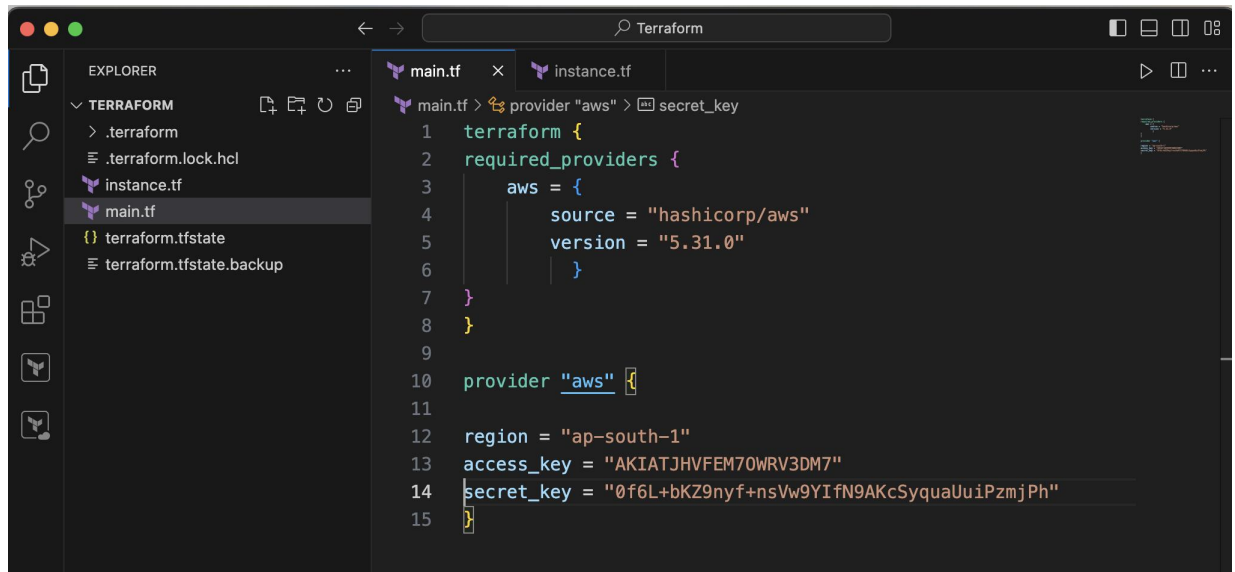
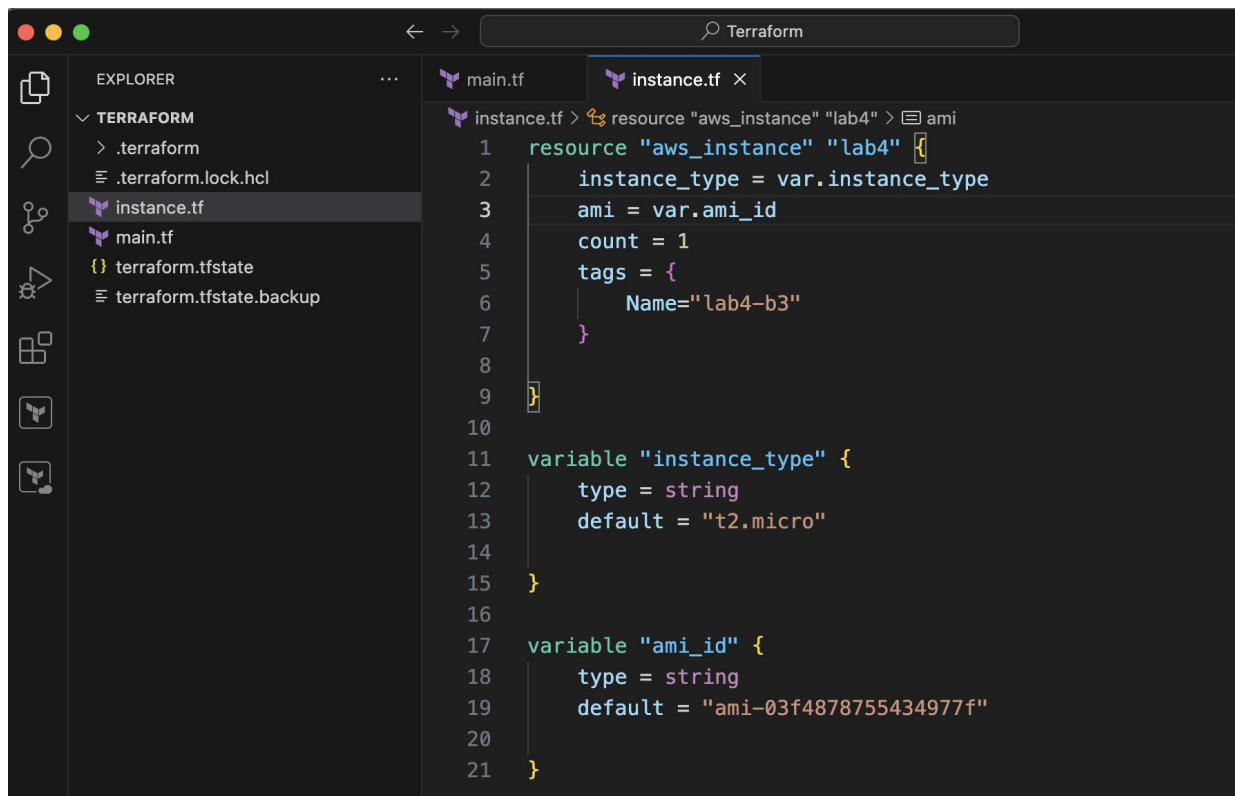


