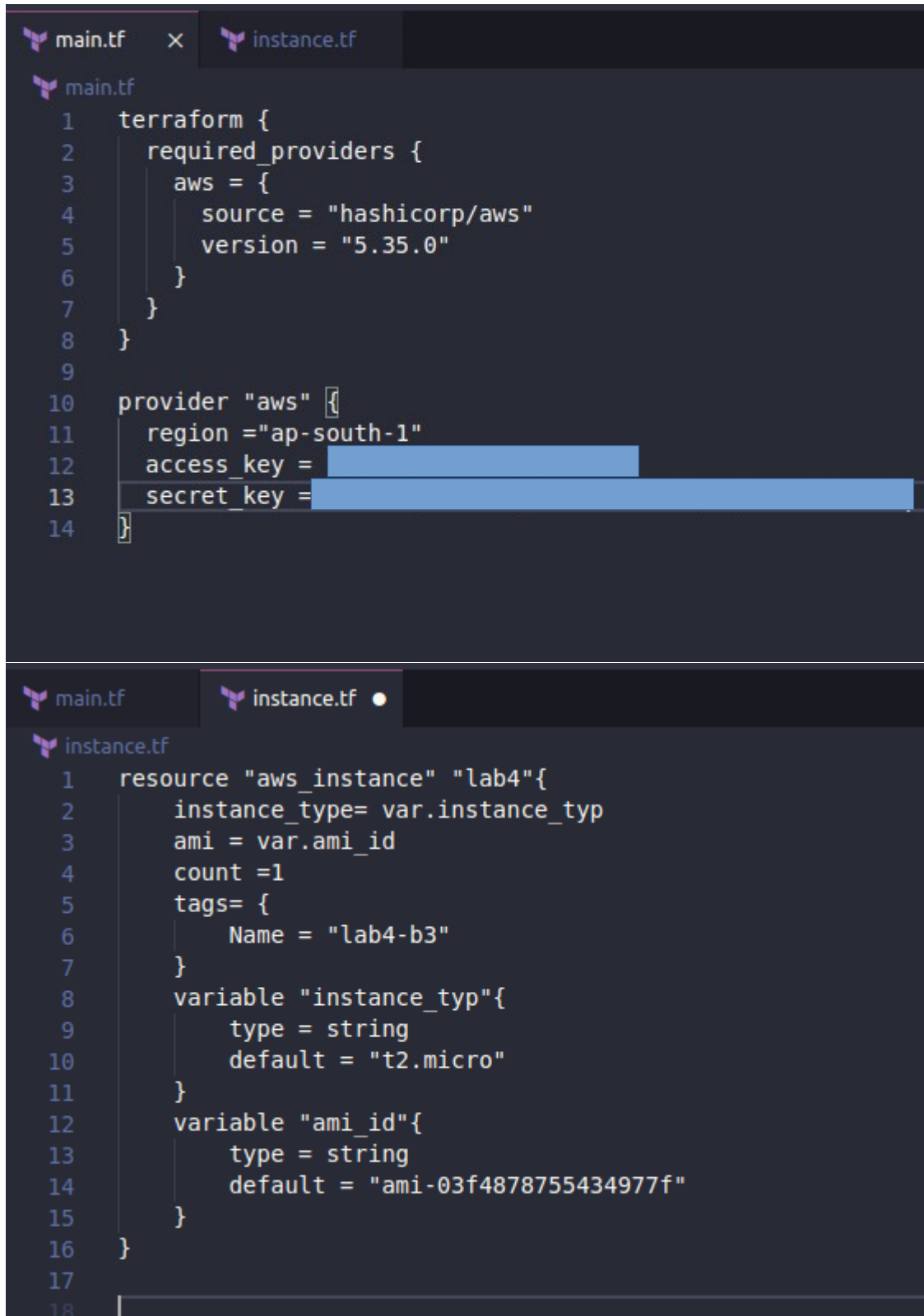


LAB-4

Terraform Variable

We will see different ways to declare variable in terraform

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



```
main.tf
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.35.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "ap-south-1"
12   access_key = 
13   secret_key = 
14 }

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   variable "instance_type" {
9     type = string
10    default = "t2.micro"
11  }
12  variable "ami_id" {
13    type = string
14    default = "ami-03f4878755434977f"
15  }
16 }
17
18
```

```
~/Documents/SPCM/Terraform v1.7.1default 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.

```
~/Documents/SPCM/Terraform v1.7.1default as  took 5s  
→ terraform validate
```

Success! The configuration is valid.

