Install Terraform

- run below script to install Terraform

#!/bin/bash

sudo apt update && sudo apt install -y wget gnupg software-properties-common

wget -O- https://apt.releases.hashicorp.com/gpg | \

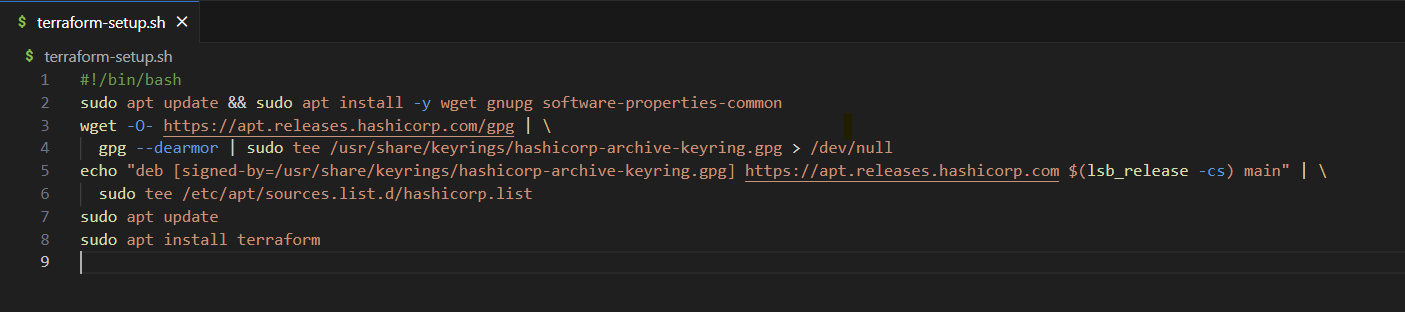
gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null

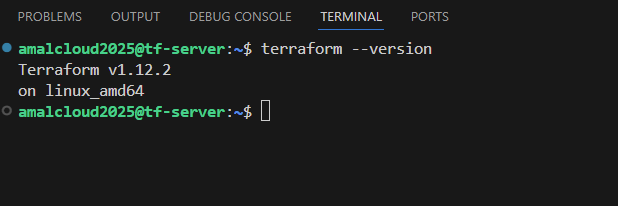
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb\_release -cs) main" | \

sudo tee /etc/apt/sources.list.d/hashicorp.list

sudo apt update

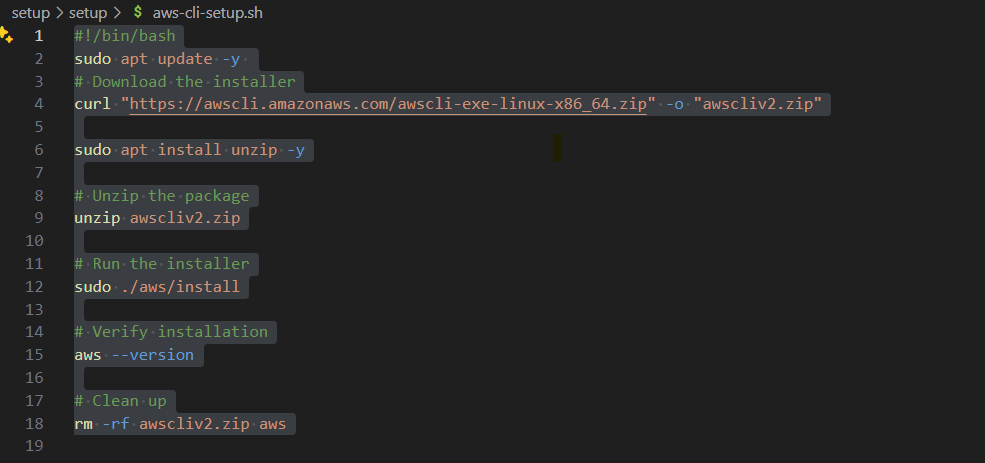
sudo apt install terraform

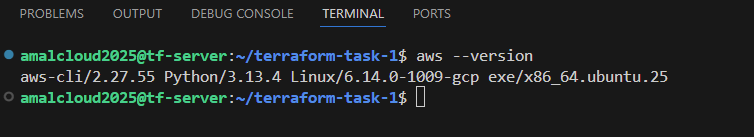




Install aws cli

Run the below script to install aws cli and configure it.