## LAB-4 Terraform Variable

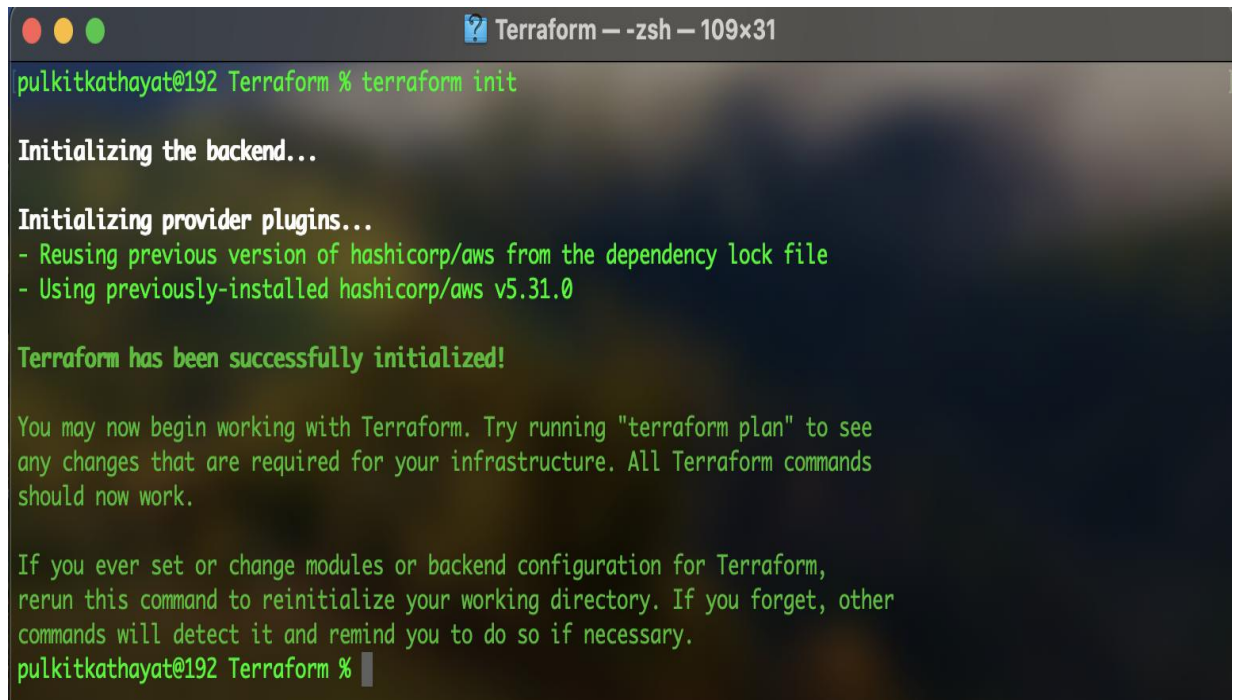
We will see different ways to declare variable in terraform **Step 1:** First we will see declaring variable in instance.tf file



```
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 provider "aws" {}
11
12 region = "ap-south-1"
13 access_key = "AKIATJHVFEM70WRV3DM7"
14 secret_key = "0f6L+bKZ9nyf+nsVw9YIfN9AKcSyquaUuiPzmjPh"
15 }
```



```
1 resource "aws_instance" "lab4" {
2   instance_type = var.instance_type
3   ami = var.ami_id
4   count = 1
5   tags = {
6     Name = "lab4-b3"
7   }
8 }
9
10
11 variable "instance_type" {
12   type = string
13   default = "t2.micro"
14 }
15
16
17 variable "ami_id" {
18   type = string
19   default = "ami-03f4878755434977f"
20 }
21 }
```

