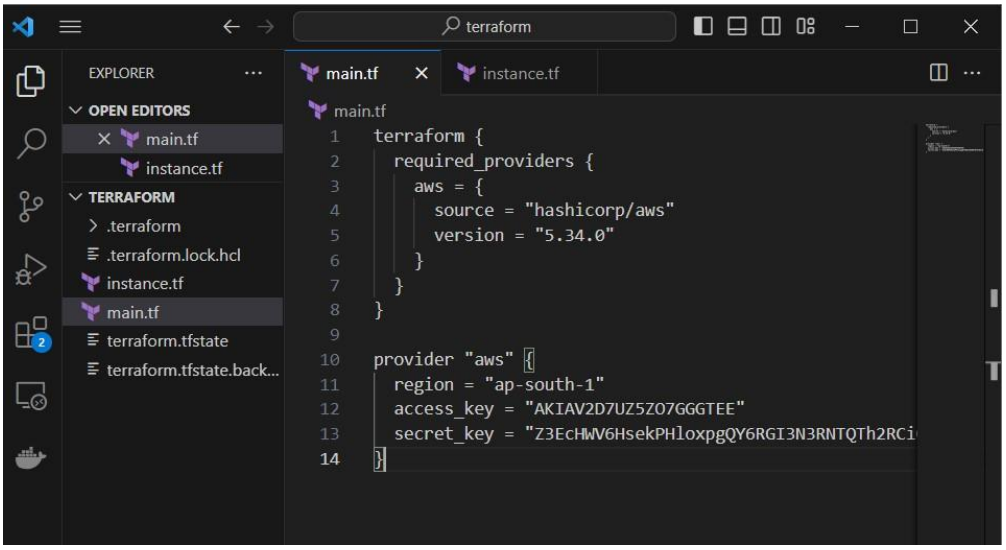
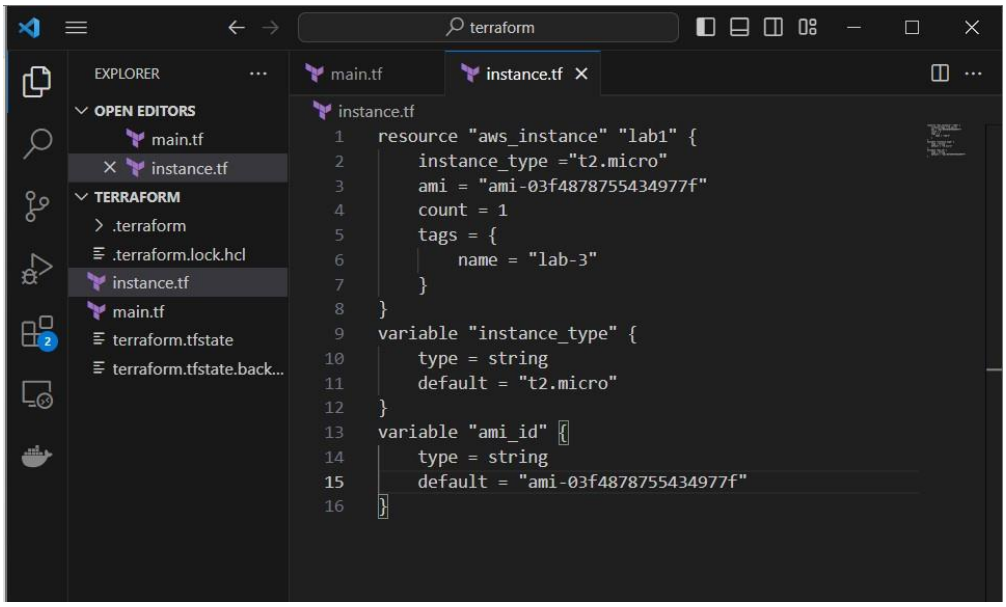
LAB-4

