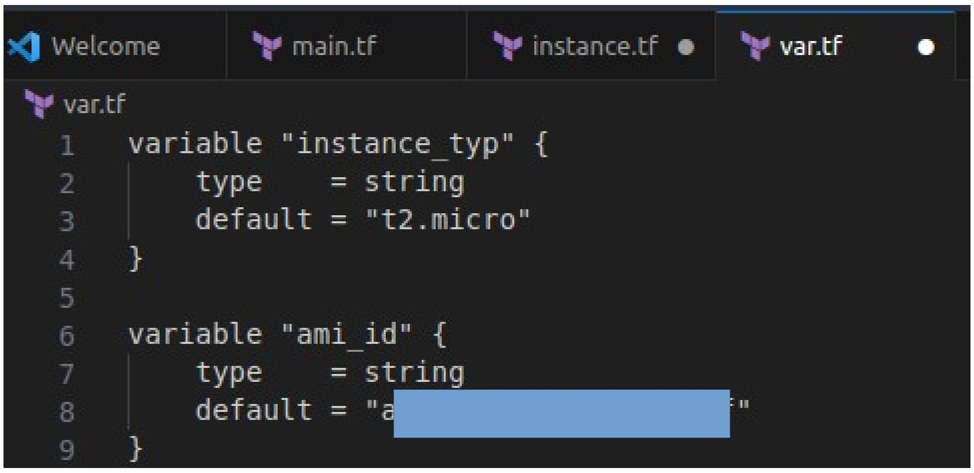
Step 2: Now we will to create a var.tf file to create variable



```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.35.0"
    }
  }
}

provider "aws" {
  region="ap-south-1"
  access_key="A[redacted]
  secret_key="W[redacted]TxjYt"
}
```



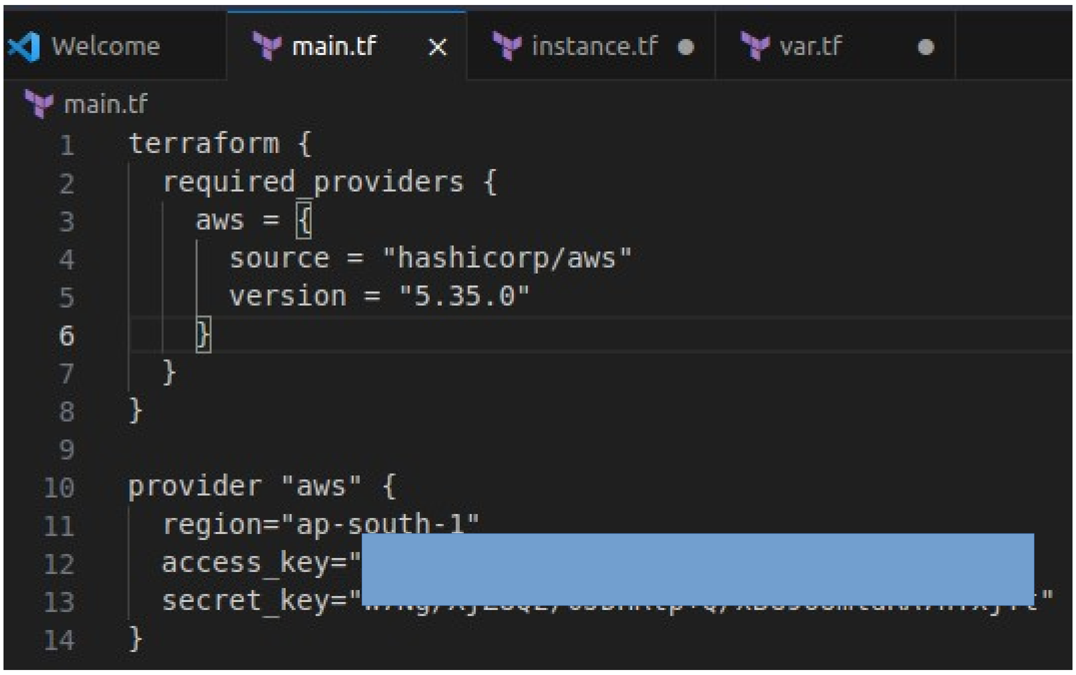
```
resource "aws_instance" "lab4" {
    instance_type = var.instance_typ
    ami           = var.ami_id
    count         = 1
    tags = {
        Name = "lab4-b3"
    }
}
```



```
variable "instance_typ" {
    type    = string
    default = "t2.micro"
}

variable "ami_id" {
    type    = string
    default = "a[redacted]"
}
```

