

LAB-5

Terraform Variable with Command Line Argument

Step1: Make changes in var.tf file



The screenshot displays the Terraform IDE interface with three files open: main.tf, instance.tf, and var.tf.

main.tf contains the following configuration:

```
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 contains the following configuration:

```
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 contains the following configuration:

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

Step 2: Now we need to run terraform cycle

```
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.
```

Now we have to ways to declare variable in CLI

First: We can give value after running terraform plan

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

Second: By declaring variable during running command

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

```
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)
```

The screenshot shows the AWS Management Console interface. On the left is a navigation menu with options like 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Instances', 'Instance Types', and 'Launch Templates'. The main content area is titled 'Instances (1)' and includes a search bar and a filter dropdown set to 'Any state'. Below this is a table listing the instance 'lab4-b3' with its ID 'i-03ac04fa9ad4b9f81', state 'Running', type 't2.micro', and availability zone 'ap-south-1a'. The status check indicates '2/2 checks passed'.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	lab4-b3	i-03ac04fa9ad4b9f81	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a

```
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
```

The screenshot shows the AWS Management Console interface. On the left, there is a navigation menu with options like 'EC2 Dashboard', 'EC2 Global View', 'Events', and 'Instances'. The main panel displays the 'Instances (1)' page. At the top, there are buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. Below these is a search bar and a filter dropdown set to 'Any state'. A table lists the instance details:

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