

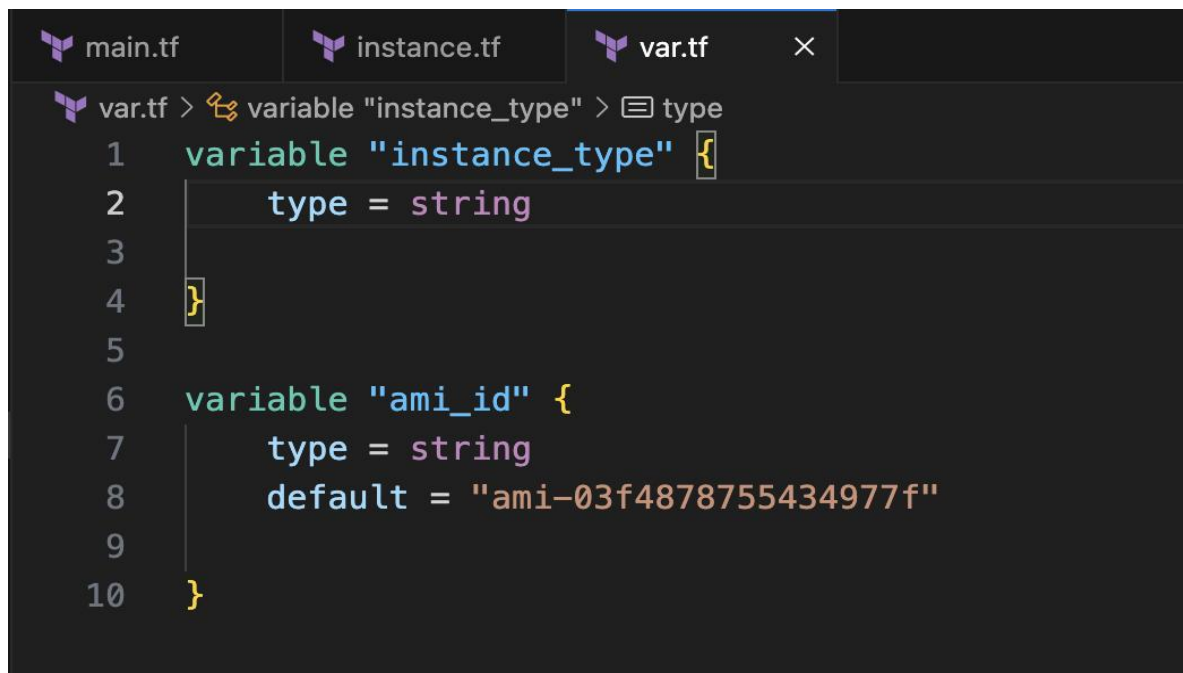
## LAB-5

# *Terraform Variable with Command Line Argument*

**Step1:** Make changes in var.tf file

```
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 }
```

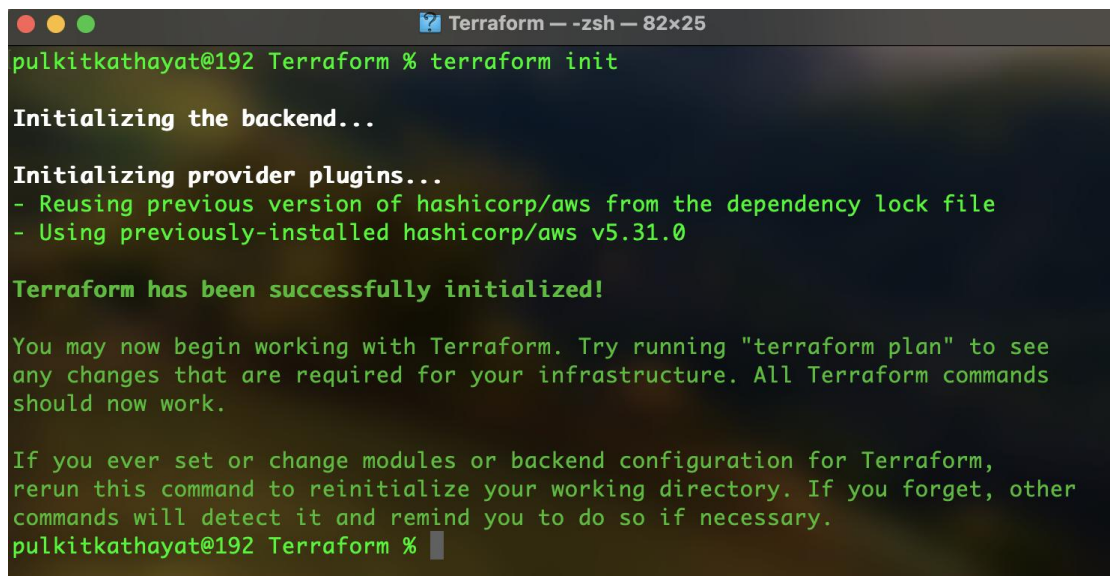
```
main.tf instance.tf × var.tf
instance.tf > resource "aws_instance" "lab4-1" > tags > Name
1 resource "aws_instance" "lab4-1" {
2   instance_type = var.instance_type
3   ami = var.ami_id
4   count = 1
5   tags = {
6     Name = "lab4-b3"
7   }
8 }
9
10
```