```
~/Documents/SPCM/Terraform v1.7.1default as  took 4s  
→ 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)
  + 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" = "lab1-b3"
  }
```

```
~/Documents/SPCM/Terraform v1.7.1default 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.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_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" = "lab1"  
  }  
}
```

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:

Instances (2) info										
Find Instance by attribute or tag (case-sensitive)										
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	lab1-b3	i-07737e000667daa44	Terminated	t2.micro	-	View alarms +	ap-south-1a	-	-	-
<input type="checkbox"/>	lab1-b3	i-043471c5a6f1f79e5	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-13-233-230-243.ap...	13.233.230.243	-

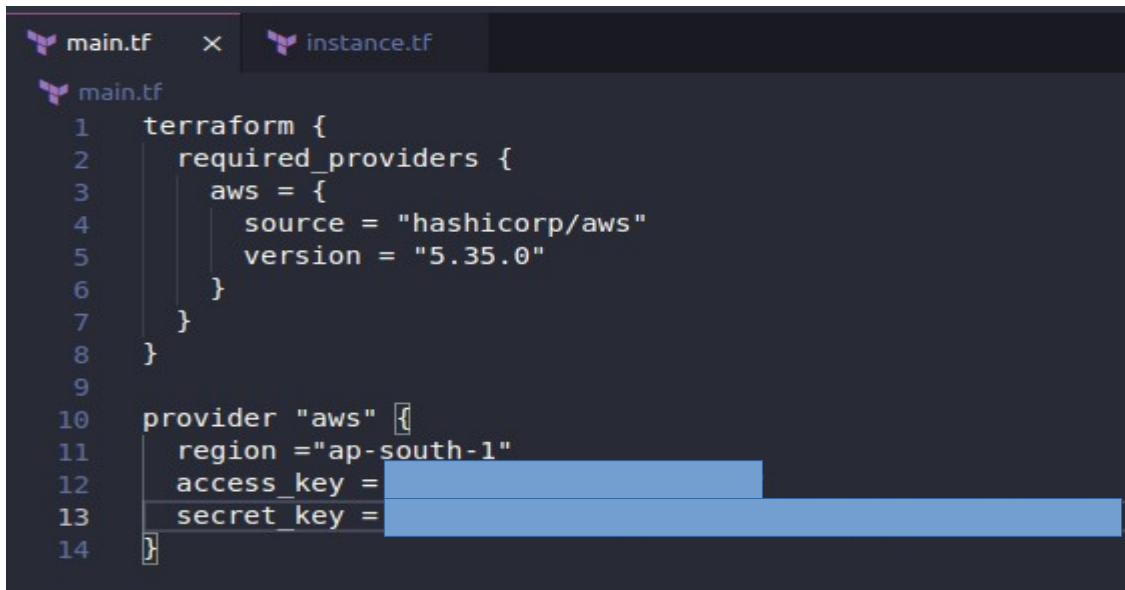
```
# aws_instance.lab1[0] will be destroyed
- resource "aws_instance" "lab1" {
  - ami
  - arn
  - associate_public_ip_address
  - availability_zone
  - cpu_core_count
  - cpu_threads_per_core
  - disable_api_stop
  - disable_api_termination
  - ebs_optimized
  - get_password_data
  - hibernation
  - id
  - instance_initiated_shutdown_behavior
  - instance_state
  - instance_type
  - ipv6_address_count
  - ipv6_addresses
  - monitoring
  - placement_partition_number
  - primary_network_interface_id
  - private_dns
  - private_ip
  - public_dns
  - public_ip
  - secondary_private_ips
  - security_groups
    - "default",
  ] -> null
  - source_dest_check
  - subnet_id
  - tags
    - "Name" = "lab1-b3"
  ] -> null
  - tags_all
    - "Name" = "lab1-b3"
  }

