

## 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  ×  instance.tf  var.tf

main.tf > provider "aws"
1  terraform {
2      required_providers {
3          aws = {
4              source = "hashicorp/aws"
5              version = "5.32.1"
6          }
7      }
8  }
9  provider "aws" {
10     region = "ap-south-1"
11     access_key = " "
12     secret_key = " "
13 }
```

```
main.tf  instance.tf  ×

instance.tf > variable "ami_id"
1  resource "aws_instance" "lab4" {
2      instance_type = var.instance_typ
3      ami = var.ami_id
4      count = 1
5      tags = {
6          Name = "lab4-b3"
7      }
8  }
9  variable "instance_typ" {
10     type = string
11     default = "t2.micro"
12 }
13 variable "ami_id" {
14     type = string
15     default = " "
16 }
```

```
arnim_taliyan@device:~/Desktop/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 v5.32.1

**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.

```
arnim_taliyan@device:~/Desktop/terraform$
```

```
arnim_taliyan@device:~/Desktop/terraform$ terraform validate  
Success! The configuration is valid.
```

```
arnim_taliyan@device:~/Desktop/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"
  + 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              = "instance"
  + 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)
```

```
arnim_taliyan@device:~/Desktop/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.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)  
}
```

```
+ 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]

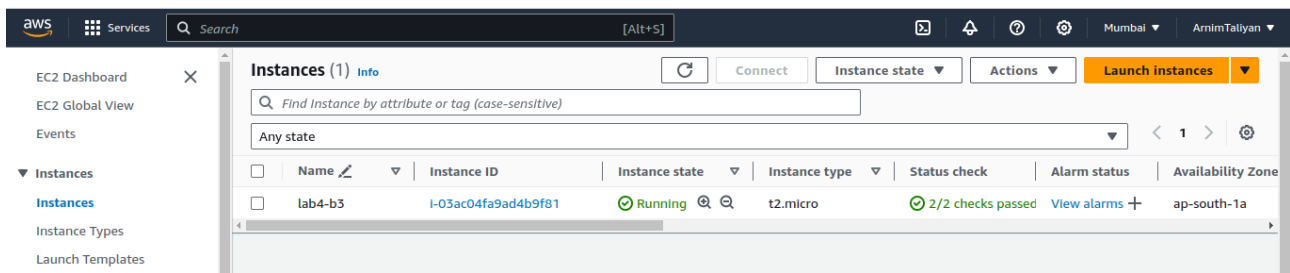
aws\_instance.lab4[0]: Still creating... [20s elapsed]

aws\_instance.lab4[0]: Still creating... [30s elapsed]

aws\_instance.lab4[0]: Creation complete after 33s [id=i-03ac04fa9ad4b9f81]

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

```
arnim_taliyan@device:~/Desktop/terraform$
```



```
arnim_taliyan@device:~/Desktop/terraform$ terraform destroy
aws_instance.lab4[0]: Refreshing state... [id=i-03ac04fa9ad4b9f81]
```

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-1:533266967718:instance/i-03ac04fa9ad4b9f81" -> 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-03ac04fa9ad4b9f81" -> 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-0201c3e243473c977" -> null
  - private_dns              = "ip-172-31-46-231.ap-south-1.compute.internal" -> null
  - private_ip               = "172.31.46.231" -> null
  - public_dns               = "ec2-13-233-110-172.ap-south-1.compute.amazonaws.com" -> null
  - public_ip               = "13.233.110.172" -> null
  - secondary_private_ips     = [] -> null
  - security_groups           = [
    - "default",
  ] -> null
}
```

```
- tags                      = {} -> null
- throughput               = 0 -> null
- volume_id                = "vol-0b51d19e3036f98af" -> 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-03ac04fa9ad4b9f81]
aws_instance.lab4[0]: Still destroying... [id=i-03ac04fa9ad4b9f81, 10s elapsed]
aws_instance.lab4[0]: Still destroying... [id=i-03ac04fa9ad4b9f81, 20s elapsed]
aws_instance.lab4[0]: Still destroying... [id=i-03ac04fa9ad4b9f81, 30s elapsed]
aws_instance.lab4[0]: Destruction complete after 30s
```

Destroy complete! Resources: 1 destroyed.

```
arnim_taliyan@device:~/Desktop/terraform$
```

aws

Services

Search

[Alt+S]