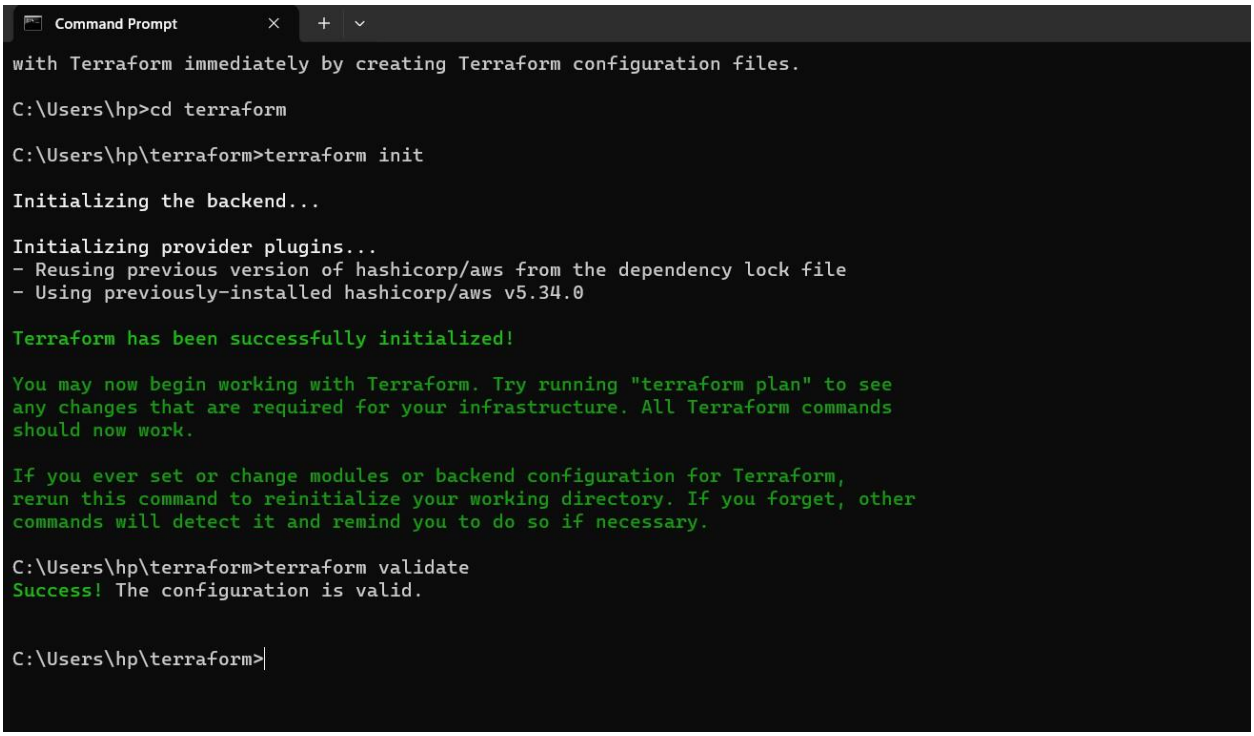
Terraform Variable

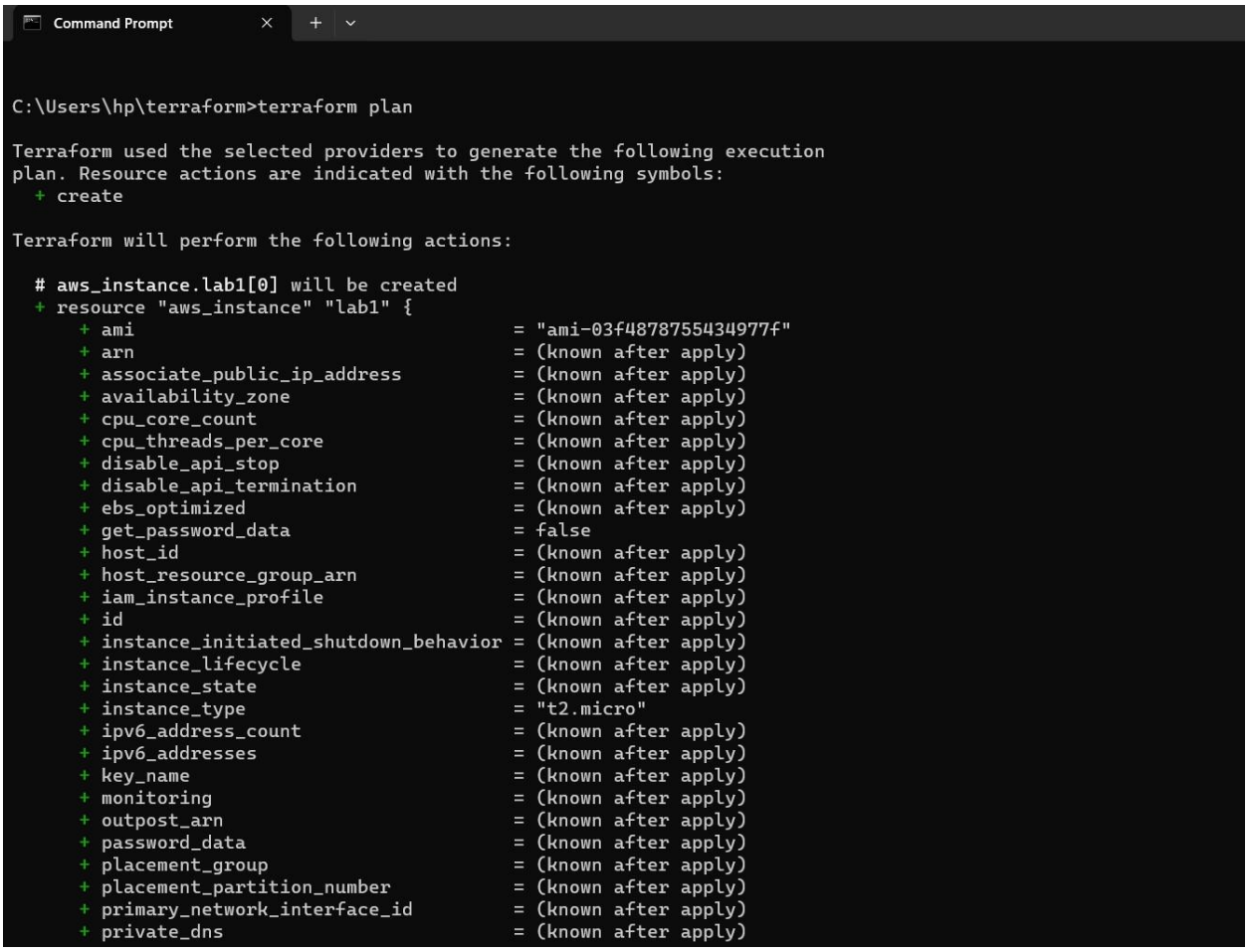
We will see different ways to declare variable in terraform

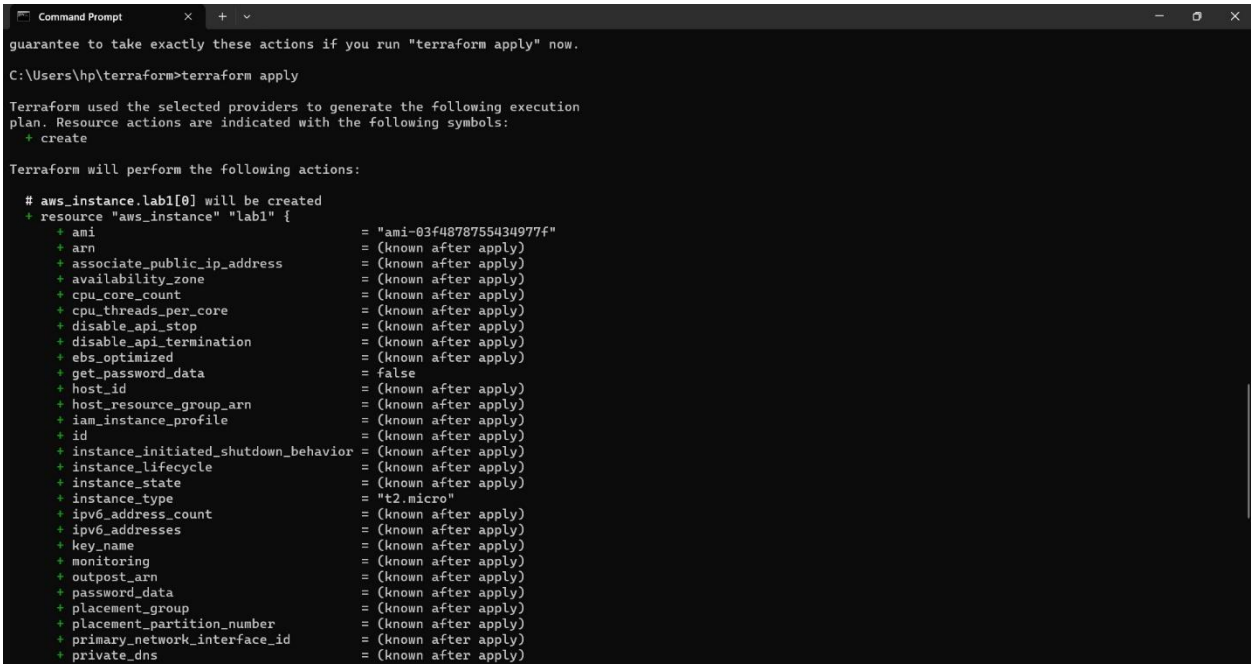
Step 1: First we will see declaring variable in instance.tf file

