

Terraform part II:

terraform init
terraform fmt
terraform validate
terraform plan
terraform apply
terraform destroy

inputs.tf, main.tf, provider.tf, output.tf

=> Terraform modules:

Directory:

01-tf-project (one directory where we are creating is called as Module)

provider.tf
inputs.tf
main.tf
output.tf

Module is a set of Terraform configuration files in a single directory

In one directory, what are all Terraform configuration files are there is only called as Module
We specify in the 'provider', which resources to use

Whatever we want to see after the Terraform script is executed is displayed in the Output section

Multiple resources I want to write on the same cloud: S3, RDS, VPC, IAM --> All of them in main.tf itself

that's why we create multiple child modules under one Module

How to write in a better way so it aligns with industry standards

Say we want to create one directory my-infra-app: root module

Within root only, I will create a file 'provider.tf'

Next we create the modules --> example : Child module:1 EC2 (input.tf, main.tf, output.tf), Child module:2 S3 (input.tf, main.tf, output.tf) ,

In this, we are not writing all resources into one single file, we write in modules

In the root directory only, we create main.tf --> invoke the child modules

In the root, we have output.tf --> invoke modules of output variables

A Terraform module is a set of Terraform configuration files in a single directory

Any directory with one or more configuration files is called as "One module"

Single directory with one or more .tf files even for a simple configuration can be considered as a module

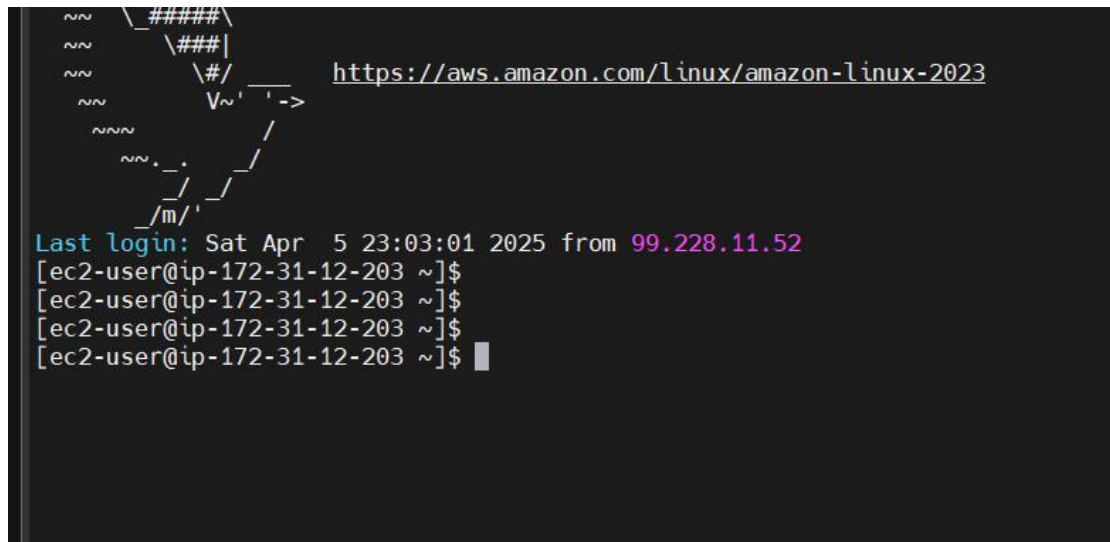
One root module can have any number of child modules in Terraform

Example: Inside one project or module, we can create multiple child modules EC2, S3, RDS, IAM as child modules

Note: We will run terraform commands from root module we will invoke child modules for execution

Terraform project setup with modules

Open Terraform VM



Create Modules folder

```
[ec2-user@ip-172-31-12-203 ~]$ mkdir 05-terraform-modules-project
[ec2-user@ip-172-31-12-203 ~]$ cd 05-terraform-modules-project/
[ec2-user@ip-172-31-12-203 05-terraform-modules-project]$ vi provider.tf
[ec2-user@ip-172-31-12-203 05-terraform-modules-project]$ mkdir modules
```

Two folders inside modules

```
[ec2-user@ip-172-31-12-203 modules]$
[ec2-user@ip-172-31-12-203 modules]$ ls -l
total 0
drwxr-xr-x. 2 ec2-user ec2-user 6 Apr 6 22:10 ec2
drwxr-xr-x. 2 ec2-user ec2-user 6 Apr 6 22:10 s3

[ec2-user@ip-172-31-12-203 modules]$ cd ec2
[ec2-user@ip-172-31-12-203 ec2]$ ls -l
total 0
[ec2-user@ip-172-31-12-203 ec2]$
[ec2-user@ip-172-31-12-203 ec2]$
[ec2-user@ip-172-31-12-203 ec2]$ touch inputs.tf
[ec2-user@ip-172-31-12-203 ec2]$ touch main.tf
[ec2-user@ip-172-31-12-203 ec2]$ touch outputs.tf
[ec2-user@ip-172-31-12-203 ec2]$

[ec2-user@ip-172-31-12-203 modules]$ cd ec2
[ec2-user@ip-172-31-12-203 ec2]$ ls -l
total 0
-rw-r--r--. 1 ec2-user ec2-user 0 Apr 6 22:26 inputs.tf
-rw-r--r--. 1 ec2-user ec2-user 0 Apr 6 22:26 main.tf
-rw-r--r--. 1 ec2-user ec2-user 0 Apr 6 22:26 outputs.tf

[ec2-user@ip-172-31-12-203 s3]$ ls -l
total 0
-rw-r--r--. 1 ec2-user ec2-user 0 Apr 6 22:28 inputs.tf
```

```
-rw-r--r--. 1 ec2-user ec2-user 0 Apr  6 22:28 main.tf
-rw-r--r--. 1 ec2-user ec2-user 0 Apr  6 22:28 outputs.tf
```