Mumbai

ArnimTaliyan

EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Instances (1) Info

Connect

Instance state

Actions

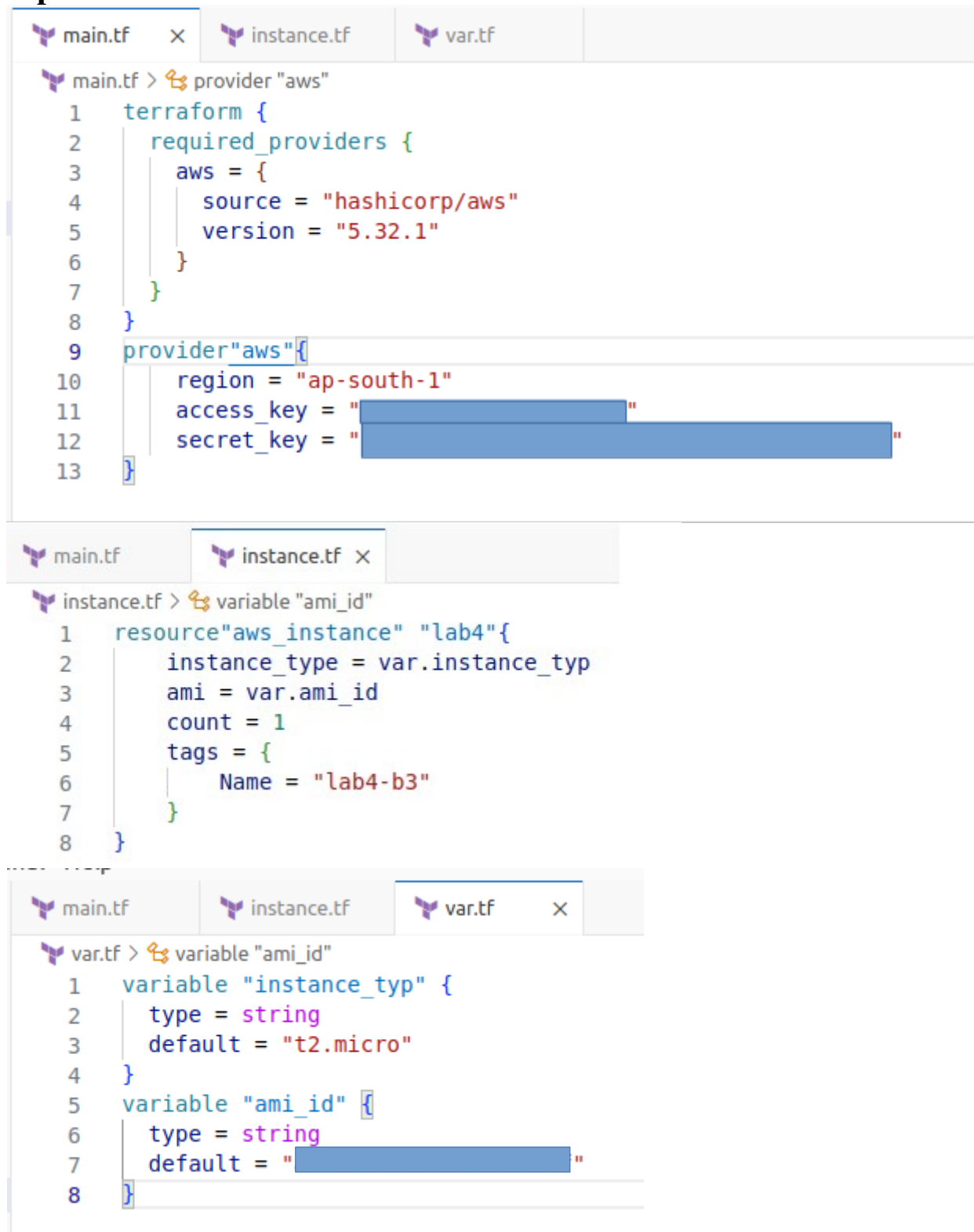
Launch instances

Find Instance by attribute or tag (case-sensitive)

Any state

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
	lab4-b3	i-03ac04fa9ad4b9f81	Terminated	t2.micro	-	<a href="#">View alarms</a>	ap-south-1

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



```
main.tf > provider "aws"
1 terraform {
2   required_providers {
3     aws = {
4       source = "hashicorp/aws"
5       version = "5.32.1"
6     }
7   }
8 }
9 provider "aws" {
10   region = "ap-south-1"
11   access_key = "
12   secret_key = "
13 }
```

```
instance.tf > variable "ami_id"
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 }
```

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

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

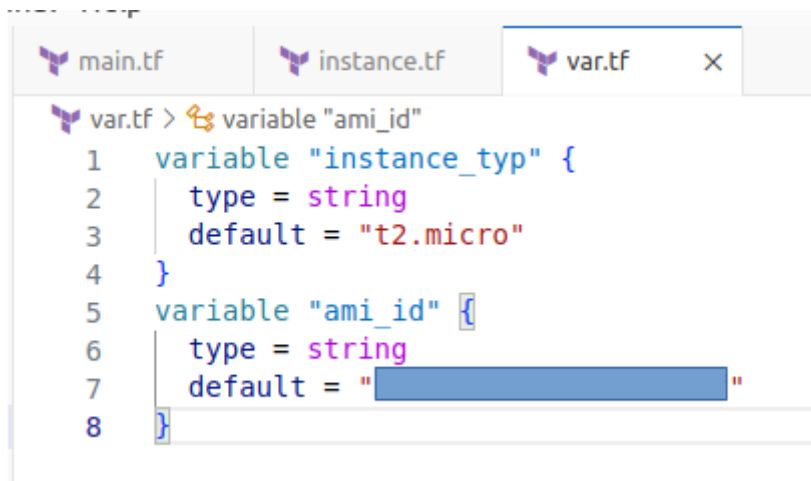
### Step 3: To create multiple instances by changing instance.tf file

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

main.tf > provider "aws"
1  terraform {
2      required_providers {
3          aws = {
4              source = "hashicorp/aws"
5              version = "5.32.1"
6          }
7      }
8  }
9  provider "aws" {
10     region = "ap-south-1"
11     access_key = "
12     secret_key = "
13 }
```

```
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
5      tags = {
6          Name = "lab4-b3-1"
7      }
8  }
9  resource "aws_instance" "lab4-2" {
10     instance_type = var.instance_type
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_type
19     ami = var.ami_id
20     count = 1
21     tags = {
22         Name = "lab4-b3-3"
23     }
24 }
```



```
main.tf instance.tf var.tf x
var.tf > variable "ami_id"
1  variable "instance_typ" {
2    type = string
3    default = "t2.micro"
4  }
5  variable "ami_id" {
6    type = string
7    default = ""
8  }
```

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