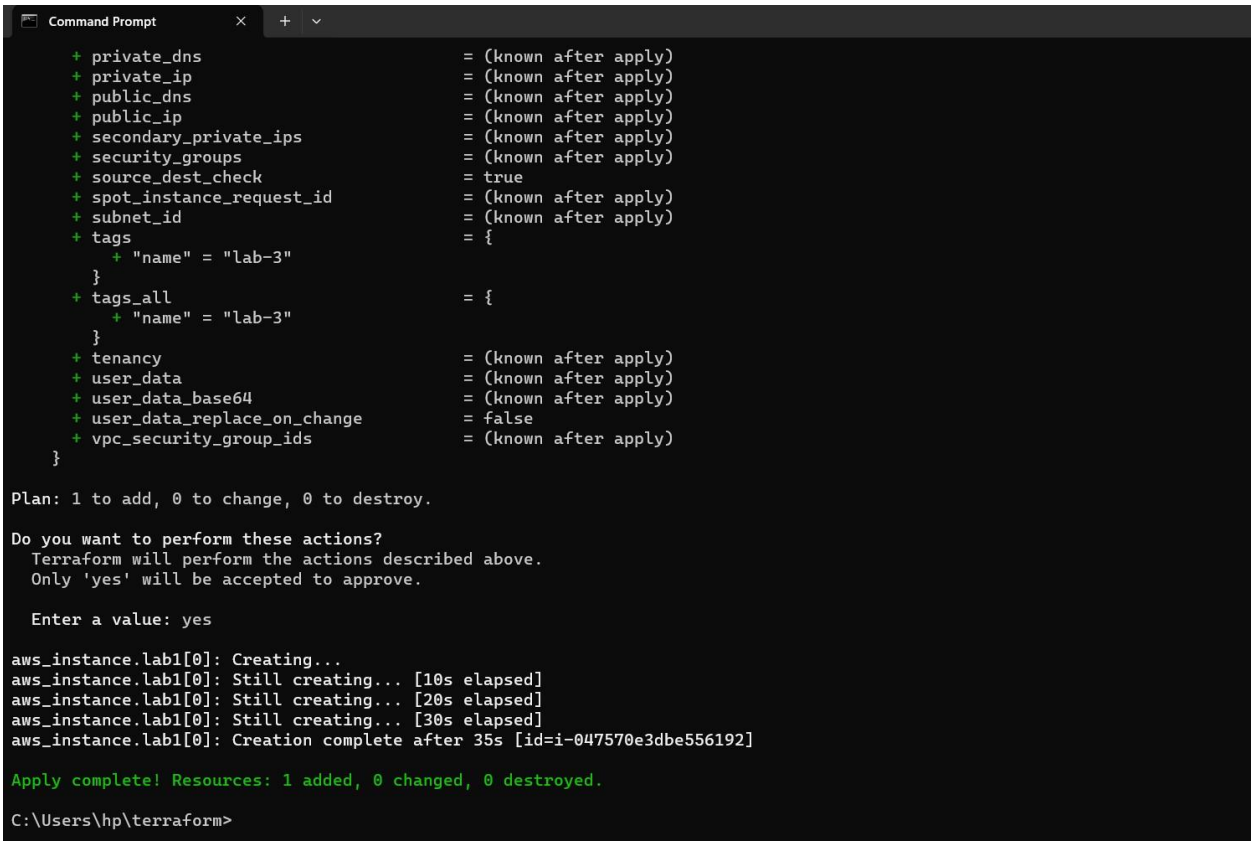


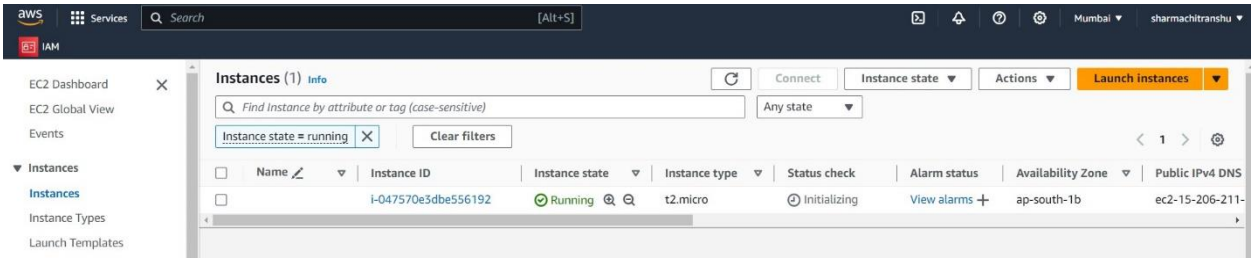


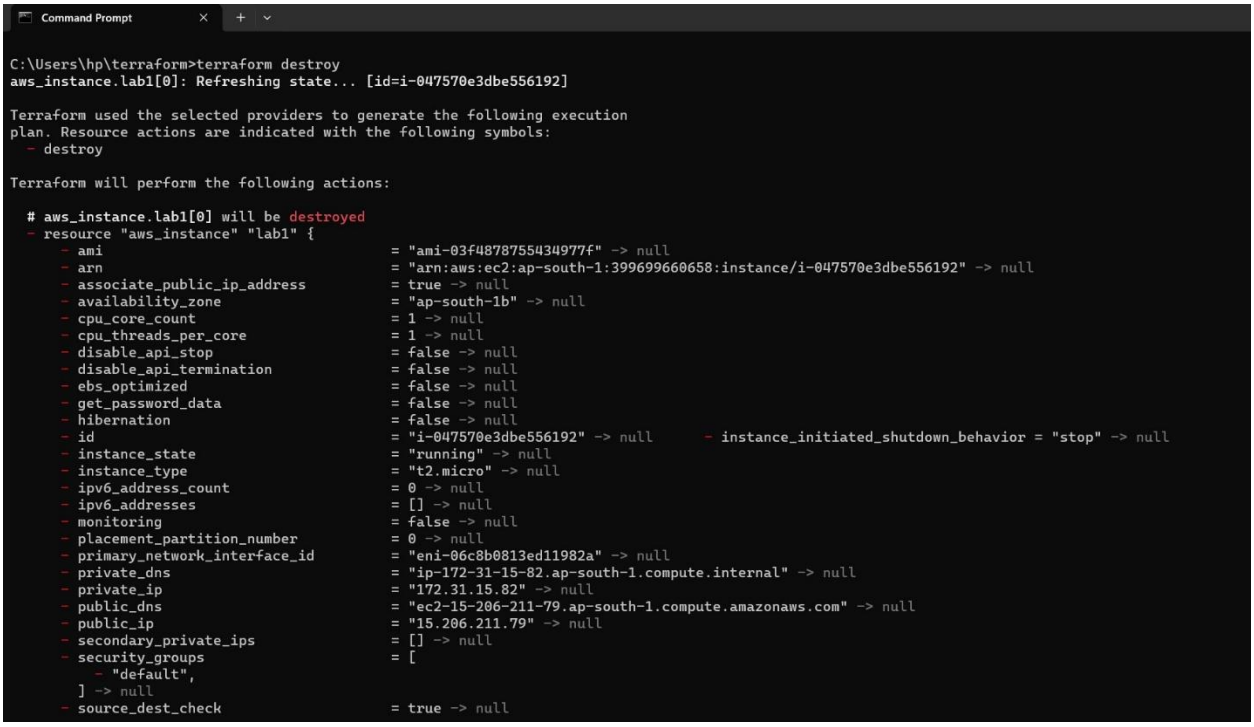

Command Prompt
with Terraform immediately by creating Terraform configuration files.
C:\Users\hp>cd terraform
C:\Users\hp\terraform>terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed 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.