Terraform

**Main.tf**

provider "aws" { region = var.region }

provider "aws" { alias = "us" region = var.region\_us }

resource "aws\_key\_pair" "web-server\_key" { key\_name = var.key\_name public\_key = file(var.public\_key\_path) }

resource "aws\_key\_pair" "web-server\_us\_key" { provider = aws.us key\_name = var.key\_name public\_key = file(var.public\_key\_path) }

resource "aws\_security\_group" "web\_sg" { name = "allow\_http\_ssh" description = "Allow SSH (22) and HTTP (80)"

ingress { from\_port = 22 to\_port = 22 protocol = "tcp" cidr\_blocks = ["0.0.0.0/0"] }

ingress { from\_port = 80 to\_port = 80 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"] } }

resource "aws\_security\_group" "web\_sg\_us" { provider = aws.us name = "allow\_http\_ssh" description = "Allow SSH (22) and HTTP (80)"

ingress { from\_port = 22 to\_port = 22 protocol = "tcp" cidr\_blocks = ["0.0.0.0/0"] }

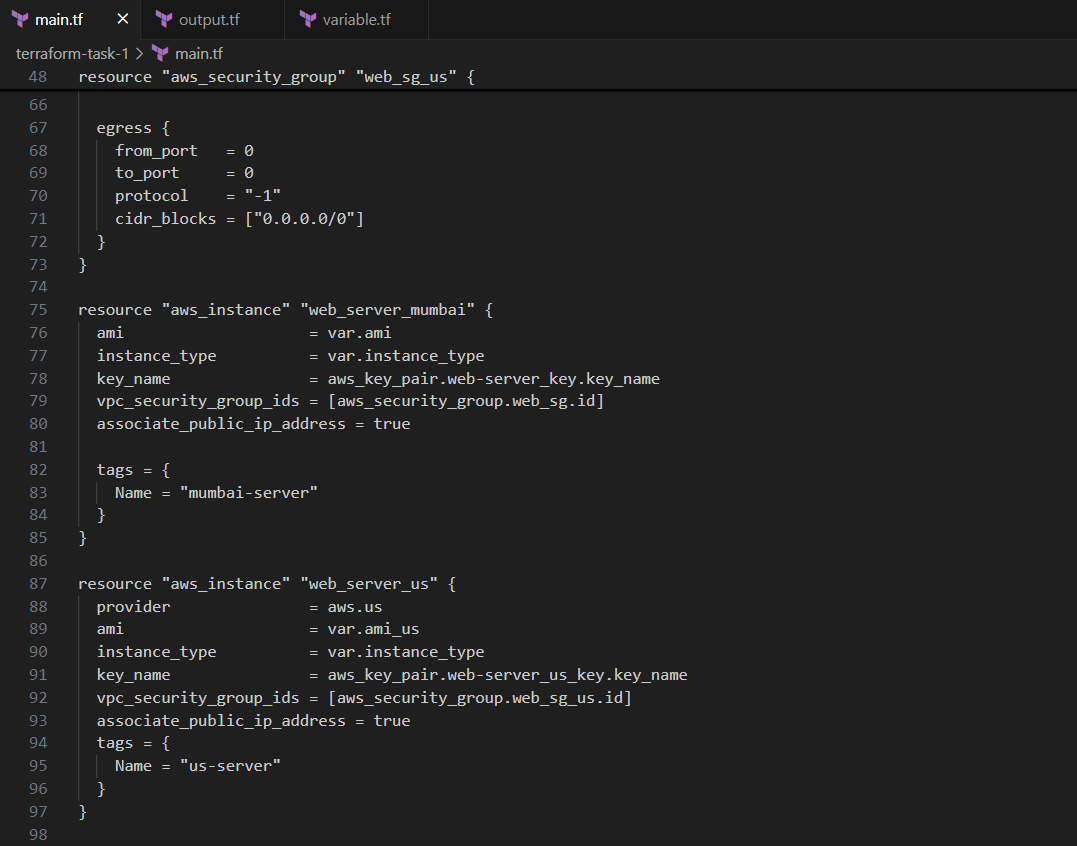
ingress { from\_port = 80 to\_port = 80 protocol = "tcp" cidr\_blocks = ["0.0.0.0/0"] }

egress { from\_port = 0 to\_port = 0 protocol = "-1" cidr\_blocks = ["0.0.0.0/0"] } }

resource "aws\_instance" "web\_server\_mumbai" { ami = var.ami instance\_type = var.instance\_type key\_name = aws\_key\_pair.web-server\_key.key\_name vpc\_security\_group\_ids = [aws\_security\_group.web\_sg.id] associate\_public\_ip\_address = true

tags = { Name = "mumbai-server" } }

resource "aws\_instance" "web\_server\_us" { provider = aws.us ami = var.ami\_us instance\_type = var.instance\_type key\_name = aws\_key\_pair.web-server\_us\_key.key\_name vpc\_security\_group\_ids = [aws\_security\_group.web\_sg\_us.id] associate\_public\_ip\_address = true tags = { Name = "us-server" } }