Same for S3 folder also

```
[ec2-user@ip-172-31-12-203 modules]$ cd s3
[ec2-user@ip-172-31-12-203 s3]$
[ec2-user@ip-172-31-12-203 s3]$
[ec2-user@ip-172-31-12-203 s3]$
[ec2-user@ip-172-31-12-203 s3]$ ls -l
total 0
-rw-r--r--. 1 ec2-user ec2-user 0 Apr  6 22:28 inputs.tf
-rw-r--r--. 1 ec2-user ec2-user 0 Apr  6 22:28 main.tf
-rw-r--r--. 1 ec2-user ec2-user 0 Apr  6 22:28 outputs.tf
```

Root directory:

```
[ec2-user@ip-172-31-12-203 05-terraform-modules-project]$ ls -l
total 4
drwxr-xr-x. 4 ec2-user ec2-user 27 Apr  6 22:10 modules
-rw-r--r--. 1 ec2-user ec2-user 254 Apr  6 22:07 provider.tf
```

We have modules folder and provider.tf file

```
[ec2-user@ip-172-31-12-203 05-terraform-modules-project]$ ls -l modules/
total 0
drwxr-xr-x. 2 ec2-user ec2-user 56 Apr  6 22:26 ec2
drwxr-xr-x. 2 ec2-user ec2-user 56 Apr  6 22:28 s3
```

```
[ec2-user@ip-172-31-12-203 ~]$ tree 05-terraform-modules-project/
05-terraform-modules-project/
├── modules
│   ├── ec2
│   │   ├── inputs.tf
│   │   ├── main.tf
│   │   └── outputs.tf
│   └── s3
│       ├── inputs.tf
│       ├── main.tf
│       └── outputs.tf
└── provider.tf
```

```
[ec2-user@ip-172-31-12-203 ~]$  
[ec2-user@ip-172-31-12-203 ~]$ tree 05-terraform-modules-project/  
05-terraform-modules-project/  
├── modules  
│   ├── ec2  
│   │   ├── inputs.tf  
│   │   ├── main.tf  
│   │   └── outputs.tf  
│   └── s3  
│       ├── inputs.tf  
│       ├── main.tf  
│       └── outputs.tf  
└── provider.tf  
  
3 directories, 7 files
```

This is the structure

```
[ec2-user@ip-172-31-12-203 ~]$ tree 05-terraform-modules-project/  
05-terraform-modules-project/  
├── modules  
│   ├── ec2  
│   │   ├── inputs.tf  
│   │   ├── main.tf  
│   │   └── outputs.tf  
│   └── s3  
│       ├── inputs.tf  
│       ├── main.tf  
│       └── outputs.tf  
└── provider.tf
```

```
[ec2-user@ip-172-31-12-203 ~]$  
[ec2-user@ip-172-31-12-203 ~]$ tree 05-terraform-modules-project/  
05-terraform-modules-project/  
├── modules  
│   ├── ec2  
│   │   ├── inputs.tf  
│   │   ├── main.tf  
│   │   └── outputs.tf  
│   └── s3  
│       ├── inputs.tf  
│       ├── main.tf  
│       └── outputs.tf  
└── provider.tf
```

```
[ec2-user@ip-172-31-12-203 ~]$ ls -l  
total 0  
drwxr-xr-x. 4 ec2-user ec2-user 152 Apr  5 20:46 01-tf-script  
drwxr-xr-x. 4 ec2-user ec2-user 186 Apr  5 22:45 01-tf-script-userdata  
drwxr-xr-x. 3 ec2-user ec2-user 180 Apr  6 00:50 04-tf-script-var  
drwxr-xr-x. 3 ec2-user ec2-user 40 Apr  6 22:08 05-terraform-modules-project  
drwxr-xr-x. 3 ec2-user ec2-user 159 Apr  6 01:56 05-tf-script-var  
-rw-r--r--. 1 ec2-user ec2-user  0 Apr  7 00:02 main.tf
```

```
[ec2-user@ip-172-31-12-203 05-terraform-modules-project]$ cat provider.tf
provider "aws" {
  region = "ca-central-1" # or your preferred AWS region like us-east-1, etc.

  # Optional if you have AWS credentials configured via CLI or environment variables
  # access_key = "YOUR_ACCESS_KEY"
  # secret_key = "YOUR_SECRET_KEY"
}
```

```
[ec2-user@ip-172-31-12-203 ec2]$ cat inputs.tf
variable "ami" {
  description = "Amazon vm image value"
  type        = string
}
```

```
variable "instance_type" {
  description = "Represents the type of instance"
  default     = "t2.micro"
}
```

```
[ec2-user@ip-172-31-12-203 ec2]$ cat main.tf
resource "aws_instance" "linux_vm" {
  ami           = var.ami
  instance_type = var.instance_type
  key_name      = "DevOpsMar30"
  security_groups = ["default"]
  tags = {
    Name = "modules-Linux-VM"
  }
}
```

```
[ec2-user@ip-172-31-12-203 ec2]$ cat outputs.tf
output "ec2_vm_public_ip" {
  value = aws_instance.linux_vm.public_ip
}
```

```
output "ec2_vm_private_ip" {
  value = aws_instance.linux_vm.private_ip
}
```

