

# 🌟 Creating a Reusable Terraform Module for EC2 Instances

Terraform is a powerful tool for managing infrastructure as code. In this tutorial, we'll create a reusable module to launch EC2 instances. We'll cover the folder structure, necessary files, and how to use the module in different configurations.

## 📁 Folder Structure

First, let's set up the folder structure for our Terraform module:

```
$ mkdir -p modules/ec2-instance  
$ cd modules/ec2-instance
```



# file structure

## Set Up AWS Credentials:

```
export AWS_ACCESS_KEY_ID="your_access_key"
```

```
export AWS_SECRET_ACCESS_KEY="your_secret_key"
```

```
root@ip-172-31-27-15:~/demo#  
root@ip-172-31-27-15:~/demo# tree
```

```
.  
├── main.tf  
├── modules  
│   └── ec2_instance  
│       ├── main.tf  
│       ├── output.tf  
│       ├── terrafor.tfvars  
│       └── variable.tf  
└── snap
```

```
4 directories, 5 files
```

# Module Configuration

The **main.tf** configuration file for the module:

```
main.tf 3, U ●
main.tf > ...
1  provider "aws" {
2      region = "eu-west-3"
3  }
4
5
6
7  resource "aws_instance" "my-instance" {
8      ami           = var.ami_value
9      instance_type = var.instance_type value
10
11     tags = {
12         Name = var.tag_id
13     }
14 }
```

# variables.tf

Define the variables for the module:

```
main.tf U ● variable.tf U X
variable.tf > variable "tag_id"
1
2 variable "ami_value" {
3     description = "Value for the AMI"
4 }
5 variable "instance_type_value" {
6     description = "Value for the instance type"
7 }
8 variable "tag_id" {
9     description = "Tag ID to apply to the instance"
10 }
```

# terraform.tfvars

Specify the values for the variables:

```
Y main.tf U ●  Y variable.tf U  Y terraform.tfvars U ●
Y terraform.tfvars > ...
1  ami_value      = "ami-053b0d53c2564acc10"
2  instance_type_value = "t3.micro"
3
4
```

# outputs.tf

Define the outputs for the module:



The screenshot shows a code editor with four tabs at the top: `main.tf`, `variable.tf`, `terraform.tfvars`, and `outputs.tf`. The `outputs.tf` tab is active and highlighted with a blue line. Below the tabs, the editor shows the content of `outputs.tf` with line numbers 1 through 4 on the left. The code defines an output named `public-ip-address` with a value of `aws_instance.my-instance.public_ip`.

```
1 output "public-ip-address" {  
2     value = aws_instance.my-instance.public_ip  
3 }  
4
```



# Using the Module

Navigate outside of the modules/ec2-instance directory: **cd ../..**

```
EXPLORER  ...  main.tf .\ U  variable.tf U  terraform.tfvars U  outputs.tf U  ec2_instance U  main.tf ... \mod

PRACTICAL
▼ project \ modules
  ec2_instance U
  main.tf U
  main.tf U
  outputs.tf U
  terraform.tfvars U
  variable.tf U

project > modules > main.tf > module "ec2_instance-1"
1  provider "aws" {
2    region = "eu-west-3-"
3  }
4
5  module "ec2_instance-1" {
6    source          = "../modules/ec2_instance"
7    ami_value       = "ami-09d83d8d719da9808"
8    instance_type_value = "t2.micro"
9    tag_id          = "instance-1"
10 }
11
12
13 module "ec2_instance-2" {
14   source          = "../modules/ec2_instance"
15   ami_value       = "ami-064983766e6ab3419"
16   instance_type_value = "t3.micro"
17   tag_id          = "instance-2"
18 }
```

# Run Terraform Commands

**cd project**

**terraform init**

**terraform apply --auto-approve**

The screenshot shows the AWS CLI terminal output for a Terraform apply command. The terminal shows the command being executed and the resulting state of the EC2 instance. A message indicates that the instance has been successfully terminated. Below the terminal output, the AWS Management Console is shown, displaying the EC2 Dashboard with a list of instances. The instance 'instance-1' with ID 'i-0294b3c6d43e67a71' is highlighted, showing its status as 'Terminated'.

```
+ private_dns_name_options (known after apply)
root@ip-172-31-30-37:~/terraform-ec2-instance-module/project#
root@ip-172-31-30-37:~/terraform-ec2-instance-module/project#
root@ip-172-31-30-37:~/terraform-ec2-instance-module/project#
root@ip-172-31-30-37:~/terraform-ec2-instance-module/project# terraform apply --auto-approve
module.ec2_instance-1.aws_instance.my-instance: Refreshing state... [id=i-0294b3c6d43e67a71]

Note: Objects have changed outside of Terraform

Terraform detected the following changes made outside of Terraform since the last "terraform apply" which may have affected this plan:

# module.ec2_instance-1.aws_instance.my-instance has been deleted
```

Successfully initiated termination of i-0294b3c6d43e67a71

Instances (1/3) Info

Find Instance by attribute or tag (case-sensitive) All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Publ
<input type="checkbox"/>	instance-2	i-0f23c1a7b2a1cc6dc	Running	t3.micro	Initializing	View alarms +	eu-west-3a	ec2-
<input checked="" type="checkbox"/>	instance-1	i-0294b3c6d43e67a71	Terminated	t2.micro	-	View alarms +	eu-west-3c	-
<input type="checkbox"/>	instance-1	i-0c3ad47544356b4d3	Running	t2.micro	Initializing	View alarms +	eu-west-3c	ec2-





# Thank you!

<https://github.com/vaibhav0342/terraform-ec2-instance-module>