**Variable.tf**

variable "region" { type = string default = "ap-south-1" }

variable "region\_us" { type = string default = "us-east-2" }

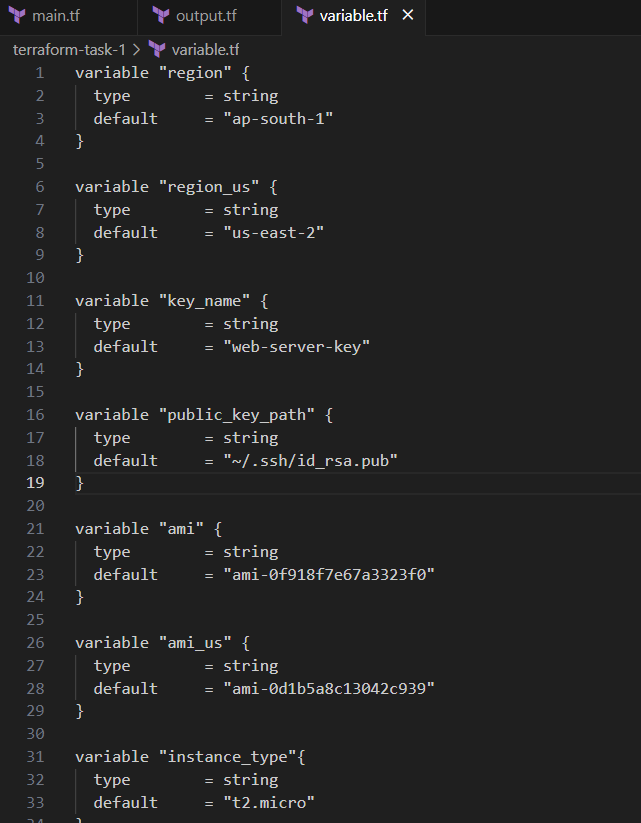
variable "key\_name" { type = string default = "web-server-key" }

variable "public\_key\_path" { type = string default = "~/.ssh/id\_rsa.pub" }

variable "ami" { type = string default = "ami-0f918f7e67a3323f0" }

variable "ami\_us" { type = string default = "ami-0d1b5a8c13042c939" }

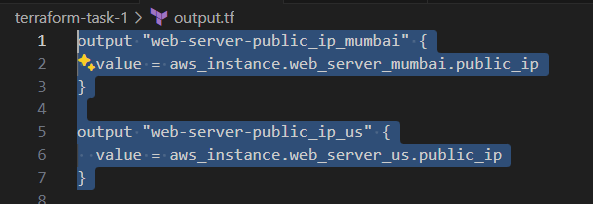
variable "instance\_type"{ type = string default = "t2.micro" }



**Output.tf**

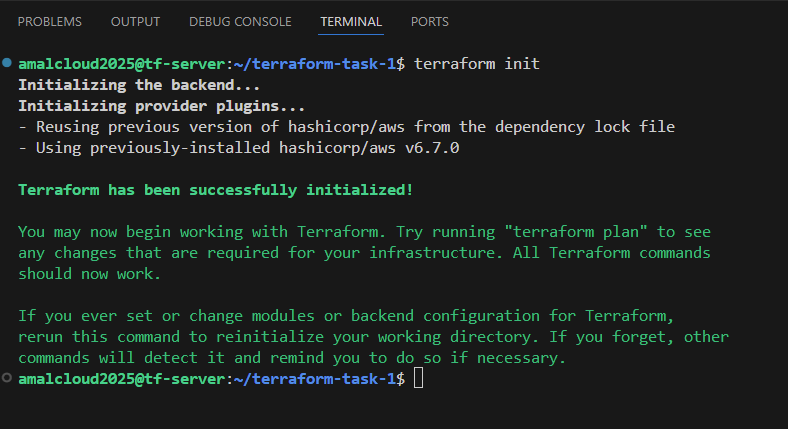
output "web-server-public\_ip\_mumbai" { value = aws\_instance.web\_server\_mumbai.public\_ip }

output "web-server-public\_ip\_us" { value = aws\_instance.web\_server\_us.public\_ip }

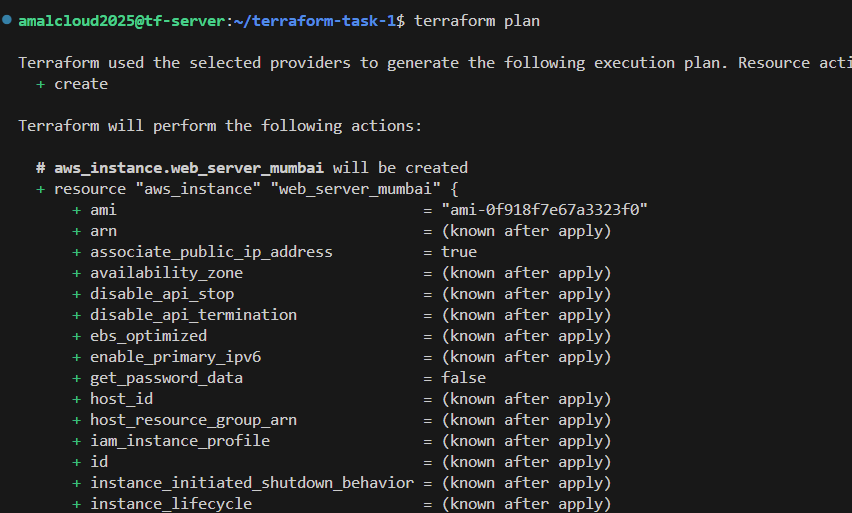


**Deploying**

terraform init



terraform plan



terraform apply

