**Modules**

**Assignment:**

Modules & workspace assignment

Create a Terraform module that provisions a \*VPC, security groups (allowing SSH & HTTP)\* , and an \*EC2 instance\* . Upload the module code to your github repo and reference the github source for your module block

The setup should use Terraform workspaces for different environments (dev, stage, prod) and allow users to specify the CIDR block of their choice.

Make use of -

Variables for user-defined inputs.

Locals for computed values.

Functions to format values dynamically.

link: https://spacelift.io/blog/what-are-terraform-modules-and-how-do-they-work

Steps:

* Let’s create module to create VPC, Security group and the EC2
* Modules will have 3 files, main.tf, variables.tf, and output.tf
* Let’s give it a try locally and then will upload this to the github
* Module main.tf

|  |
| --- |
| resource "aws\_vpc" "test\_vpc" {  cidr\_block = var.vpc\_block  tags = {  Name = var.vpc\_name  }  }  resource "aws\_subnet" "test\_subnet" {  vpc\_id = aws\_vpc.test\_vpc.id  cidr\_block = var.subnet\_block  availability\_zone = var.availability\_zone  tags = {  Name = var.subnet\_name  }  }  resource "aws\_security\_group" "test\_sg" {  name = var.security\_groups\_group\_name  description = var.security\_groups\_desc  vpc\_id = aws\_vpc.test\_vpc.id  ingress {  from\_port = var.ssh\_port  to\_port = var.ssh\_port  protocol = var.protocol  cidr\_blocks = var.traffic\_from\_anywhere  }  ingress { #allowing http port  from\_port = var.http\_port  to\_port = var.http\_port  protocol = var.protocol  cidr\_blocks = var.traffic\_from\_anywhere #from anywhere  }  }  resource "aws\_instance" "workspace\_instance" {  ami = var.ami\_value  instance\_type = var.ec2\_type  subnet\_id = aws\_subnet.test\_subnet.id  security\_groups = [aws\_security\_group.test\_sg.id]  availability\_zone = aws\_subnet.test\_subnet.availability\_zone  } |

module Variable.tf file

|  |
| --- |
| variable "vpc\_block" {  description = "CIDR range"  }  variable "vpc\_name" {  description = "Name of the VPC"  }  variable "subnet\_block" {  description = "CIDR block of the subnet"  }  variable "subnet\_name" {  description = "Name of the VPC"  }  variable "availability\_zone" {  description = "The avaiblity zone"  }  variable "security\_groups\_group\_name" {  description = "security group name"  }  variable "security\_groups\_desc" {  default = "Allow http and ssh ports"  description = "security group name description"  }  variable "ssh\_port" {  default = "22"  description = "SSH port"  }  variable "http\_port" {  default = "80"  description = "HTTP port"  }  variable "protocol" {  default = "tcp"  description = "tcp protocol"  }  variable "traffic\_from\_anywhere" {  default = ["0.0.0.0/0"]  description = "Allow traffic from anywhere"  }  variable "ami\_value" {  description = "The AMI value"  }  variable "ec2\_type" {} |

