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

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

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

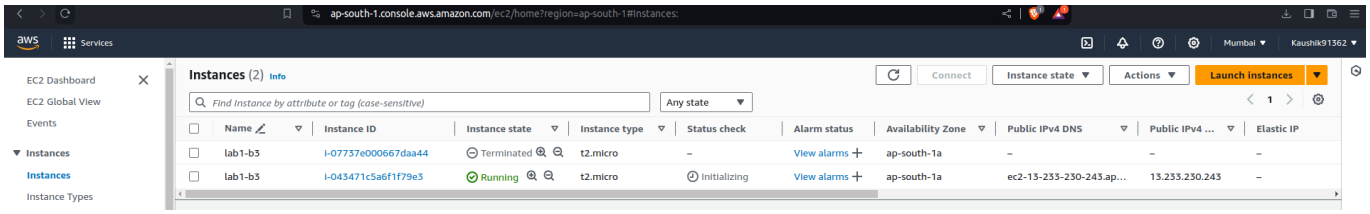
instance.tf

```
resource "aws_instance" "lab4"{
    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 = "ami-03f4878755434977f"
    }
}
```

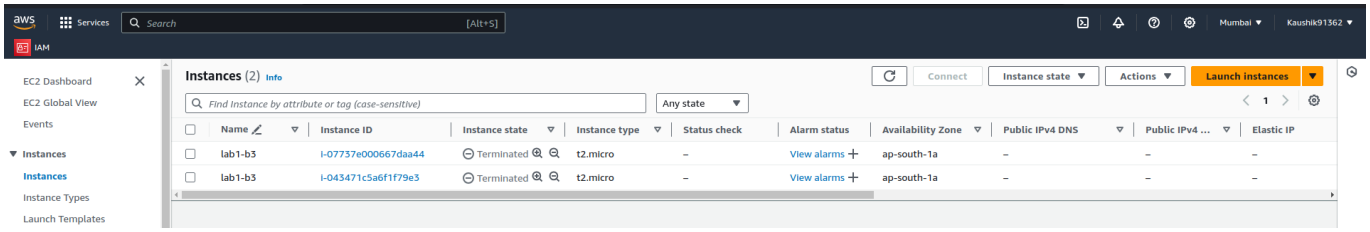
```
~/Documents/SPCM/Terraform v1.7.1default as
➜ terraform init

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                                 = {
```



```
~/Documents/SPCM/Terraform  v1.7.1default as  took 1m3s
→ terraform destroy
aws_instance.lab1[0]: Refreshing state... [id=i-043471c5a6f1f79e3]

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-1:698194348131:instance/i-043471c5a6f1f79e3" -> 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-043471c5a6f1f79e3" -> 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-0d6bdf6dc42345c16" -> null
      - private_dns                          = "ip-172-31-44-208.ap-south-1.compute.internal" -> null
      - private_ip                           = "172.31.44.208" -> null
      - public_dns                           = "ec2-13-233-230-243.ap-south-1.compute.amazonaws.com" -> null
      - public_ip                            = "13.233.230.243" -> null
      - secondary_private_ips                = [] -> null
      - security_groups                      = [
          - "default",
        ] -> null
      - source_dest_check                    = true -> null
      - subnet_id                            = "subnet-0d78b64e981bd0f9d" -> null
      - tags                                 = {
          - "Name" = "lab1-b3"
        } -> null
      - tags_all                             = {
          - "Name" = "lab1-b3"
```



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

main.tf

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

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

instance.tf

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

}
```

var.tf

```
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 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"
    }
}
```

main.tf instance.tf var.tf

var.tf

```
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.