= "ami-03f48785543977f" -> null
= "arn:aws:ec2:ap-south-1:698194348131:instance/i-043471c5a6f1f79e3" -> null
= true -> null
= "ap-south-1a" -> null
= 1 -> null
= 1 -> null
= false -> null
= false -> null
= false -> null
= false -> null
= false -> null
= "i-043471c5a6f1f79e3" -> null
= "stop" -> null
= "running" -> null
= "t2.micro" -> null
= 0 -> null
= [] -> null
= false -> null
= 0 -> null
= "eni-0d6bdf6dc42345c16" -> null
= "ip-172-31-44-208.ap-south-1.compute.internal" -> null
= "172.31.44.208" -> null
= "ec2-13-233-230-243.ap-south-1.compute.amazonaws.com" -> null
= "13.233.230.243" -> null
= [] -> null
= [
  - "default",
] -> null
= true -> null
= "subnet-0d78b64e981bd0f9d" -> null
= {
  - "Name" = "lab1-b3"
} -> null
= {
  - "Name" = "lab1-b3"
}
```

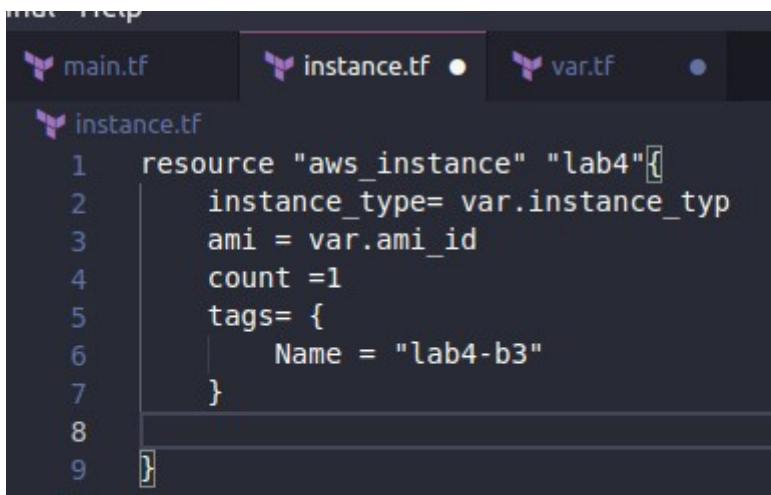
The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, IAM role information, a search bar, and various utility icons. Below this, the left-hand navigation pane lists options like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, and Launch Templates. The main area displays the 'Instances' page, which includes a filter bar at the top with a search input and a state dropdown set to 'Any state'. A table below lists the instances:

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	lab1-b3	i-07737e000667daa44	Terminated	t2.micro	-	View alarms +	ap-south-1a	-	-	-
<input type="checkbox"/>	lab1-b3	i-043471c5a6f1f79e3	Running	t2.micro	-	View alarms +	ap-south-1a	-	-	-

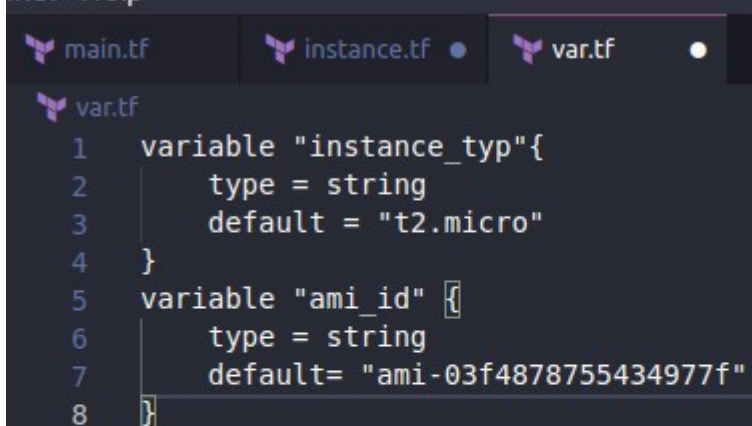
Step 2: Now we will create a var.tf file to create variable



```
main.tf
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.35.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "ap-south-1"
12   access_key = 
13   secret_key = 
14 }
```



```
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 }
```



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

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

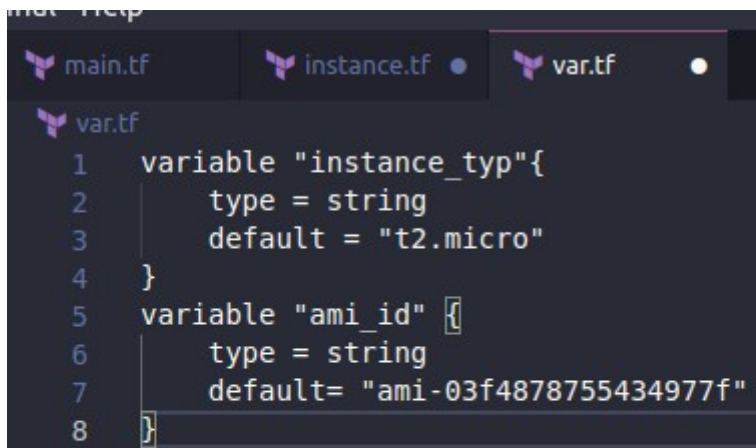
Step 3: To create multiple instances by changing instance.tf file

```
main.tf  x  instance.tf

main.tf
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.35.0"
6     }
7   }
8 }
9
10 provider "aws" {
11   region = "ap-south-1"
12   access_key = 
13   secret_key = 
14 }
```

```
main.tf  instance.tf  var.tf

instance.tf
1 resource "aws_instance" "lab4-1"{
2   instance_type= var.instance_typ
3   ami = var.ami_id
4   count =1
5   tags= {
6     Name = "lab4-b3-1"
7   }
8 }
9 resource "aws_instance" "lab4-2"{
10  instance_type= var.instance_typ
11  ami = var.ami_id
12  count =1
13  tags= {
14    Name = "lab4-b3-2"
15  }
16 }
17 resource "aws_instance" "lab4-3"{
18  instance_type= var.instance_typ
19  ami = var.ami_id
20  count =1
21  tags= {
22    Name = "lab4-b3-3"
23  }
24 }
```



The image shows a code editor window with three tabs: main.tf, instance.tf, and var.tf. The var.tf tab is active, displaying the following Terraform configuration:

```
1 variable "instance_typ"{
2     type = string
3     default = "t2.micro"
4 }
5 variable "ami_id" {
6     type = string
7     default= "ami-03f4878755434977f"
8 }
```

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