C:\Users\hp\terraform>

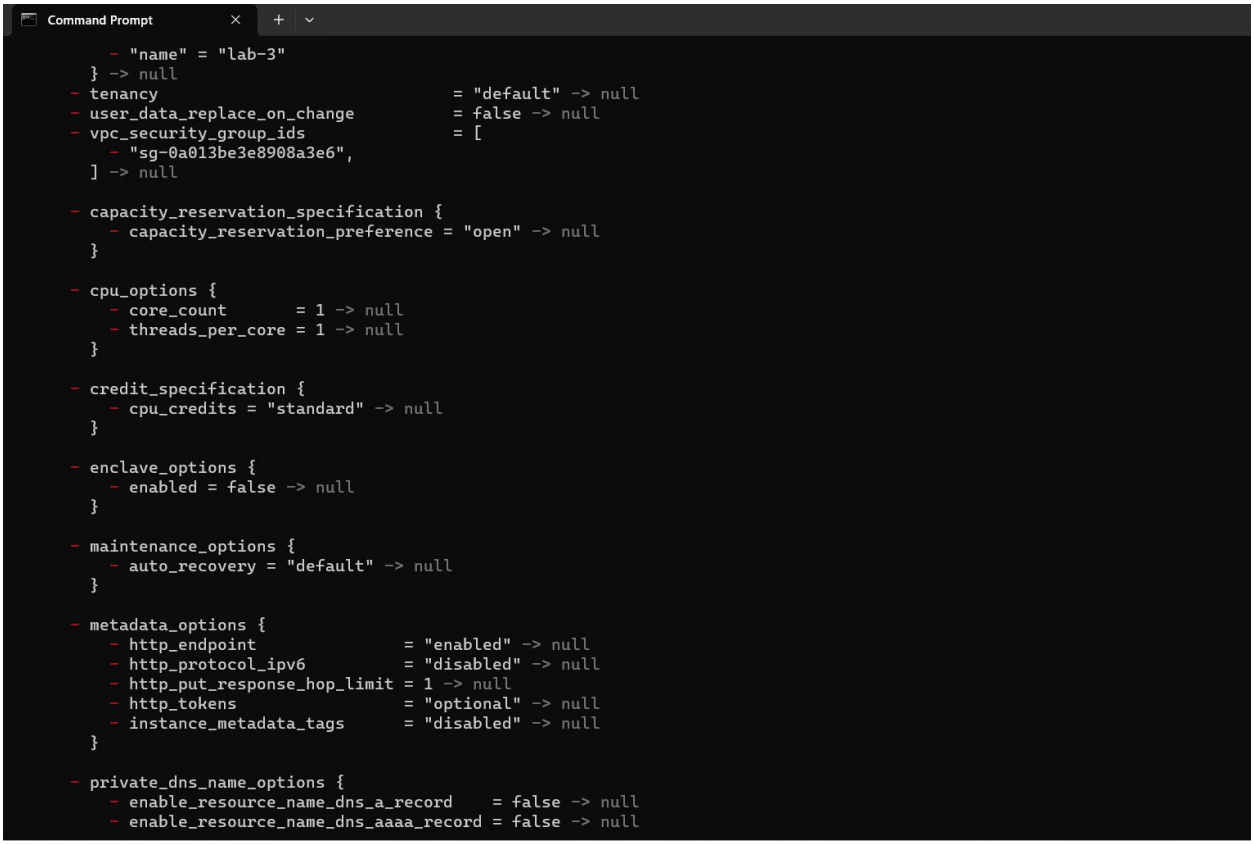

Command Prompt
C:\Users\hp\terraform>terraform plan
Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
+ create
Terraform will perform the following actions:
# aws_instance.lab1[0] will be created
+ resource "aws_instance" "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)