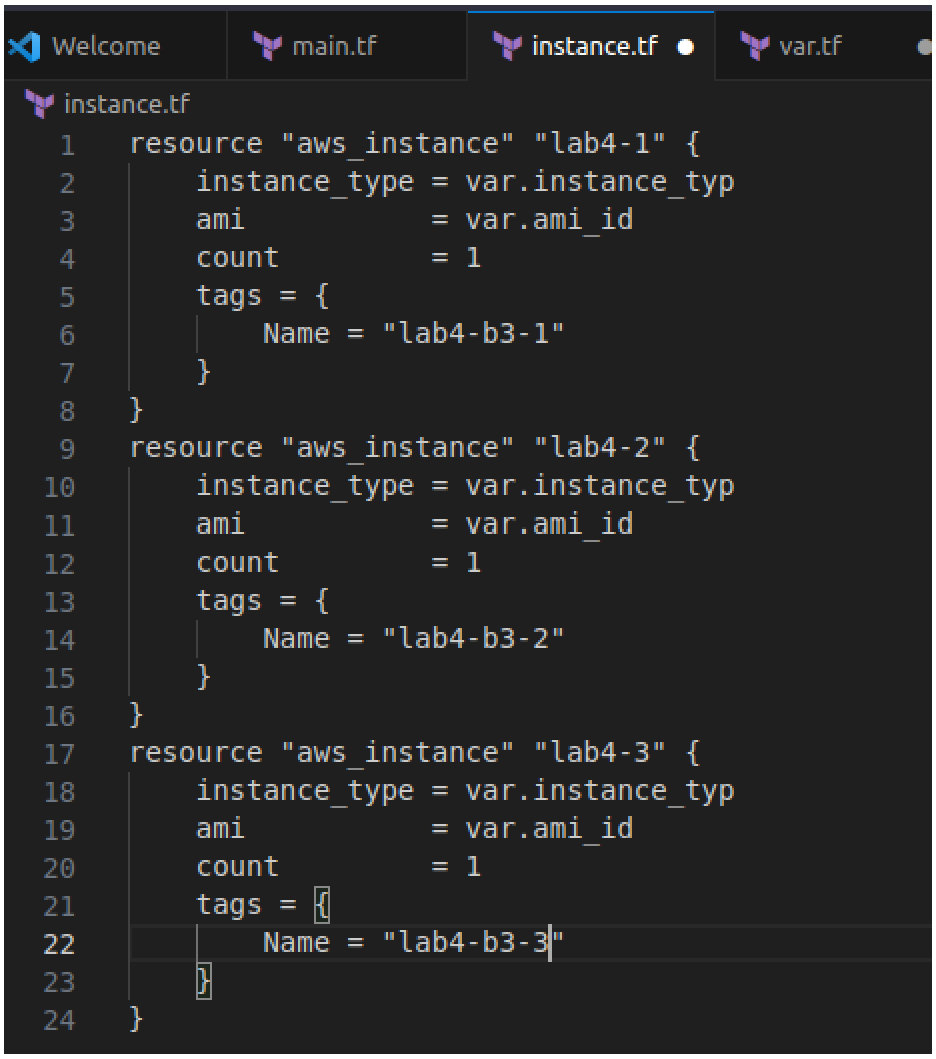
Now by again running the terraform plan and terraform apply instance will be created.

Step 3: To create multiple instances by changing instance.tf file



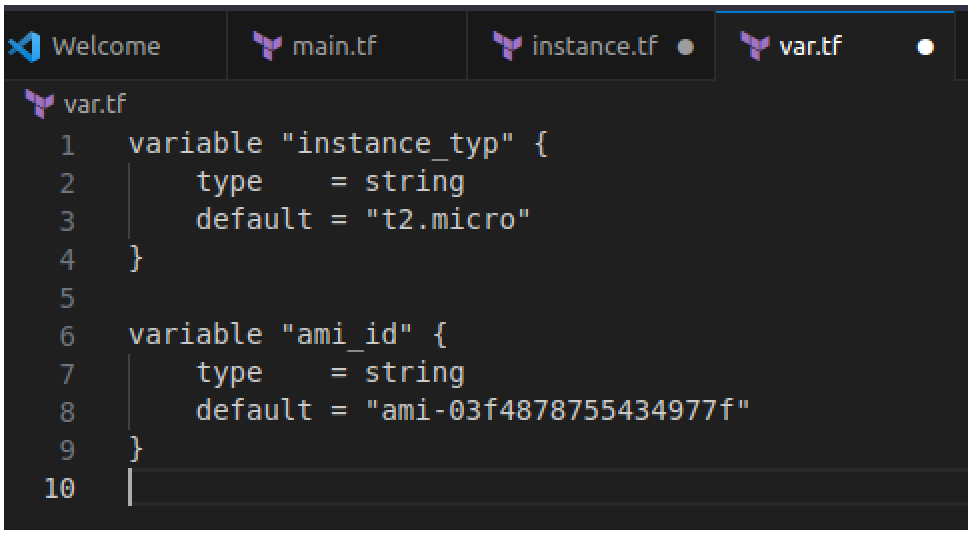
main.tf

```
terraform {
  required_providers {
    aws = {
      source = "hashicorp/aws"
      version = "5.35.0"
    }
  }
}

provider "aws" {
  region="ap-south-1"
  access_key="[redacted]
  secret_key="[redacted]"
}
```



instance.tf

```
resource "aws_instance" "lab4-1" {
    instance_type = var.instance_typ
    ami           = var.ami_id
    count         = 1
    tags = {
      Name = "lab4-b3-1"
    }
}
resource "aws_instance" "lab4-2" {
    instance_type = var.instance_typ
    ami           = var.ami_id
    count         = 1
    tags = {
      Name = "lab4-b3-2"
    }
}
resource "aws_instance" "lab4-3" {
    instance_type = var.instance_typ
    ami           = var.ami_id
    count         = 1
    tags = {
      Name = "lab4-b3-3"
    }
}
```



```
variable "instance_typ" {
    type    = string
    default = "t2.micro"
}

variable "ami_id" {
    type    = string
    default = "ami-03f4878755434977f"
}

```

Now by again running the terraform plan and terraform apply multiple instance will be created.