1. **Terraform Installed:**
2. **AWS Credentials Configured:** You should have your AWS credentials set up using the AWS CLI (aws configure) or environment variables.

**Steps:**

1. **Create a new directory for the Terraform project:**

**mkdir terraform**

**cd terraform**

1. **Create the Terraform Configuration File:** Create a file called main.tf inside the directory with the following content:

provider "aws" {

  region = "us-east-2"

}

resource "aws\_instance" "terraform" {

  ami           = "ami-085f9c64a9b75eed5" # Ubuntu 22.04 AMI for Ohio region

  instance\_type = "t2.micro"

  tags = {

    Name = "terraform-ec2"

  }

}

1. **Initialize Terraform:** Initialize your Terraform workspace by running the following command:

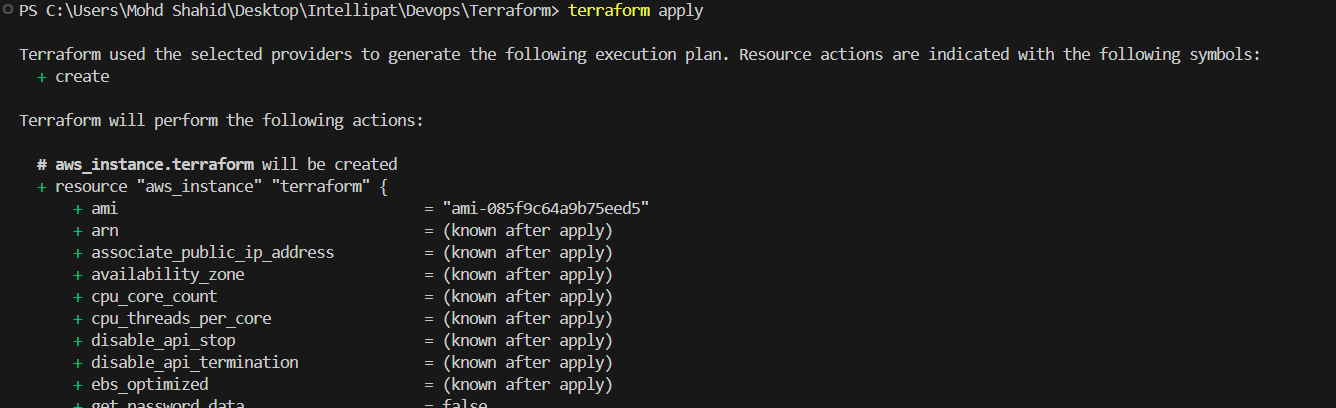
**terraform init**

1. **Preview the Changes:** After initialization, you can preview the changes Terraform will make by running:

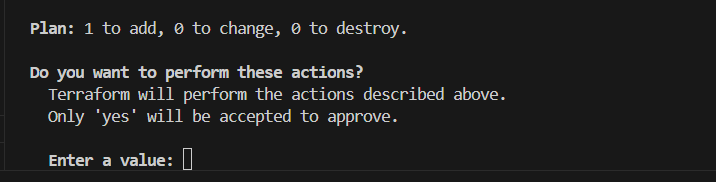
**terraform plan**

1. **Apply the Changes:** Apply the configuration to create the EC2 instance in the default VPC's subnet:

**terraform apply**



When prompted, type yes to confirm the action.



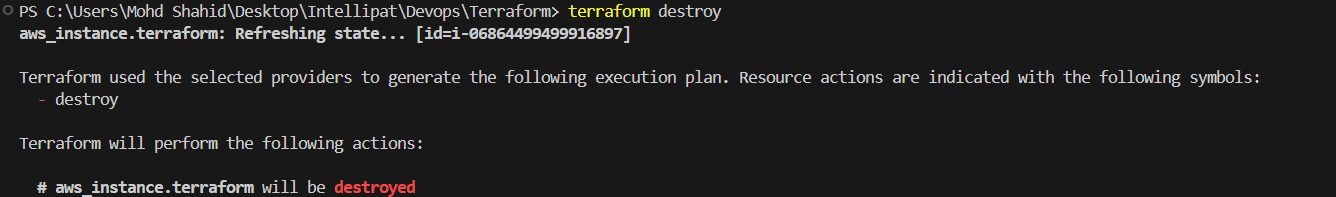


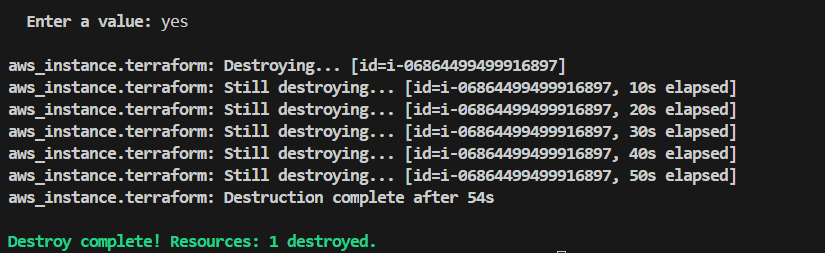
**Explanation:**

* **Provider Block (provider "aws"):** Specifies the AWS region to use. Ohio is us-east-2.
* **Resource Block (aws\_instance "example"):** Defines an EC2 instance resource. The ami is an Ubuntu 22.04 AMI specific to the Ohio region, and the instance\_type is t2.micro, which is eligible for the free tier.