Command Prompt
guarantee to take exactly these actions if you run "terraform apply" now.
C:\Users\hp\terraform>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.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)

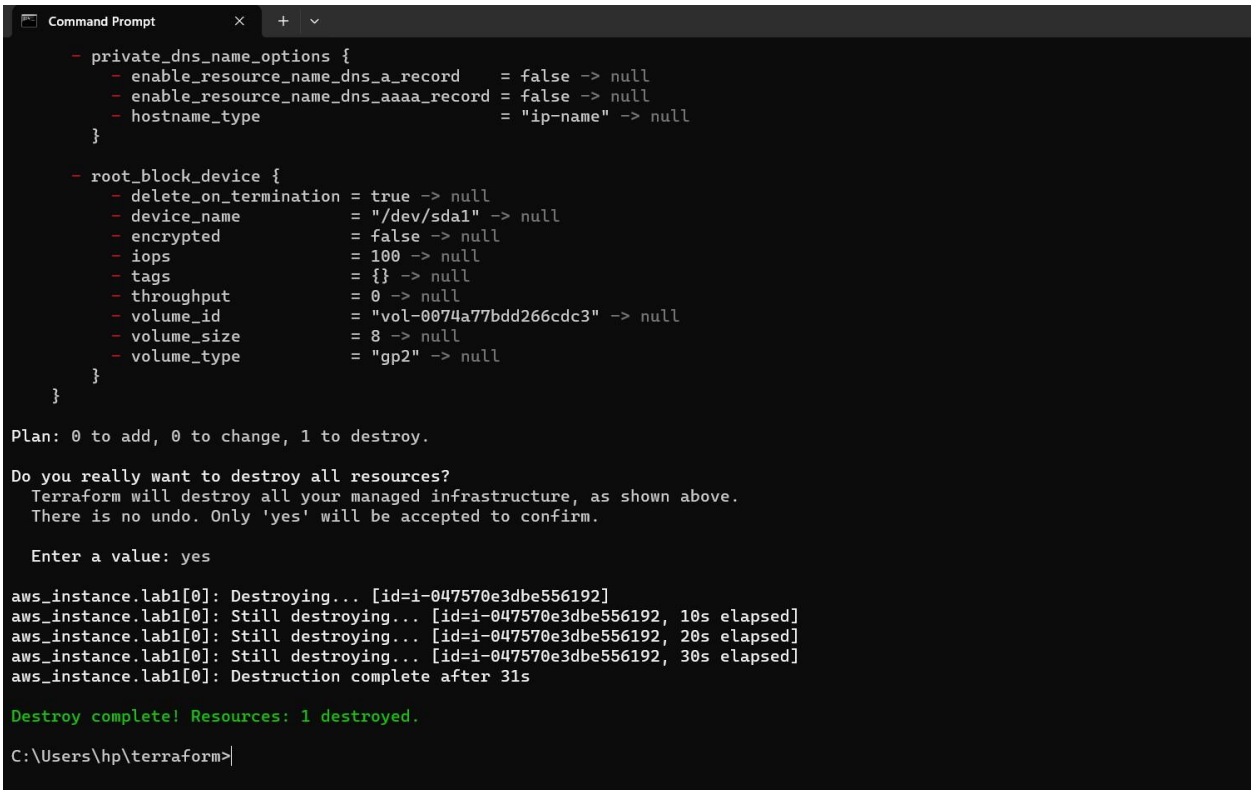

Command Prompt
+ private_dns = (known after apply)
+ private_ip = (known after apply)
+ public_dns = (known after apply)
+ public_ip = (known after apply)
+ secondary_private_ips = (known after apply)
+ security_groups = (known after apply)
+ source_dest_check = true
+ spot_instance_request_id = (known after apply)
+ subnet_id = (known after apply)
+ tags = {
+ "name" = "lab-3"
}
+ tags_all = {
+ "name" = "lab-3"
}
+ 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 35s [id=i-047570e3dbe556192]
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
C:\Users\hp\terraform>


aws
Services
Search
[Alt+S]
Mumbai
sharmachitranshu
IAM
EC2 Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Instances (1) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
Any state
Instance state = running
Clear filters
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
i-047570e3dbe556192
Running
t2.micro
Initializing
View alarms
ap-south-1b
ec2-15-206-211-

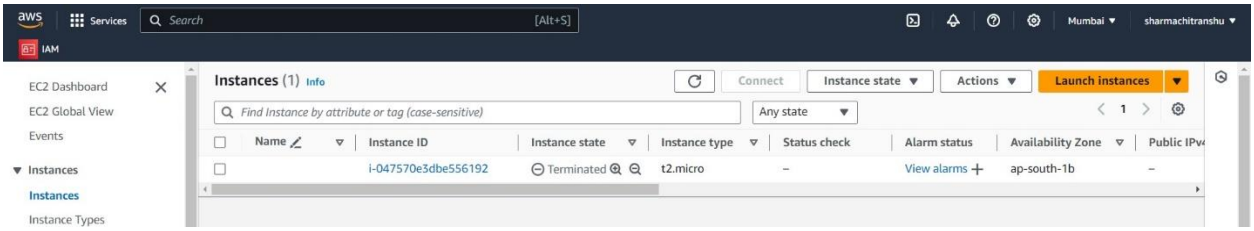

Command Prompt
C:\Users\hp\terraform>terraform destroy
aws_instance.lab1[0]: Refreshing state... [id=i-047570e3dbe556192]
Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
- destroy
Terraform will perform the following actions:
# aws_instance.lab1[0] will be destroyed
- resource "aws_instance" "lab1" {
- ami = "ami-03f48787554349777f" -> null
- arn = "arn:aws:ec2:ap-south-1:399699660658:instance/i-047570e3dbe556192" -> 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-047570e3dbe556192" -> 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-06c8b0813ed11982a" -> null
- private_dns = "ip-172-31-15-82.ap-south-1.compute.internal" -> null
- private_ip = "172.31.15.82" -> null
- public_dns = "ec2-15-206-211-79.ap-south-1.compute.amazonaws.com" -> null
- public_ip = "15.206.211.79" -> null
- secondary_private_ips = [] -> null
- security_groups = [
- "default",
] -> null
- source_dest_check = true -> null