A terminal window titled "Terraform — -zsh — 109x31" with standard macOS window controls (red, yellow, green buttons). The terminal shows the execution of the "terraform init" command. The output includes status messages for backend and provider initialization, a list of actions taken (reusing hashicorp/aws from a lock file and using v5.31.0), a success confirmation, and instructions on how to use Terraform and when to reinitialize. The prompt "pulkitkathayat@192 Terraform %" is visible at the bottom with a cursor.

```
pulkitkathayat@192 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 v5.31.0

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.
pulkitkathayat@192 Terraform %
```

```
pulkitkathayat@192 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.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_group             = (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.

---

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

 Terraform — -zsh — 1

```
pulkitkathayat@192 Terraform % terraform validate
```

Success! The configuration is valid.

```
pulkitkathayat@192 Terraform %
```

Terraform will perform the following actions:

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

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

The screenshot shows the AWS Management Console's EC2 Instances page. At the top, there are navigation tabs: "Instances (1)", "Instance Types", "Launch Templates", "Spot Requests", and "Savings Plans". Below the tabs, there's a search bar labeled "Find Instance by attribute or tag (case-sensitive)". To the right of the search bar are filters for "Any state" and pagination controls showing "1" instance. A toolbar contains buttons for "Connect", "Instance state", "Actions", and a prominent orange "Launch instances" button. Below this is a table listing the instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	lab4-b3	i-04e1e93a57dd54b1e	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>	ap-south-1b	ec2-

On the left side of the console, there's a sidebar menu with options: "EC2 Dashboard", "EC2 Global View", "Events", "Instances" (selected), "Instance Types", "Launch Templates", "Spot Requests", and "Savings Plans".

