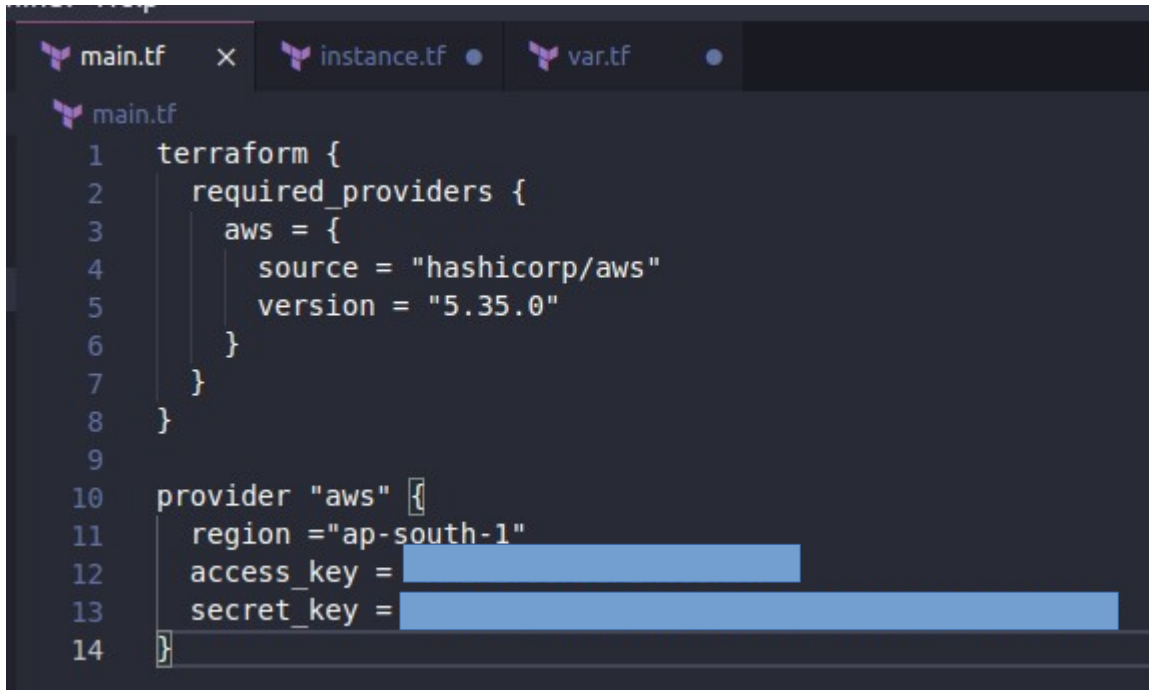


LAB-5

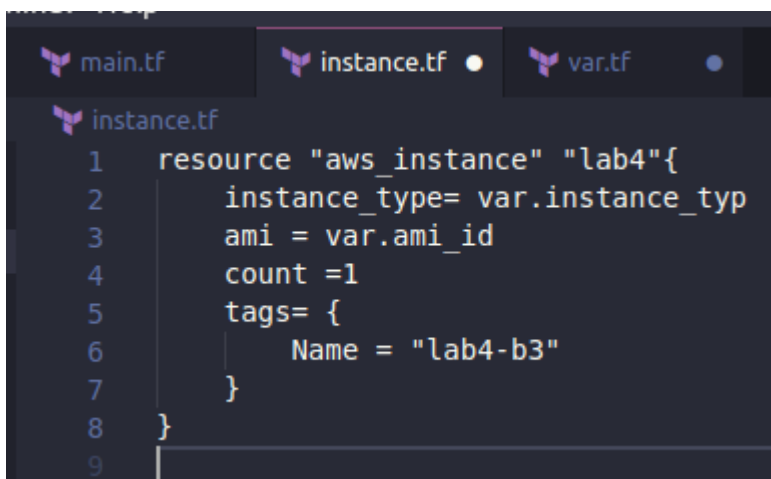
Terraform Variable with Command Line Argument