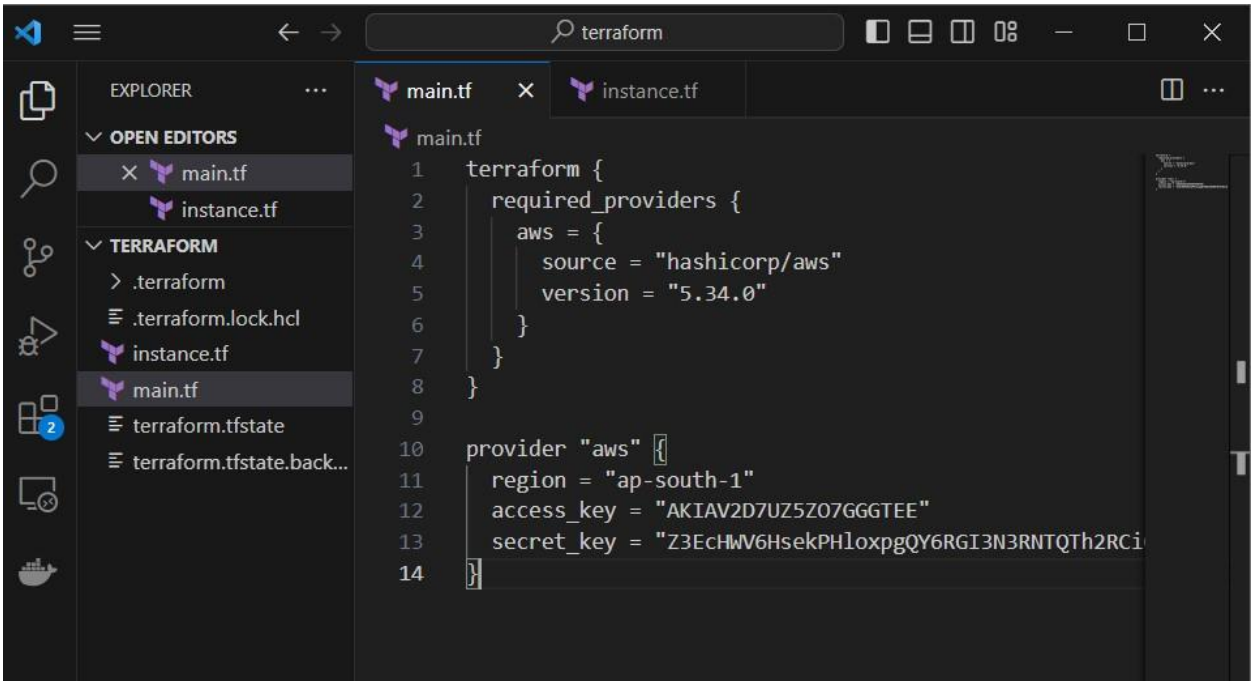
```
Command Prompt
        - "name" = "lab-3"
      } -> null
    - tenancy                              = "default" -> null
    - user_data_replace_on_change          = false -> null
    - vpc_security_group_ids               = [
        - "sg-0a013be3e8908a3e6",
      ] -> null

    - capacity_reservation_specification {
        - capacity_reservation_preference = "open" -> null
      }

    - cpu_options {
        - core_count       = 1 -> null
        - threads_per_core = 1 -> null
      }

    - credit_specification {
        - cpu_credits = "standard" -> null
      }

    - enclave_options {
        - enabled = false -> null
      }

    - maintenance_options {
        - auto_recovery = "default" -> null
      }

    - metadata_options {
        - http_endpoint               = "enabled" -> null
        - http_protocol_ipv6          = "disabled" -> null
        - http_put_response_hop_limit = 1 -> null
        - http_tokens                 = "optional" -> null
        - 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
```