```
pulkitkathayat@192 Terraform % terraform destroy
```

```
aws_instance.lab4[0]: Refreshing state... [id=i-04e1e93a57dd54b1e]
```

```
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-1:225999921982:instance/i-04e1e93a57dd54b1e" -> 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-04e1e93a57dd54b1e" -> 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-060f78518903bb7ce" -> null  
  - private_dns                       = "ip-172-31-0-152.ap-south-1.compute.internal" -> null  
  - private_ip                        = "172.31.0.152" -> null  
  - public_dns                        = "ec2-3-110-196-236.ap-south-1.compute.amazonaws.com" -> null  
  - public_ip                         = "3.110.196.236" -> null  
  - secondary_private_ips              = [] -> null  
  - security_groups                   = [  
    - "default",  
  ] -> null  
  - source_dest_check                 = true -> null  
  - subnet_id                         = "subnet-0af728688777a3754" -> null  
  - tags                              = {  
    - "Name" = "lab4-b3"  
  } -> null  
  - tags_all                          = {  
    - "Name" = "lab4-b3"  
  } -> null  
  - tenancy                           = "default" -> null  
  - user_data_replace_on_change       = false -> null  
  - vpc_security_group_ids            = [  
    - "sg-0bf6ac3abde81c033",  
  ] -> 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  
  }
```

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

- root_block_device {
  - delete_on_termination = true -> null
  - device_name = "/dev/sda1" -> null
  - encrypted = false -> null
  - iops = 100 -> null
  - kms_key_id = "" -> null
  - throughput = 0 -> null
  - volume_id = "vol-44fa2a0bf3b82c3a4" -> null
  - volume_size = 8 -> null
  - volume_type = "gp2" -> null
}
}

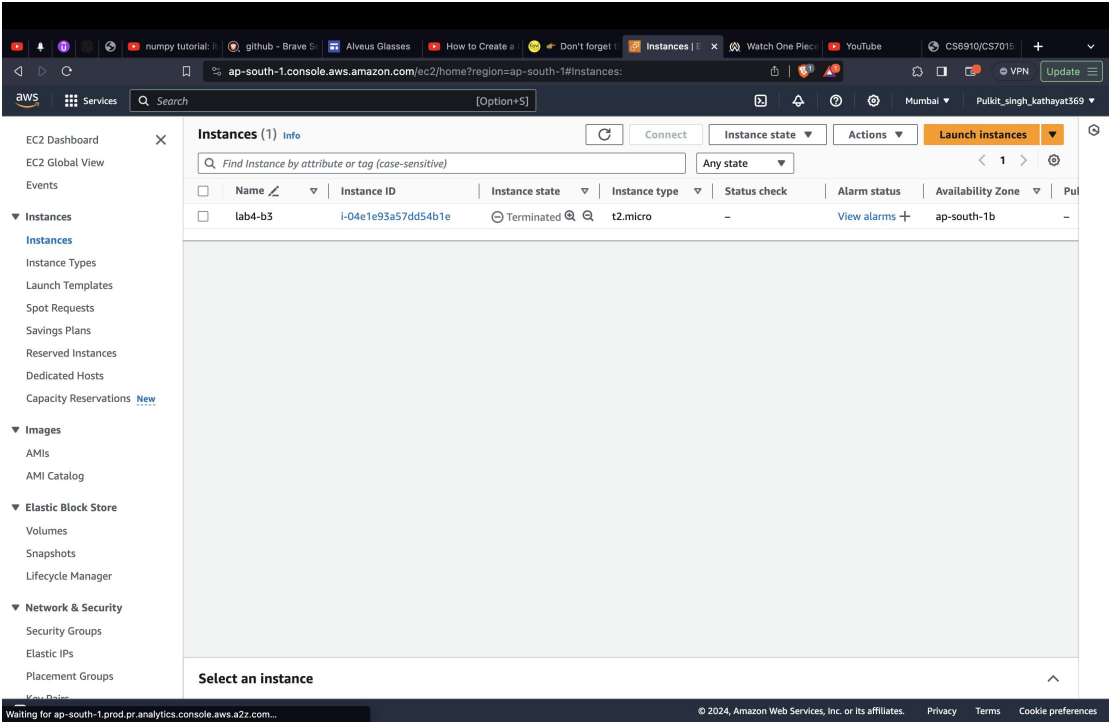
Plan: 0 to add, 0 to change, 1 to destroy.

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.

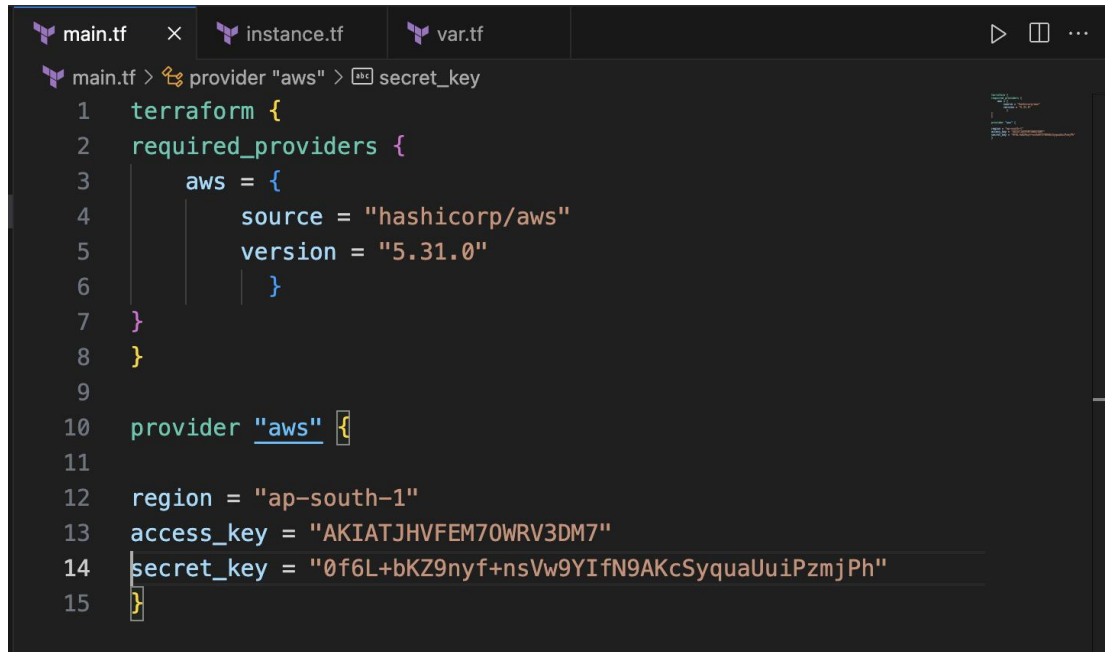
Enter a value: yes

aws_instance.lab4[0]: Destroying... [id=i-04e1e93a57dd54b1e]
aws_instance.lab4[0]: Still destroying... [id=i-04e1e93a57dd54b1e, 10s elapsed]
aws_instance.lab4[0]: Still destroying... [id=i-04e1e93a57dd54b1e, 20s elapsed]
aws_instance.lab4[0]: Still destroying... [id=i-04e1e93a57dd54b1e, 30s elapsed]
aws_instance.lab4[0]: Destruction complete after 30s

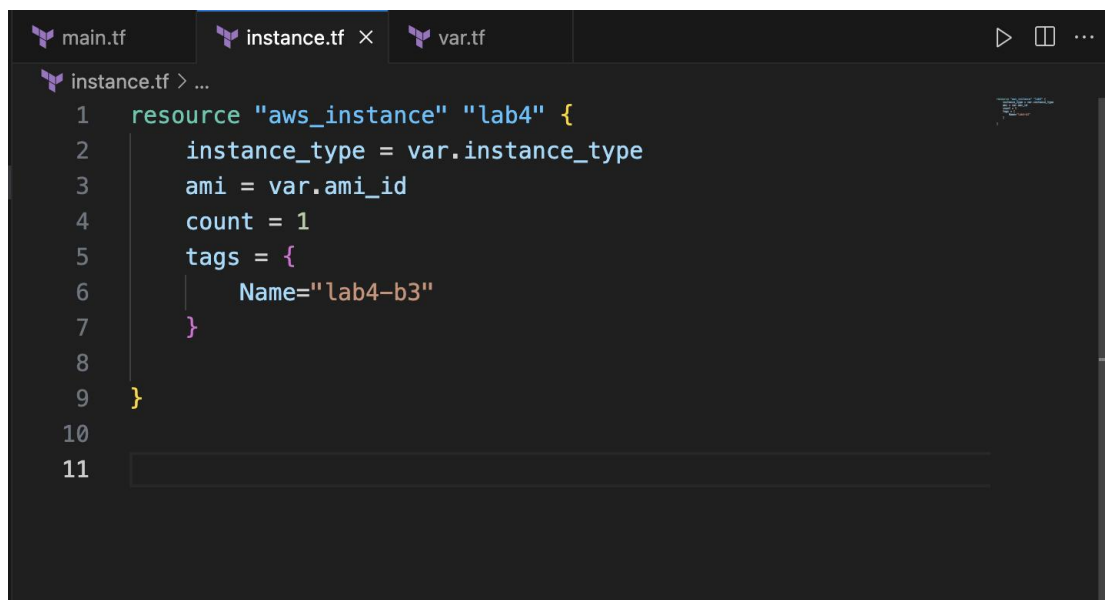
Destroy complete! Resources: 1 destroyed.
pulkittkathayat@192 Terraform %
```