Now we are going into S3

```
[ec2-user@ip-172-31-12-203 ec2]$ cd ..
[ec2-user@ip-172-31-12-203 modules]$ cd s3
[ec2-user@ip-172-31-12-203 s3]$ ls -l
total 0
-rw-r--r--. 1 ec2-user ec2-user 0 Apr  6 22:28 inputs.tf
-rw-r--r--. 1 ec2-user ec2-user 0 Apr  6 22:28 main.tf
-rw-r--r--. 1 ec2-user ec2-user 0 Apr  6 22:28 outputs.tf
```

```
[ec2-user@ip-172-31-12-203 s3]$
[ec2-user@ip-172-31-12-203 s3]$ cat main.tf
resource "aws_s3_bucket" "terraform_bucket" {
  bucket = "terraform_bucket_test"
```

```
    acl = "private"
}
[ec2-user@ip-172-31-12-203 s3]$
```

```
[ec2-user@ip-172-31-12-203 s3]$ cat main.tf
resource "aws_s3_bucket" "terraform_bucket" {
    bucket = "terraform_bucket_test"
    acl = "private"
}
```

```
[ec2-user@ip-172-31-12-203 s3]$ cat main.tf
resource "aws_s3_bucket" "terraform-bucket" {
    bucket = "terraform_bucket_test"
    acl = "private"
}
```




In root folder

```
[ec2-user@ip-172-31-12-203 05-terraform-modules-project]$ cat main.tf
module "my_ec2" {
    source = "./modules/ec2"
    ami = "ami-02cd5b9bfb2512340"
}
```

```
module "my_s3" {
    source = "./modules/s3"
}
```

```
[ec2-user@ip-172-31-12-203 05-terraform-modules-project]$ cat outputs.tf
output "ec2_vm_public_ip" {
    value = module.my_ec2.public_ip
}
```

```
output "ec2_vm_private_ip" {
    value = module.my_ec2.private_ip
}
```

<input type="checkbox"/>	TerraformEC2	i-0f3b562c215e434b7	Running  	t2.micro	2/2 checks passed
<input type="checkbox"/>	modules-Linux...	i-02a3cf3bf2f09c327	Running  	t2.micro	Initializing

(and 2 more similar warnings elsewhere)

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

Outputs:

ec2_vm_private_ip = "172.31.6.112"

ec2_vm_public_ip = "15.156.92.112"

[ec2-user@ip-172-31-12-203 05-terraform-modules-project]\$

Find buckets by name

Name	AWS Region	IAM Access Analyzer
elasticbeanstalk-ca-central-1-577638386543	Canada (Central) ca-central-1	View analyzer for ca-central-1
elasticbeanstalk-us-east-1-577638386543	US East (N. Virginia) us-east-1	View analyzer for us-east-1
terraform-bucket38498	Canada (Central) ca-central-1	View analyzer for ca-central-1

Now it shows the bucket also

(and one more similar warning elsewhere)

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

Outputs:

ec2_vm_private_ip = "172.31.6.112"

ec2_vm_public_ip = "15.156.92.112"

s3_name_bucket = "terraform-bucket38498"

[ec2-user@ip-172-31-12-203 05-terraform-modules-project]\$

There is no undo. Only yes will be accepted to confirm.

Enter a value: yes

module.my_ec2.aws_instance.linux_vm: Destroying... [id=i-02a3cf3bf2f09c327]

module.my_s3.aws_s3_bucket.terraform-bucket: Destroying... [id=terraform-bucket38498]

module.my_s3.aws_s3_bucket.terraform-bucket: Destruction complete after 0s

module.my_ec2.aws_instance.linux_vm: Still destroying... [id=i-02a3cf3bf2f09c327, 10s elapsed]

module.my_ec2.aws_instance.linux_vm: Still destroying... [id=i-02a3cf3bf2f09c327, 20s elapsed]

module.my_ec2.aws_instance.linux_vm: Destruction complete after 30s

Destroy complete! Resources: 2 destroyed.

[ec2-user@ip-172-31-12-203 05-terraform-modules-project]\$

Broadcast message from root@ip-172-31-12-203.ca-central-1.compute.internal (Mon 2025-04-07 03:22:10 UTC):