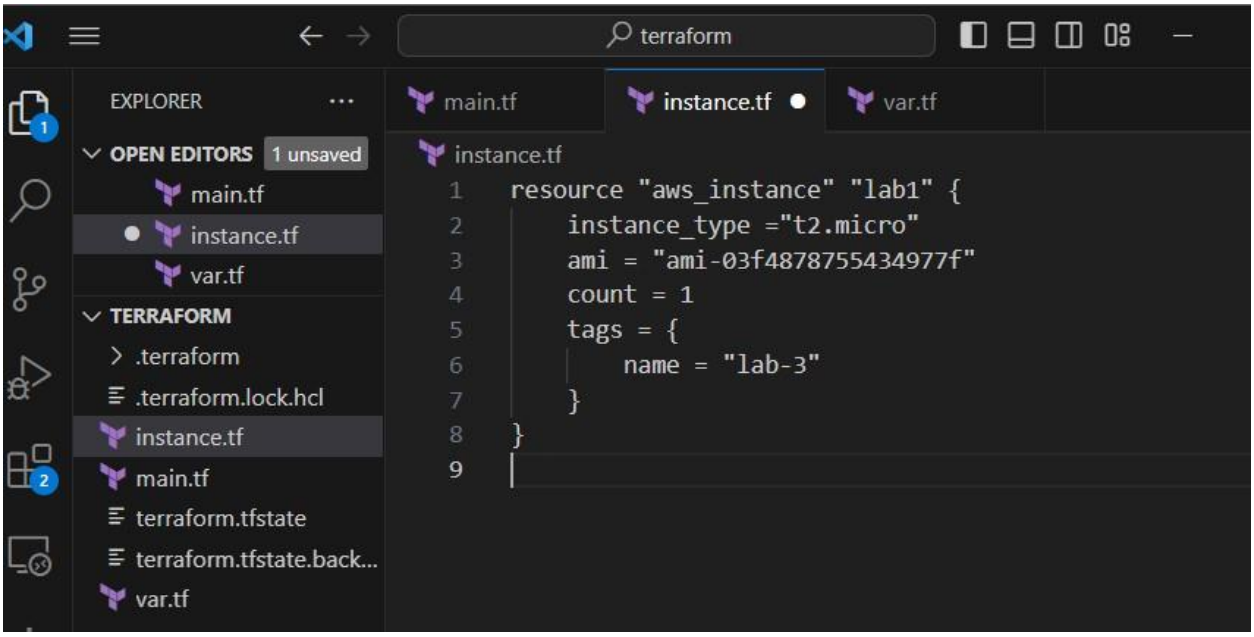


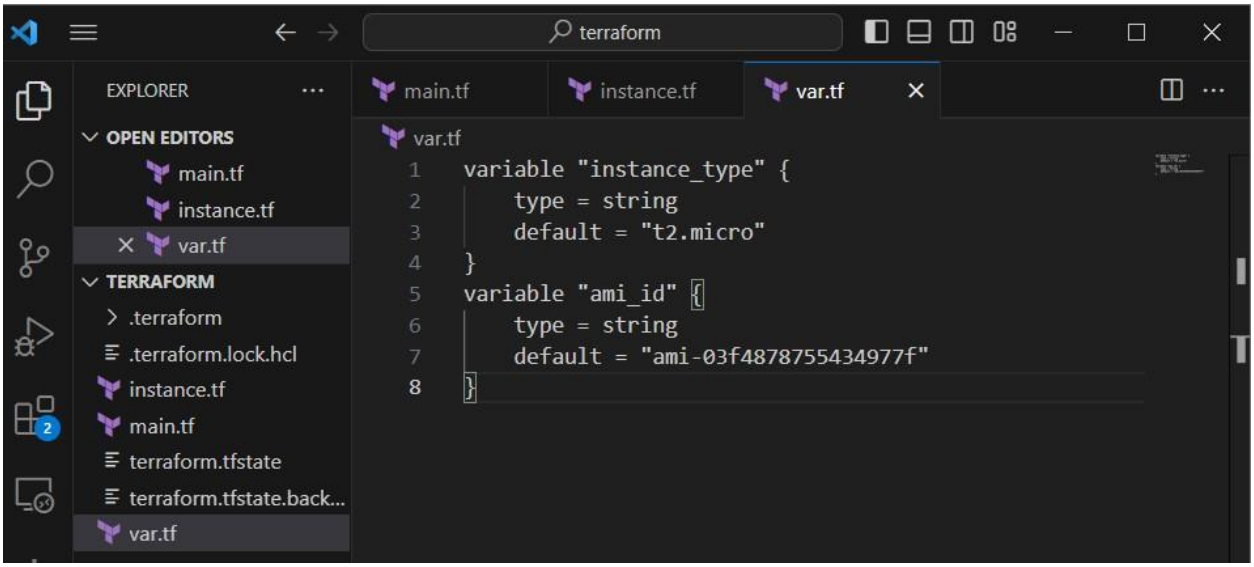
```
Command Prompt
    - private_dns_name_options {
        - enable_resource_name_dns_a_record    = false -> null
        - enable_resource_name_dns_aaaa_record = false -> null
        - hostname_type                        = "ip-name" -> null
      }

    - 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-0074a77bdd266cdc3" -> 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-047570e3dbe556192]
aws_instance.lab1[0]: Still destroying... [id=i-047570e3dbe556192, 10s elapsed]
aws_instance.lab1[0]: Still destroying... [id=i-047570e3dbe556192, 20s elapsed]
