

## Lab Exercise 4

### Terraform Variables

#### 1. Create a Terraform Directory:

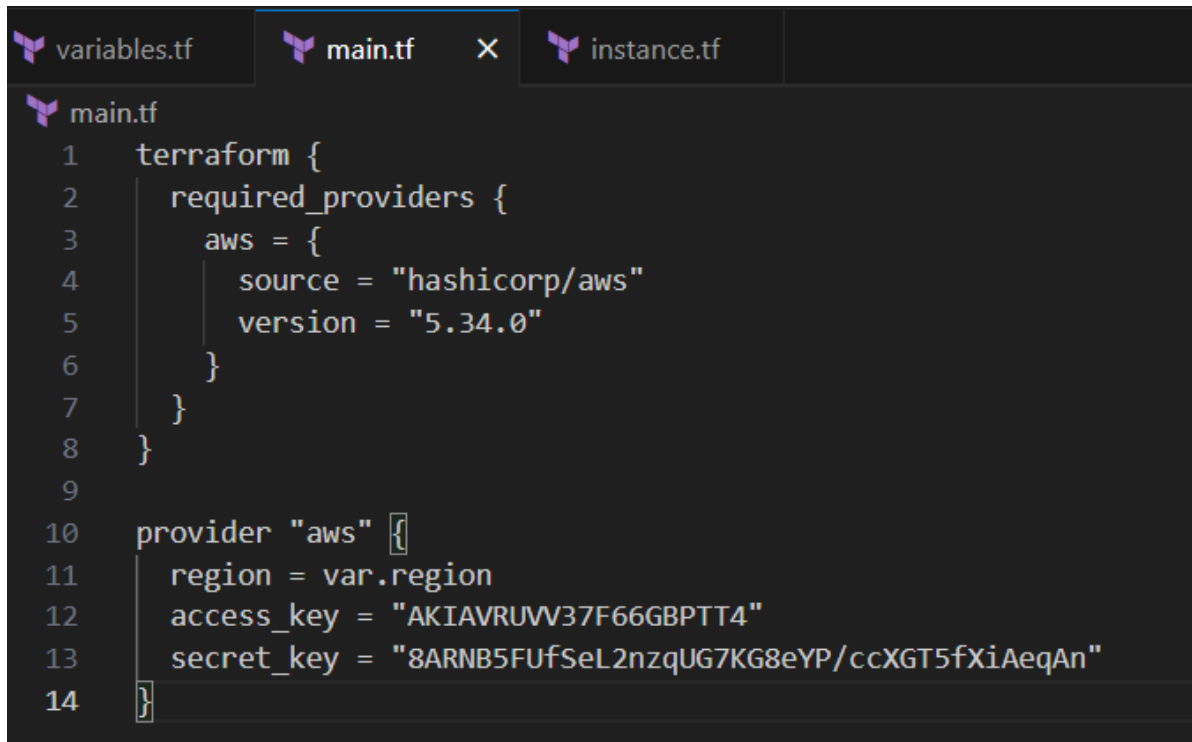
```
PS E:\> mkdir terraform-variable

Directory: E:\

Mode                LastWriteTime         Length Name
----                -
d-----          06-02-2024    21:36             terraform-variable

PS E:\> cd .\terraform-variable\
PS E:\terraform-variable> |
```