Step1: Make changes in var.tf file



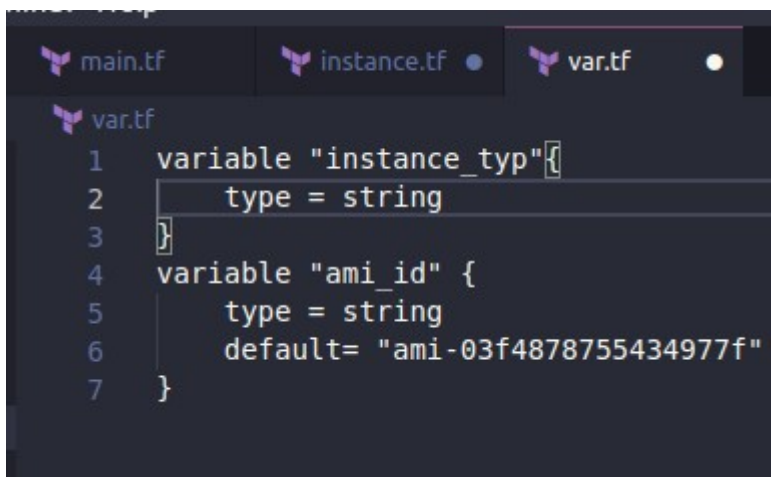
This screenshot shows the `main.tf` file in a code editor. The file contains Terraform configuration for the AWS provider. It includes a `terraform` block with `required_providers` and a `provider` block for AWS. The `access_key` and `secret_key` fields in the provider block are redacted with blue bars.

```
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 = [REDACTED]
13   secret_key = [REDACTED]
14 }
```



This screenshot shows the `instance.tf` file in a code editor. It defines an `aws_instance` resource named `lab4`. The resource uses variables `var.instance_type` and `var.ami_id`. The `tags` block includes a `Name` tag with the value `"lab4-b3"`.

```
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
```



This screenshot shows the `var.tf` file in a code editor. It defines two variables: `instance_type` and `ami_id`. Both variables are of type `string`. The `ami_id` variable has a default value of `"ami-03f4878755434977f"`.

```
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

```
~/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 4s
→ 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

```
~/Documents/SPCM/Terraform v1.7.1default as took 4s
→ 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.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_index = (known after apply)
```

Second: By declaring variable during running command

```
~/Documents/SPCH/Terraform v1.7.1default as took 30s
→ 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.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)
}
```

```
~/Documents/SPCH/Terraform v1.7.1default as took 5s
→ 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.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)
}
```

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

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
lab1-b3	i-07737e00667daa44	Terminated	t2.micro	-	View alarms +	ap-south-1a	-	-	-
lab1-b3	i-043471c5a6f1f79e3	Terminated	t2.micro	-	View alarms +	ap-south-1a	-	-	-
lab1-b3	i-004985016820cb298	Running	t2.micro	Initializing	View alarms +	ap-south-1a	ec2-3-111-215-151.ap-...	3.111.215.151	-


```
~/Documents/SPCM/Terraform v1.7.1default as
→ terraform destroy
var.instance_type
  Enter a value: t2.micro

aws_instance.lab1[0]: Refreshing state... [id=i-004985016820cb298]

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-004985016820cb298" -> 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-004985016820cb298" -> 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-0acdffa7b21b482333" -> null
  - private_dns              = "ip-172-31-43-111.ap-south-1.compute.internal" -> null
  - private_ip               = "172.31.43.111" -> null
  - public_dns               = "ec2-3-111-215-151.ap-south-1.compute.amazonaws.com" -> null
  - public_ip                = "3.111.215.151" -> null
  - secondary_private_ips     = [] -> null
  - security_groups           = [
    "default"
  ]
}
```

aws

EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Instances (3) Info

Find Instance by attribute or tag (case-sensitive)

Any state

Refresh

Connect

Instance state

Actions

Launch Instances

< 1 >

Reset

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IF
<input type="checkbox"/>	lab1-b3	i-07737e000667daa44	Terminated	t2.micro	...	View alarms +	ap-south-1a	-	-	...	-
<input type="checkbox"/>	lab1-b3	i-043471c5a6f1f79e3	Terminated	t2.micro	...	View alarms +	ap-south-1a	-	-	...	-
<input type="checkbox"/>	lab1-b3	i-004985016820cb298	Terminated	t2.micro	...	View alarms +	ap-south-1a	-	-	...	-