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

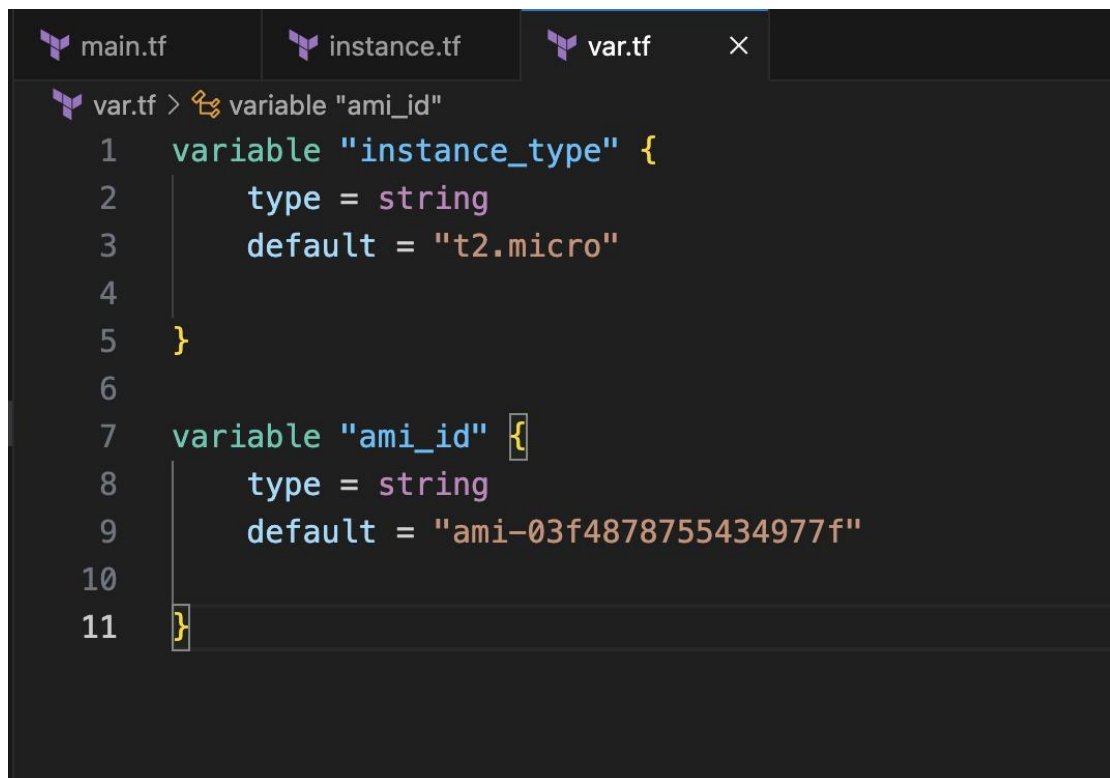


```
main.tf × instance.tf var.tf
main.tf > provider "aws" > secret_key
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 provider "aws" {}
11
12 region = "ap-south-1"
13 access_key = "AKIATJHVFEM70WRV3DM7"
14 secret_key = "0f6L+bKZ9nyf+nsVw9YIfN9AKcSyquaUuiPzmjPh"
15 }
```



```
main.tf instance.tf × var.tf
instance.tf > ...
1 resource "aws_instance" "lab4" {
2   instance_type = var.instance_type
3   ami = var.ami_id
4   count = 1
5   tags = {
6     Name = "lab4-b3"
7   }
8 }
9
10
11
```