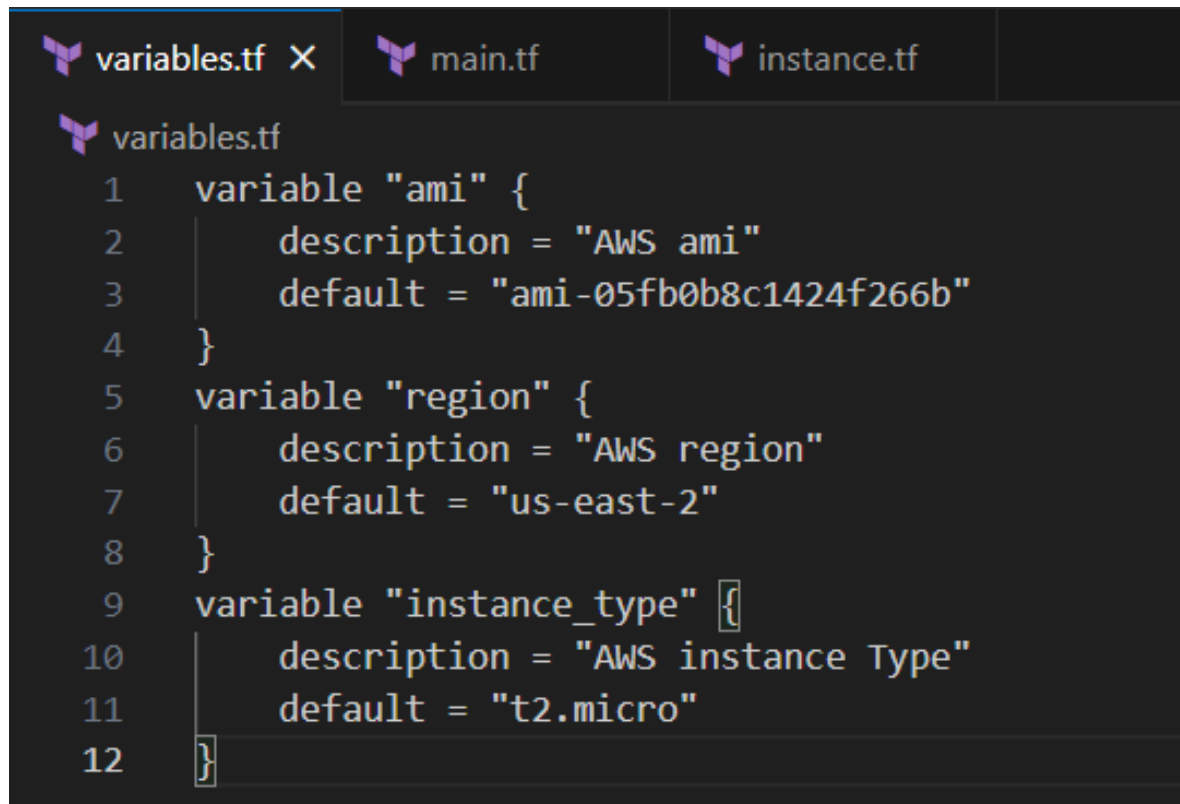
#### 2. Create a Terraform Configuration File:



```
variables.tf  main.tf  instance.tf

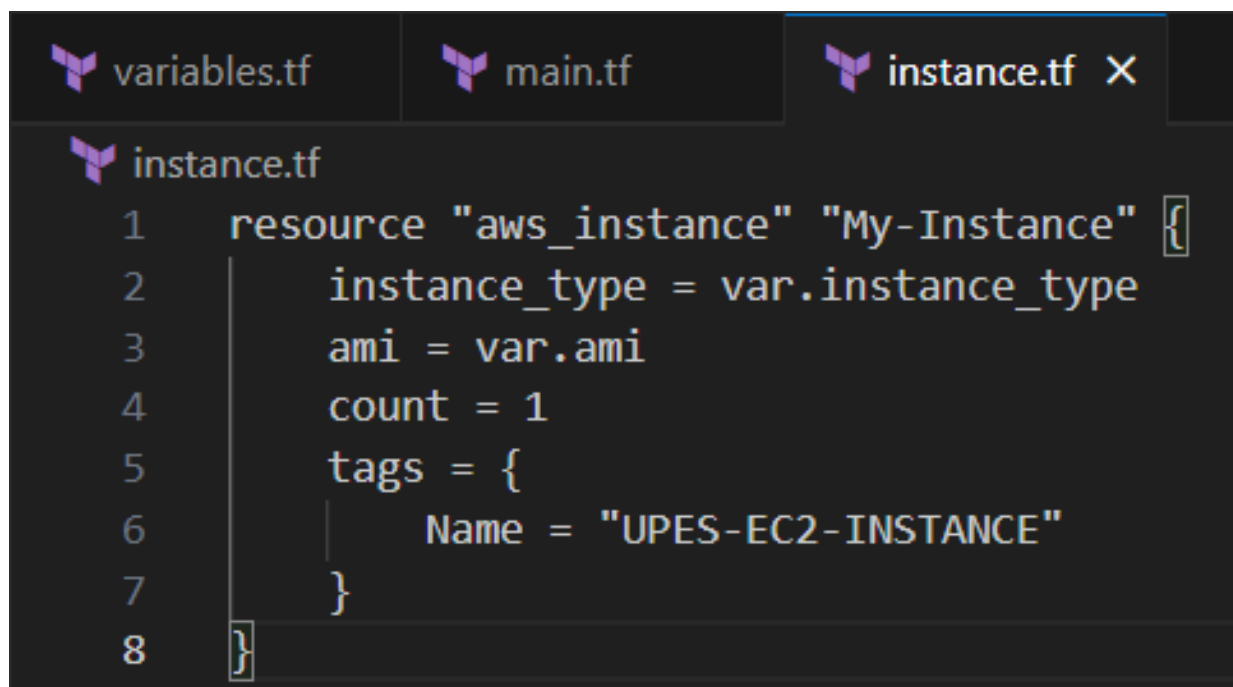
main.tf
1 terraform {
2     required_providers {
3         aws = {
4             source = "hashicorp/aws"
5             version = "5.34.0"
6         }
7     }
8 }
9
10 provider "aws" {
11     region = var.region
12     access_key = "AKIAVRUVV37F66GBPTT4"
13     secret_key = "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"
14 }
```