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

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

**Step 2:** Now we need to run terraform cycle



The screenshot shows a terminal window titled "Terraform -- zsh -- 82x25". The user has entered the command `terraform init`. The output is as follows:

```
pulkitkathayat@192 Terraform % terraform init

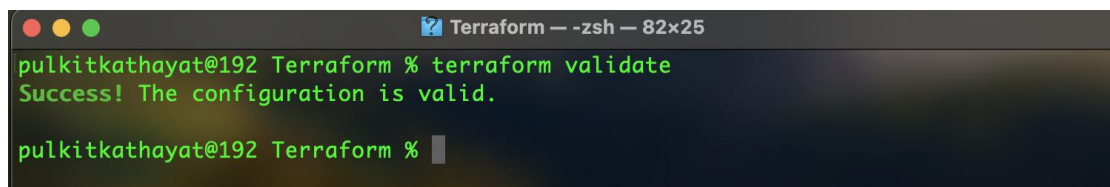
Initializing the backend...

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 %
```



The screenshot shows a terminal window titled "Terraform -- zsh -- 82x25". The user has entered the command `terraform validate`. The output is as follows:

```
pulkitkathayat@192 Terraform % terraform validate
Success! The configuration is valid.

pulkitkathayat@192 Terraform %
```

Now we have to ways to declare variable in CLI First: We can give value after running terraform plan

```
pulkitkathayat@192 Terraform % terraform plan
var.instance_type
Enter a value: t2.micro

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.lab4-1[0] will be created
+ resource "aws_instance" "lab4-1" {
  + 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)
```

## Second: By declaring variable during running command

```
pulkitkathayat@192 Terraform % terraform plan -var 'instance_type=t2.micro'

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.lab4-1[0] will be created
+ resource "aws_instance" "lab4-1" {
  + 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)
}
```



```
pulkitkathayat@192 Terraform % terraform apply
var.instance_type
Enter a value: t2.micro
```

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.lab4-1[0] will be created
+ resource "aws_instance" "lab4-1" {
  + 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.

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.lab4-1[0]: Creating...
aws_instance.lab4-1[0]: Still creating... [10s elapsed]
aws_instance.lab4-1[0]: Still creating... [20s elapsed]
aws_instance.lab4-1[0]: Still creating... [30s elapsed]
aws_instance.lab4-1[0]: Creation complete after 32s [id=i-05133db0d8bffa6d3]
```

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.  
pulkitkathayat@192 Terraform %

aws Services Search [Option+S]

EC2 Dashboard EC2 Global View Events

Instances Instances Instance Types Launch Templates Spot Requests

Instances (2) info

Find Instance by attribute or tag (case-sensitive) Any state

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	lab4-b5	i-05133db0d8b06d3	Running	t2.micro	Initializing	View alarms	ap-south-1b	ec2
<input type="checkbox"/>	lab4-b3	i-04e1e93a57dd54b1e	Terminated	t2.micro	-	View alarms	ap-south-1b	-

```
pulkitkathayat@192 Terraform % terraform destroy
var.instance_type
Enter a value: t2.micro

aws_instance.lab4-1[0]: Refreshing state... [id=i-05133db0d8b06d3]

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.lab4-1[0] will be destroyed
- resource "aws_instance" "lab4-1" {
  ami                    = "ami-03f4878755434977f" -> null
  arn                    = "arn:aws:ec2:ap-south-1:225999921982:instance/i-05133db0d8b06d3" -> 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-05133db0d8b06d3" -> 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-00299f752204ddb0" -> null
  private_dns             = "ip-172-31-0-17.ap-south-1.compute.internal" -> null
  private_ip              = "172.31.0.17" -> null
  public_dns              = "ec2-13-233-68-98.ap-south-1.compute.amazonaws.com" -> null
  public_ip               = "13.233.68.98" -> 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
    }

    - 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
      - tags                  = {} -> null
      - throughput            = 0 -> null
      - volume_id             = "vol-069c57647b06c5740" -> 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.lab4-1[0]: Destroying... [id=i-05133db0d8bff06d3]
aws_instance.lab4-1[0]: Still destroying... [id=i-05133db0d8bff06d3, 10s elapsed]
aws_instance.lab4-1[0]: Still destroying... [id=i-05133db0d8bff06d3, 20s elapsed]
aws_instance.lab4-1[0]: Still destroying... [id=i-05133db0d8bff06d3, 30s elapsed]
aws_instance.lab4-1[0]: Destruction complete after 31s

```

**Destroy complete! Resources: 1 destroyed.**

pulkitkathayat@192 Terraform %

Instances (2) <small>Info</small>								
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>			Any state		< 1 >			
<input type="checkbox"/>	Name <small>↗</small>	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pul
<input type="checkbox"/>	lab4-b3	i-05133db0d8bff06d3	Terminated	t2.micro	-	<a href="#">View alarms</a>	ap-south-1b	-
<input type="checkbox"/>	lab4-b3	i-04e1e93a57dd54b1e	Terminated	t2.micro	-	<a href="#">View alarms</a>	ap-south-1b	-