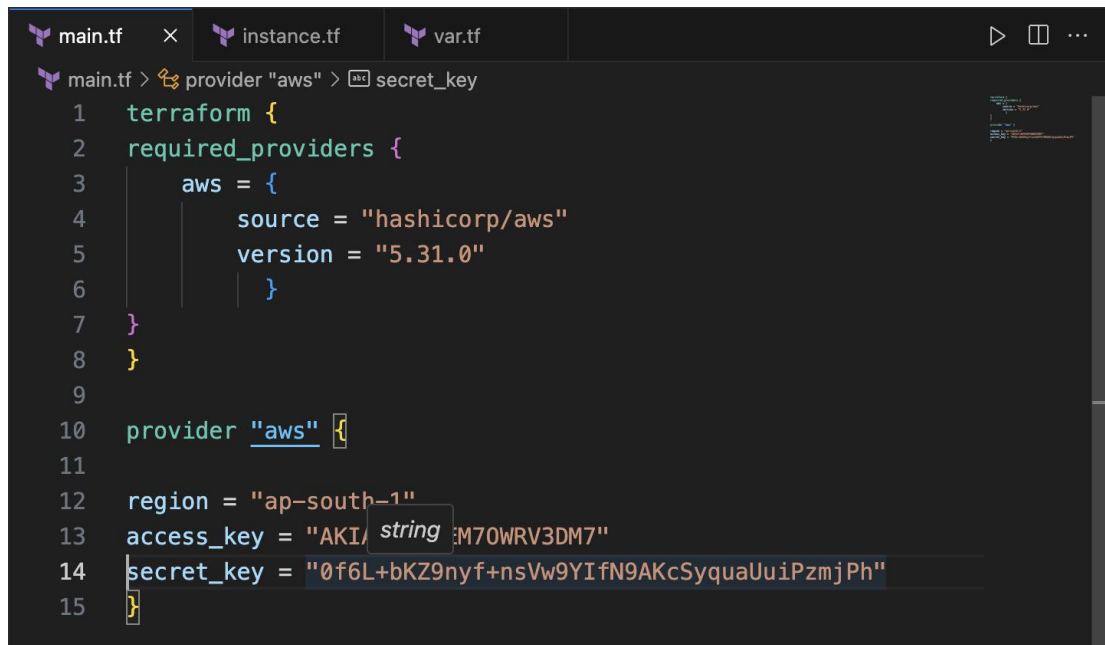


```
main.tf instance.tf var.tf ×
var.tf > variable "ami_id"
1  variable "instance_type" {
2      type = string
3      default = "t2.micro"
4
5  }
6
7  variable "ami_id" {
8      type = string
9      default = "ami-03f4878755434977f"
10
11 }
```