### 3. Define Variables:



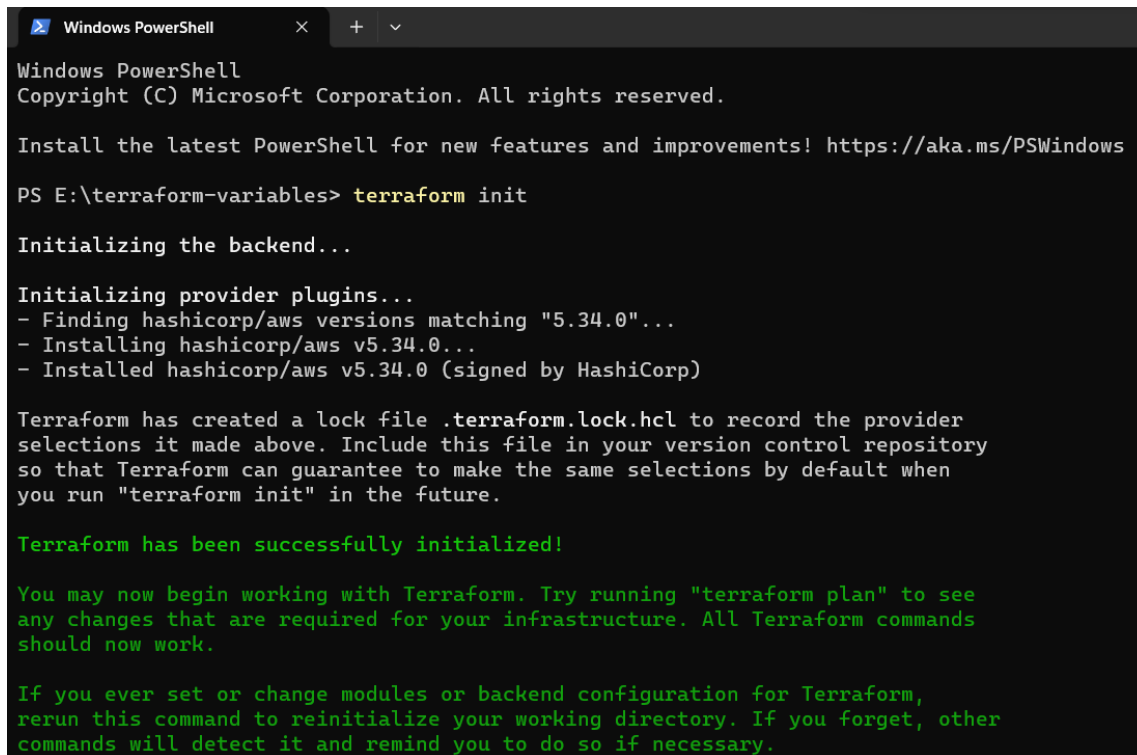
```
variables.tf
1  variable "ami" {
2      description = "AWS ami"
3      default = "ami-05fb0b8c1424f266b"
4  }
5  variable "region" {
6      description = "AWS region"
7      default = "us-east-2"
8  }
9  variable "instance_type" {
10     description = "AWS instance Type"
11     default = "t2.micro"
12 }
```

### 4. Use Variables in instance.tf:



```
instance.tf
1  resource "aws_instance" "My-Instance" {
2      instance_type = var.instance_type
3      ami = var.ami
4      count = 1
5      tags = {
6          Name = "UPES-EC2-INSTANCE"
7      }
8  }
```

## 5. Initialize and Apply:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\terraform-variables> terraform init

Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/aws versions matching "5.34.0"...
- Installing hashicorp/aws v5.34.0...
- Installed hashicorp/aws v5.34.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.

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.
```

```
Windows PowerShell
PS E:\terraform-variables> 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.My-Instance[0] will be created
+ resource "aws_instance" "My-Instance" {
  + ami                    = "ami-05fb0b8c1424f266b"
  + 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)
```

Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

Any state

Instance state = running

Clear filters

< 1 >

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input type="checkbox"/>	UPES-EC2-INS...	i-00c748c380161c593	<span>Running</span>	t2.micro	<span>Initializing</span>	<a href="#">View alarms +</a>	us-east-2a	ec2-18-191-14-9

## 6. Clean Up:

```

- 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-04326ca5898c93d1d" -> 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.My-Instance[0]: Destroying... [id=i-00c748c380161c593]
aws_instance.My-Instance[0]: Still destroying... [id=i-00c748c380161c593, 10s elapsed]
aws_instance.My-Instance[0]: Still destroying... [id=i-00c748c380161c593, 20s elapsed]
aws_instance.My-Instance[0]: Still destroying... [id=i-00c748c380161c593, 30s elapsed]
aws_instance.My-Instance[0]: Destruction complete after 32s

```

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

PS E:\terraform-variables> |

Instances (1) <a href="#">Info</a>							
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>		<input type="text" value="Any state"/>		<span>&lt; 1 &gt;</span> <a href="#">⚙</a>			
<input type="checkbox"/>	Name ↗	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	UPES-EC2-INS...	i-00c748c380161c593	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-2a