1. **Run the Terraform destroy command:** This command will destroy all resources that were created by your previous Terraform deployment:

**terraform destroy**





1. **Update the main.tf to include Elastic IP:**

Update the previous main.tf file to attach an Elastic IP to the new EC2 instance. Below is the updated content:

provider "aws" {

  region = "us-east-2"

}

resource "aws\_instance" "terraform" {

  ami           = "ami-085f9c64a9b75eed5" # Ubuntu 22.04 AMI for Ohio region

  instance\_type = "t2.micro"

  tags = {

    Name = "terraform-ec2-with-eip"

  }

}

resource "aws\_eip" "terraform\_eip" {

  instance = aws\_instance.terraform.id

}

* + **aws\_instance "terraform":** Defines the EC2 instance configuration.
  + **aws\_eip "terraform\_eip":** Allocates an Elastic IP and associates it with the EC2 instance.

1. **Re-initialize Terraform:** After updating the configuration, re-initialize Terraform (if required):

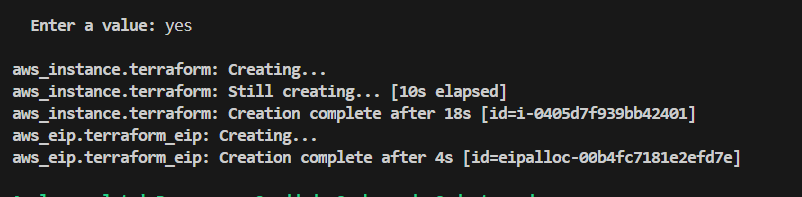
**terraform init**

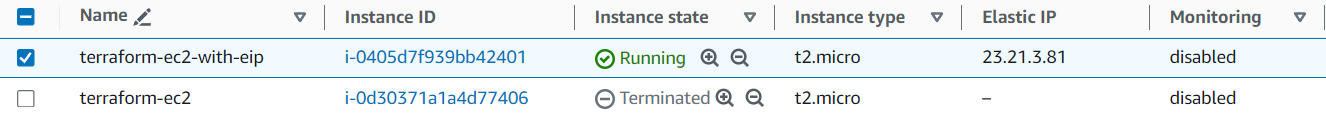
1. **Preview the Changes:** To verify the configuration and see the planned changes, run:

**terraform plan**

1. **Apply the Changes:** To create the EC2 instance with an Elastic IP, run the following command:

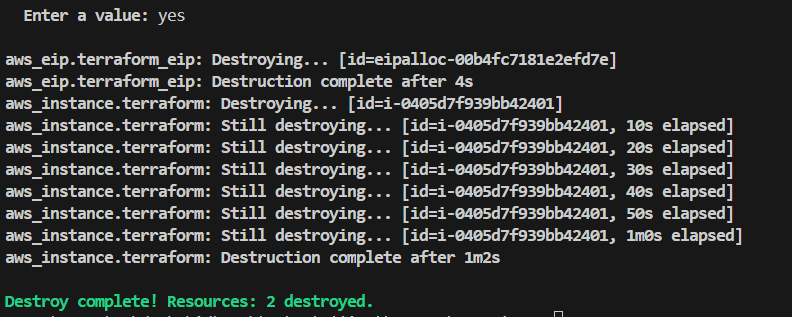
**terraform apply**

****



1. **Run the Terraform destroy command:** This command will destroy all resources that were created by your previous Terraform deployment:

**terraform destroy**

****

1. **Update the main.tf to include both EC2 instances with custom names:**

provider "aws" {

  alias  = "virginia"

  region = "us-east-1"

}

provider "aws" {

alias = "ohio"

  region = "us-east-2"

}

resource "aws\_instance" " virginia" {

provider = aws.virginia

  ami           = "ami- 0e86e20dae9224db8"

  instance\_type = "t2.micro"

  tags = {

    Name = " hello-virginia"

  }

}

resource "aws\_instance" "ohio" {

  ami           = "ami-085f9c64a9b75eed5"

  instance\_type = "t2.micro"

  tags = {

    Name = " hello-ohio"

  }

}

* + **Two AWS providers** are defined, one for each region (Ohio and N. Virginia), using the alias field to distinguish between them.
  + **Two EC2 instances** are created in separate regions, with the tags section assigning custom names to each instance:
    - Ohio instance: hello-ohio
    - N. Virginia instance: hello-virginia

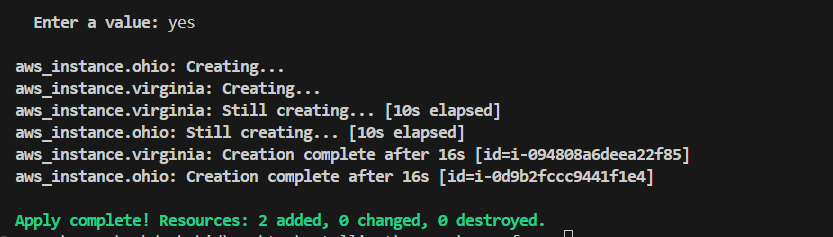
1. **Preview the Changes:** To verify the configuration and see the planned changes, run:

**terraform plan**

1. **Apply the Changes:** Apply the configuration to create the two EC2 instances in their respective regions:

**terraform apply**

When prompted, type yes to confirm.



**Result:**

* You will now have:
  + One EC2 instance in the **Ohio** region (us-east-2) named hello-ohio.



* + One EC2 instance in the **N. Virginia** region (us-east-1) named hello-virginia.