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



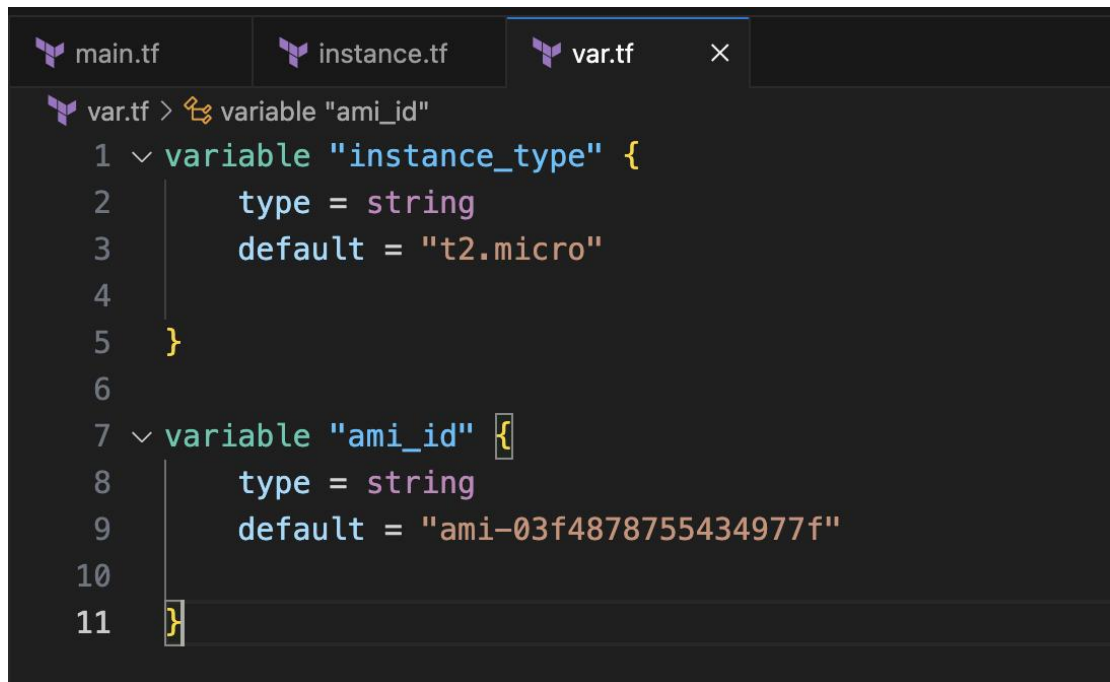
**Step 3:** To create multiple instances by changing instance.tf file



The screenshot shows a code editor with three tabs: main.tf, instance.tf, and var.tf. The main.tf tab is active, displaying Terraform configuration code. The code defines the AWS provider within a terraform block, sets the required\_providers block, and configures the provider block with specific credentials and region. The secret\_key value is highlighted with a blue selection bar.

```
main.tf > provider "aws" > secret_key
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.31.0"
6     }
7   }
8 }
9
10 provider "aws" {
11
12   region = "ap-south-1"
13   access_key = "AKIA...string M70WRV3DM7"
14   secret_key = "0f6L+bKZ9nyf+nsVw9YIfN9AKcSyquaUuiPzmjPh"
15 }
```

```
main.tf instance.tf x var.tf
instance.tf > resource "aws_instance" "lab4-3"
1 resource "aws_instance" "lab4-1" {
2     instance_type = var.instance_type
3     ami = var.ami_id
4     count = 1
documents/Terraform/main.tf {
6     Name="lab4-b3-1"
7 }
8
9 }
10
11 resource "aws_instance" "lab4-2" {
12     instance_type = var.instance_type
13     ami = var.ami_id
14     count = 1
15     tags = {
16         Name="lab4-b3-2"
17     }
18 }
19
20
21 resource "aws_instance" "lab4-3" {
22     instance_type = var.instance_type
23     ami = var.ami_id
24     count = 1
25     tags = {
26         Name="lab4-b3-3"
27     }
28 }
29
30
31
```



```
main.tf instance.tf var.tf ×
var.tf > variable "ami_id"
1  variable "instance_type" {
2      type = string
3      default = "t2.micro"
4
5  }
6
7  variable "ami_id" {
8      type = string
9      default = "ami-03f4878755434977f"
10
11 }
```

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