

Terraform Modules Tutorial

Step 1: Project Structure

Terraform modules are just directories with `.tf` files that can be reused. We'll create a structure like this:

```
terraform-modules-example/  
  main.tf  
  variables.tf  
  outputs.tf  
  modules/  
    ec2-instance/  
      main.tf  
      variables.tf  
      outputs.tf
```

Step 2: Write the EC2 Module

```
modules/ec2-instance/main.tf  
-----  
resource "aws_instance" "this" {  
  ami           = var.ami  
  instance_type = var.instance_type  
  
  tags = {  
    Name = var.instance_name  
  }  
}  
  
resource "aws_security_group" "this" {  
  name           = "${var.instance_name}-sg"  
  description    = "Allow SSH inbound traffic"  
  
  ingress {  
    from_port = 22  
    to_port   = 22  
    protocol  = "tcp"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
  
  egress {  
    from_port = 0  
    to_port   = 0  
    protocol  = "-1"  
    cidr_blocks = ["0.0.0.0/0"]  
  }  
}  
  
modules/ec2-instance/variables.tf  
-----  
variable "ami" {  
  description = "AMI ID for the EC2 instance"  
  type        = string  
}  
  
variable "instance_type" {  
  description = "EC2 instance type"  
  type        = string  
  default     = "t2.micro"  
}  
  
variable "instance_name" {  
  description = "Tag Name for the EC2 instance"  
  type        = string  
}  
  
modules/ec2-instance/outputs.tf  
-----  
output "instance_id" {
```

```

    value = aws_instance.this.id
  }

  output "public_ip" {
    value = aws_instance.this.public_ip
  }

```

Step 3: Call the Module from Root

```

main.tf
-----
provider "aws" {
  region = "us-east-1"
}

module "my_ec2" {
  source          = "../modules/ec2-instance"
  ami             = "ami-0c55b159cbfafa1f0"
  instance_type   = "t2.micro"
  instance_name   = "MyTerraformEC2"
}

outputs.tf
-----
output "ec2_id" {
  value = module.my_ec2.instance_id
}

output "ec2_public_ip" {
  value = module.my_ec2.public_ip
}

```

Step 4: Run Terraform Commands

```

terraform init
terraform plan
terraform apply -auto-approve

```

Step 5: Benefits of Modules

- Reusable: Can use the same ec2-instance module for multiple instances.
- Maintainable: Logic stays separate from variables.
- Scalable: You can add VPC, RDS, or S3 as more modules.

Example: Reusing the Module

```

module "app_server" {
  source          = "../modules/ec2-instance"
  ami             = "ami-0c55b159cbfafa1f0"
  instance_type   = "t2.micro"
  instance_name   = "AppServer"
}

module "db_server" {
  source          = "../modules/ec2-instance"
  ami             = "ami-0c55b159cbfafa1f0"
  instance_type   = "t2.micro"
  instance_name   = "DBServer"
}

```