aws_instance.lab1[0]: Still destroying... [id=i-047570e3dbe556192, 30s elapsed]
aws_instance.lab1[0]: Destruction complete after 31s

Destroy complete! Resources: 1 destroyed.

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



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

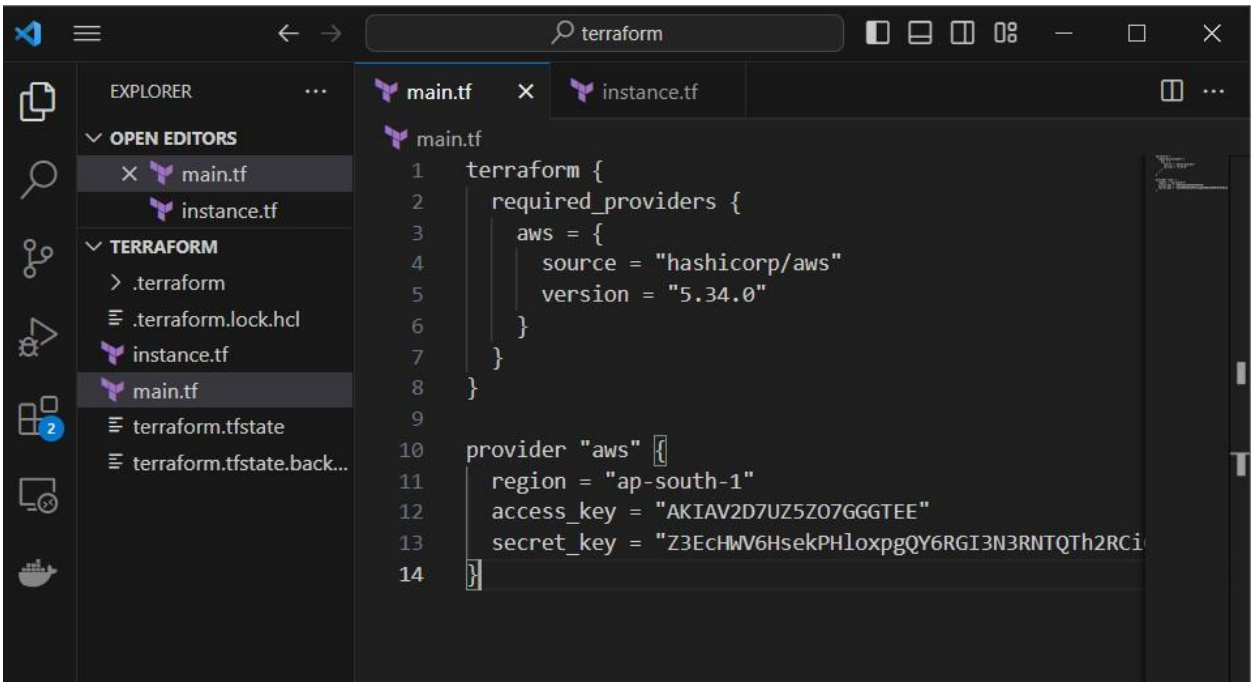


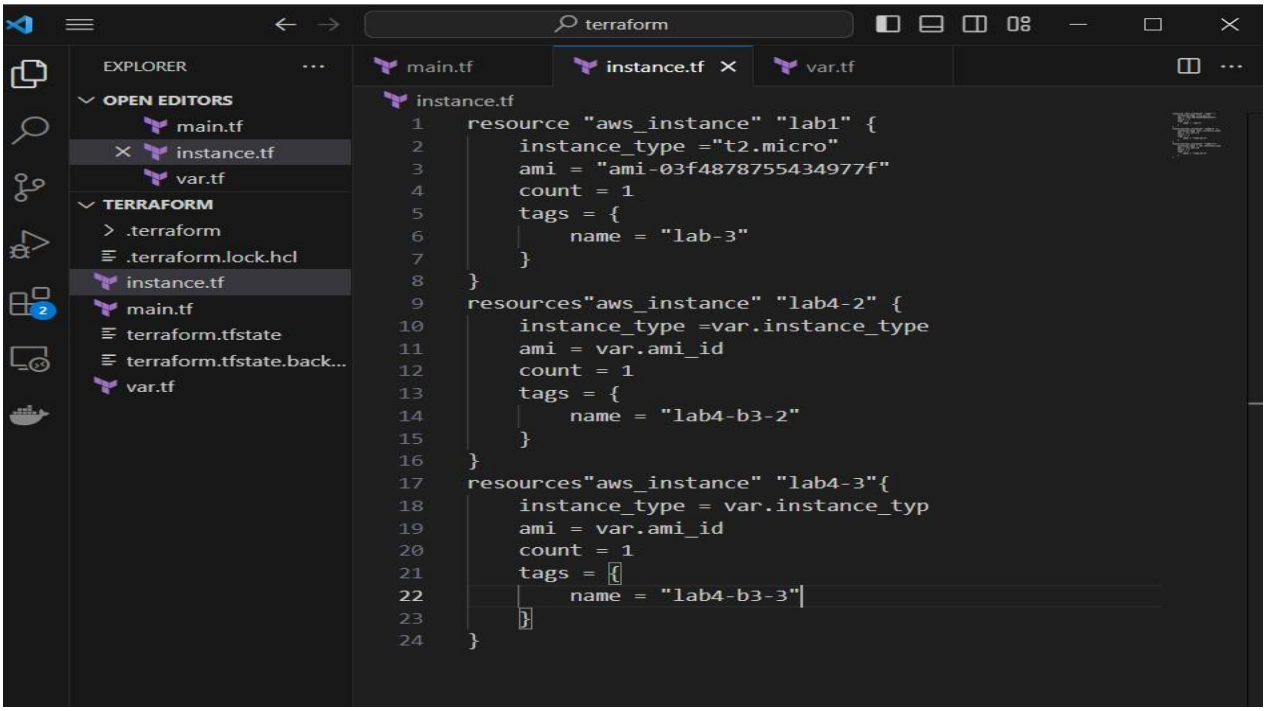


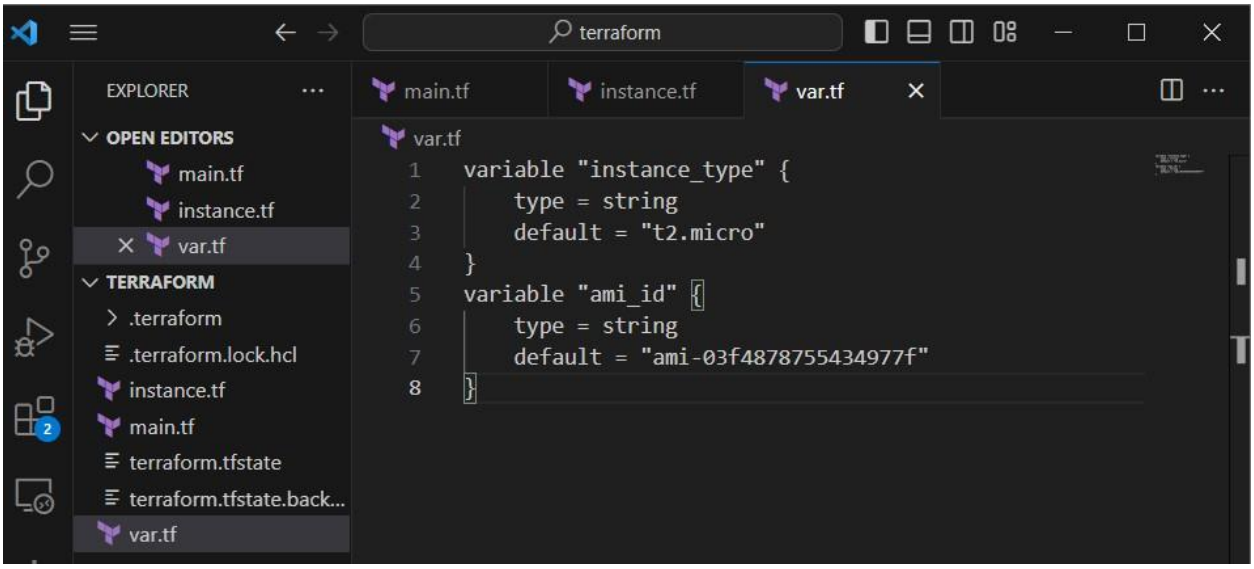


Now by again running the terraform plan and terraform apply instance will be created.

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







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