Terraform Assignment - 5

You have been asked to:

- Destroy the previous deployments
- Create a script to install apache2
- Run this script on a newly created EC2 instance
- Print the IP address of the instance in a file on the local, once deployed

Solution:

1) Run Terraform Destroy:

```
Destroy complete! Resources: 6 destroyed.
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

2) Verify : terraform state list

The output should be empty if all resources have been destroyed.

```
ubuntu@ip-172-31-7-35:~/terraform-project$ terraform state list
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

3) Create a Script to Install Apache2:

Create a file named install-apache2.sh with the following content:

```
  GNU nano 7.2
sudo apt update -y
sudo apt install apache2 -y
sudo systemctl start apache2
sudo systemctl enable apache2
echo "Apache2 is successfully installed and running."
```

4) Make the script executable: sudo chmod +x install-apache2.sh

```
ubuntu@ip-172-31-7-35:~/terraform-project$ sudo chmod +x install-apache2.sh
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

5) Run the Script on a Newly Created EC2 Instance: update the Terraform configuration to include the EC2 instance and user data to run the script

```
provider "aws" {
  region = "us-east-1"
}

# Create EC2 Instance
resource "aws_instance" "apache_instance" {
  ami           = "ami-04b4f1a9cf54c11d0"
  instance_type = "t2.micro"

  # Pass the Apache2 install script as user data
  user_data = <<-EOF
    #!/bin/bash
    sudo apt update -y
    sudo apt install apache2 -y
    sudo systemctl start apache2
    sudo systemctl enable apache2
    echo "Apache2 is successfully installed and running."
  EOF

  tags = {
    Name = "Apache-EC2-Instance"
  }
}
```

6) Use Terraform's outputs to fetch the public IP of the instance:

```
output "instance_public_ip" {
  value = aws_instance.apache_instance.public_ip
}
```

7) Initialize Terraform:

```
ubuntu@ip-172-31-7-35:~/terraform-project$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.91.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

8) Plan:

```
Plan: 1 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + instance_public_ip = (known after apply)


Note: You didn't use the -out option to save this plan, so Terrafo
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

9) Apply:

```
aws_instance.apache_instance: Creating...
aws_instance.apache_instance: Still creating... [10s elapsed]
aws_instance.apache_instance: Creation complete after 15s [id=i-0cc8c578c8fc942db]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.

Outputs:

instance_public_ip = "35.175.196.188"
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

10) Once the deployment is complete, use the following command to write the IP to a file on your local machine:
terraform output instance_public_ip > instance_ip.txt
This will create a file instance_ip.txt containing the public IP address of the EC2 instance.

```
ubuntu@ip-172-31-7-35:~/terraform-project$ terraform output instance_public_ip > instance_ip.txt
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

11) Verify: Verify the contents of the instance_ip.txt file
cat instance_ip.txt

```
ubuntu@ip-172-31-7-35:~/terraform-project$ cat instance_ip.txt
"35.175.196.188"
ubuntu@ip-172-31-7-35:~/terraform-project$
```

i-0c2ef8ee473810c4e (my-terraform)

PublicIPs: 3.110.54.55 PrivateIPs: 172.31.7.35

12) Open a browser and navigate to the public IP (e.g., http://<public_ip>) to confirm Apache2 is running.