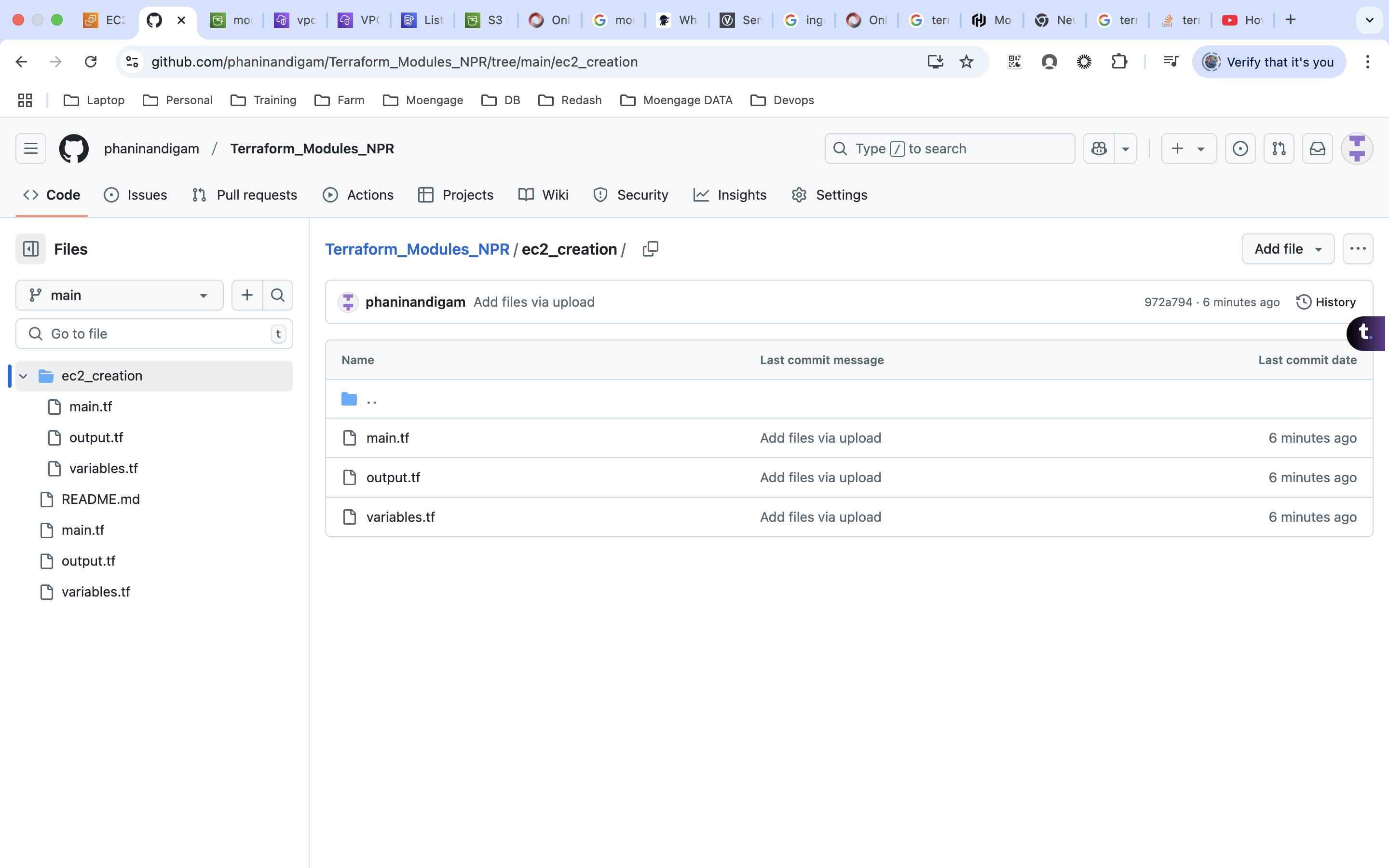
* Project main file main.tf

|  |
| --- |
| provider "aws" {  region = "ap-south-1"  }  module "ec2\_creation" {  source = "./Modules/ec2\_creation"  vpc\_block = var.cidr\_block  vpc\_name = var.vpc\_name  subnet\_block = var.subnet\_range  subnet\_name = var.subnet\_name  availability\_zone = var.availability\_zone  security\_groups\_group\_name = var.security\_groups\_name  security\_groups\_desc = var.security\_groups\_desc  ami\_value = "ami-00bb6a80f01f03502"  ec2\_type = var.type  } |

* Project variable file variables.tf

|  |
| --- |
| variable "cidr\_block" {  description = "Enter the CIDR range for the VPC"  }  variable "vpc\_name" {  default = "test\_vpc"  }  variable "subnet\_range" {  description = "Enter the subnet block range"  }  variable "subnet\_name" {  default = "test\_subnet"  }  variable "availability\_zone" {  default = "ap-south-1a"  }  variable "security\_groups\_name" {  default = "Test\_SG"  description = "The name of the security group"  }  variable "security\_groups\_desc" {  default = "Allowing ssh and http ports"  }  variable "type" {  description = "Enter the instance type"  } |

* when we run the apply command, it will ask us to enter the cidr range for VPC and subnet and the instance type. (We have not provided the default values for these)  
  A screenshot of a computer

