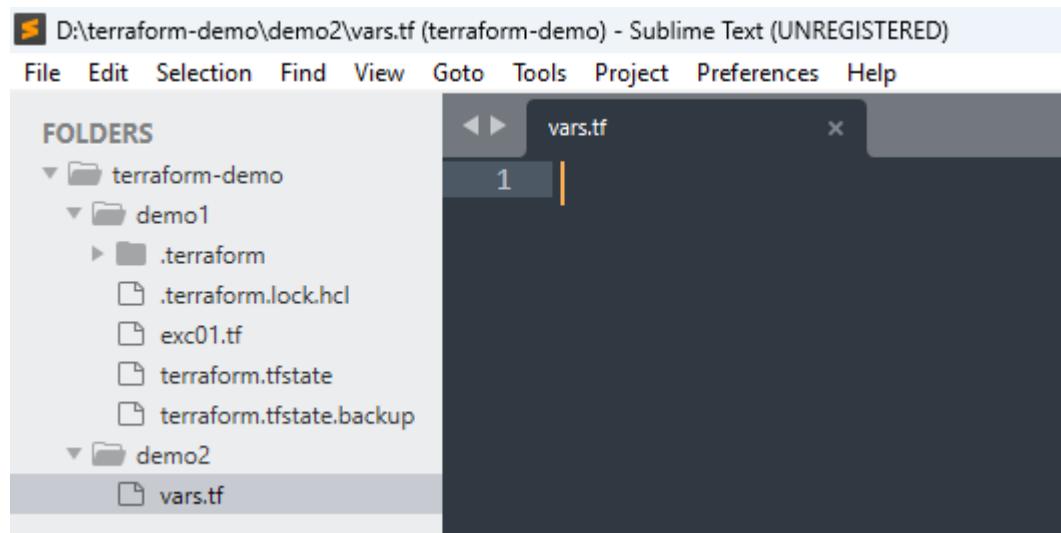


Variables

1. In this lab you are going to see terraform way of consuming variables.
2. So, variables as we know helps us move critical or confidential data out of our script. Also, if there are some values that changes based on your environment, on your project, you can define them as variable. So, you don't need to make much change in your code. You can just change the value of variables. Or if you're again reusing it, you want to reuse your code for different environment, different project. So, you can define things like AMI, tags, key pair these things as variables.
3. First you are going to create a new directory for this lab. Again, go to your code editor now right click on your terraform-demo folder and create a sub folder out of it like you did in the previous lab.
4. Then in that new folder create a new file with name **vars.tf** in this file you are going to describe your variables.



5. Now in vars.tf file you are going to write the code shown below. In this code you described your region, the available zone in that particular region and the AMI ID available.

D:\terraform-demo\demo2\vars.tf • (terraform-demo) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- terraform-demo
 - demo1
 - .terraform
 - .terraform.lock.hcl
 - exc01.tf
 - terraform.tfstate
 - terraform.tfstate.backup
 - demo2
 - vars.tf

vars.tf

```
1 variable "REGION" {  
2   default = "ap-south-1"  
3 }  
4  
5 variable "ZONE1" {  
6   default = "ap-south-1a"  
7 }  
8  
9 variable "AMIS" {  
10  type = map(any)  
11  default = {  
12    ap-south-1 = "ami-0e670eb768a5fc3d4"  
13  }  
14 }
```

- Now you are going to create a new file with name providers.tf and you are going to write this code in it.

D:\terraform-demo\demo2\providers.tf (terraform-demo) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- terraform-demo
 - demo1
 - .terraform
 - .terraform.lock.hcl
 - exc01.tf
 - terraform.tfstate
 - terraform.tfstate.backup
 - demo2
 - providers.tf
 - vars.tf

vars.tf

providers.tf

```
1 provider "aws" {  
2   region = var.REGION  
3 }
```

- Again, create a new file in the same folder and this time you are going to write the main code for this lab.

D:\terraform-demo\demo2\inst.tf (terraform-demo) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

FOLDERS

- terraform-demo
 - demo1
 - .terraform
 - .terraform.lock.hcl
 - exc01.tf
 - terraform.tfstate
 - terraform.tfstate.backup
 - demo2
 - inst.tf
 - providers.tf
 - vars.tf

vars.tf providers.tf inst.tf

```
1 resource "aws_instance" "demo-instance" {
2   ami           = var.AMIS[var.REGION]
3   instance_type = "t2.micro"
4   availability_zone = var.ZONE1
5   key_name      = "demo-terraform"
6   vpc_security_group_ids = ["sg-08be2f03824940936"]
7   tags = {
8     Name        = "Demo-Instance-Terraform"
9     Environment = "Terraform"
10  }
11
12 }
```

8. Once this is all done and saved into your folder. Now you need to open gitbash and run the commands.
9. First go in to the new folder. Then initialize the code.

terraform init

```
$ terraform.exe init

Initializing the backend...

Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.38.0...
- Installed hashicorp/aws v5.38.0 (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.
```

10. Now you are going to validate the code.

terraform validate

```
LAPTOP-G2CAKBK8 MINGW64 /d/terraform-demo/demo2
$ terraform validate
Success! The configuration is valid.
```

11. Then format it properly using this command.

terraform fmt

12. Once the formatting is done now you need to check the plan.

terraform plan

```
+ 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 = [
  + "sg-08be2f03824940936",
]
}

Plan: 1 to add, 0 to change, 0 to destroy.
```

13. Now run the execution command which is

terraform apply

```
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.demo-instance: Creating...
aws_instance.demo-instance: Still creating... [10s elapsed]
aws_instance.demo-instance: Still creating... [20s elapsed]
aws_instance.demo-instance: Still creating... [30s elapsed]
aws_instance.demo-instance: Creation complete after 32s [id=i-057109ae645d39d54]

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

14. Now navigate to AWS Console and you will see that your instance has been created but time you have used variables.

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Available
Demo-Instance-Terraform	i-057109ae645d39d54	Running	t2.micro	Initializing	View alarms	ap-south-1

Instance: i-057109ae645d39d54 (Demo-Instance-Terraform)

Details Status and alarms New Monitoring Security Networking Storage Tags

Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-057109ae645d39d54 (Demo-Instance-Terraform)	13.233.61.83 [open address]	172.31.47.187
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-13-233-61-83.ap-south-1.compute.amazonaws.com [open address]

Instance: i-057109ae645d39d54 (Demo-Instance-Terraform)

Details Status and alarms New Monitoring Security Networking Storage Tags

Tags

Key	Value
Name	Demo-Instance-Terraform
Environment	Terraform

15. Once you are done just run the command shown below, it will destroy all the resources which were created.

terraform 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.demo-instance: Destroying... [id=i-057109ae645d39d54]
aws_instance.demo-instance: Still destroying... [id=i-057109ae645d39d54, 10s elapsed]
aws_instance.demo-instance: Still destroying... [id=i-057109ae645d39d54, 20s elapsed]
aws_instance.demo-instance: Still destroying... [id=i-057109ae645d39d54, 30s elapsed]
aws_instance.demo-instance: Destruction complete after 30s

Destroy complete! Resources: 1 destroyed.
```