  AI-generated content may be incorrect.
* Moved the module code to the github and updated the module block  
  https://github.com/phaninandigam/Terraform\_Modules\_NPR.git  
  

|  |
| --- |
| module "ec2\_creation" {  source = "github.com/phaninandigam/Terraform\_Modules\_NPR/ec2\_creation"  # source = "./Modules/ec2\_creation"  vpc\_block = var.cidr\_block  vpc\_name = format("%s\_%s","${terraform.workspace}",var.vpc\_name)  subnet\_block = var.subnet\_range  subnet\_name = format("%s\_%s","${terraform.workspace}",var.subnet\_name)  availability\_zone = var.availability\_zone  security\_groups\_group\_name = var.security\_groups\_name  security\_groups\_desc = var.security\_groups\_desc  ami\_value = "ami-00bb6a80f01f03502"  ec2\_type = var.type  ec2\_name = format("%s\_%s","${terraform.workspace}","ec2-Instance")  } |

* Will add workspaces now, I will be using s3 as a backend  
  terraform workspace new dev  
  terraform workspace new prod  
  terraform workspace new stage

|  |
| --- |
| terraform {  backend "s3" {  bucket = "modules-workspace-tfstate-files" #bucketname  key = "statefiles" #path  region = "ap-south-1"  # dynamodb\_table = "terraform\_state\_lock\_file"  use\_lockfile = true  }  } |

* Now, based on the workspace, we will be adding the vpc, subnet and in the instance names  
  A screenshot of a computer

  AI-generated content may be incorrect.

|  |
| --- |
| module "ec2\_creation" {  # source = "github.com/phaninandigam/Terrafrom\_ec2\_creation\_modules"  source = "./Modules/ec2\_creation"  vpc\_block = var.cidr\_block  vpc\_name = format("%s\_%s","${terraform.workspace}",var.vpc\_name)  subnet\_block = var.subnet\_range  subnet\_name = format("%s\_%s","${terraform.workspace}",var.subnet\_name)  availability\_zone = var.availability\_zone  security\_groups\_group\_name = var.security\_groups\_name  security\_groups\_desc = var.security\_groups\_desc  ami\_value = "ami-00bb6a80f01f03502"  ec2\_type = var.type  ec2\_name = format("%s\_%s","${terraform.workspace}","ec2-Instance")  } |

A screenshot of a computer

AI-generated content may be incorrect.

Complete Code:

**main.tf**

|  |
| --- |
| provider "aws" {  region = "ap-south-1"  }  module "ec2\_creation" {  source = "github.com/phaninandigam/Terraform\_Modules\_NPR/ec2\_creation"  # source = "./Modules/ec2\_creation"  vpc\_block = var.cidr\_block  vpc\_name = format("%s\_%s","${terraform.workspace}",var.vpc\_name)  subnet\_block = var.subnet\_range  subnet\_name = format("%s\_%s","${terraform.workspace}",var.subnet\_name)  availability\_zone = var.availability\_zone  security\_groups\_group\_name = var.security\_groups\_name  security\_groups\_desc = var.security\_groups\_desc  ami\_value = "ami-00bb6a80f01f03502"  ec2\_type = var.type  ec2\_name = format("%s\_%s","${terraform.workspace}","ec2-Instance")  }  terraform {  backend "s3" {  bucket = "modules-workspace-tfstate-files" #bucketname  key = "statefiles" #path  region = "ap-south-1"  # dynamodb\_table = "terraform\_state\_lock\_file"  use\_lockfile = true  }  }  locals {  instance = {  "default" = "t2.micro"  "dev" = "t2.micro"  "staging" = "t3.small"  "prod" = "t3.medium"  }  } |

**variables.tf**

|  |
| --- |
| variable "cidr\_block" {  description = "Enter the CIDR range for the VPC"  }  variable "vpc\_name" {  default = "test\_vpc"  }  variable "subnet\_range" {  description = "Enter the subnet block range"  }  variable "subnet\_name" {  default = "test\_subnet"  }  variable "availability\_zone" {  default = "ap-south-1a"  }  variable "security\_groups\_name" {  default = "Test\_SG"  description = "The name of the security group"  }  variable "security\_groups\_desc" {  default = "Allowing ssh and http ports"  }  variable "type" {  description = "Enter the instance type"  } |

**modules code:**

<https://github.com/phaninandigam/Terrafrom_ec2_creation_modules.git>