

## **Module 8: Terraform Assignment - 2**

**Assignment Submitted By:-Hitesh Chauhan**

**Course Offered: -Advanced Cloud Computing and Devops**

**Assignment By: -Intellipaat**

**Trainer: -Kumar**

**Date Of Submission: -03/03/2025**

**Tasks To Be Performed:**

1. Destroy the previous deployment
2. Create a new EC2 instance with an Elastic IP

# **SOLUTION**

## 1. Destroy the previous deployment.

Now in this task we need to run another command for destroy the deployment. the command is **terraform destroy**

```
root@DESKTOP-V0HL7L3:/home/hitesh/terraform/assignment01# terraform apply

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# aws_instance.assignment-4 will be created
+ resource "aws_instance" "assignment-4" {
  + ami                    = "ami-0cb91c7de36eed2cb"
  + arn                    = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone      = (known after apply)
  + cpu_core_count         = (known after apply)
  + cpu_threads_per_core   = (known after apply)
  + disable_api_stop       = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized          = (known after apply)
  + enable_primary_ipv6    = (known after apply)
  + get_password_data      = false
  + host_id                = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile   = (known after apply)
  + id                     = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
  + instance_lifecycle     = (known after apply)
  + instance_state         = (known after apply)
  + instance_type          = "t2.micro"
  + ipv6_address_count     = (known after apply)
  + ipv6_addresses        = (known after apply)
  + key_name               = "docker"
  + monitoring             = (known after apply)
  + outpost_arn            = (known after apply)
  + password_data          = (known after apply)
  + placement_group        = (known after apply)
  + placement_partition_number = (known after apply)
  + primary_network_interface_id = (known after apply)
  + private_dns            = (known after apply)
  + private_ip             = (known after apply)

```

```
  - instance_metadata_tags = "disabled" -> null
}
- private_dns_name_options {
  - enable_resource_name_dns_a_record = false -> null
  - enable_resource_name_dns_aaaa_record = false -> null
  - hostname_type = "ip-name" -> null
}
- root_block_device {
  - delete_on_termination = true -> null
  - device_name = "/dev/sda1" -> null
  - encrypted = false -> null
  - iops = 3000 -> null
  - tags = {} -> null
  - tags_all = {} -> null
  - throughput = 125 -> null
  - volume_id = "vol-09547bf3f482e681c" -> null
  - volume_size = 8 -> null
  - volume_type = "gp3" -> null
  # (1 unchanged attribute hidden)
}
}
```

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

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.  
There is no undo. Only 'yes' will be **accepted** to confirm.

Enter a value: **yes**

```
aws_instance.assignment-4: Destroying... [id=i-006635019b2022a07]
aws_instance.assignment-4: Still destroying... [id=i-006635019b2022a07, 10s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-006635019b2022a07, 20s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-006635019b2022a07, 30s elapsed]
aws_instance.assignment-4: Still destroying... [id=i-006635019b2022a07, 40s elapsed]
aws_instance.assignment-4: Destruction complete after 44s
```

**Destroy complete! Resources: 1 destroyed.**

```
root@DESKTOP-V0HL7L3:/home/hitesh/terraform/assignment01#
```

Now The Deployment has been deleted.

So We need to create another task Create a new EC2 instance with an Elastic IP.

```
assignment02.tf > ...
1  provider "aws" {
2      region = "us-east-2"
3      access_key = ""
4      secret_key = ""
5  }
6  resource "aws_instance" "assignment-2" {
7      ami = ""
8      instance_type = "t2.micro"
9      key_name = ""
10     tags = {
11         Name = "assignment-2"
12     }
13 }
14 resource "aws_eip" "eip" {
15     vpc = true
16 }
17 resource "aws_eip_association" "eip_assoc" {
18     instance_id = aws_instance.assignment-2.id
19     allocation_id = aws_eip.eip.id
20 }
21
```

Now We need to create a another directory name is assignment under terraform folder.

After Create the file name is assignment02.tf.in this filr we need to paste the code.

```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform# mkdir assignment02
root@DESKTOP-VOHL7L3:/home/hitesh/terraform# cd assignment
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment# cd ..
root@DESKTOP-VOHL7L3:/home/hitesh/terraform# cd assignment02/
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment02# ll
total 8
drwxr-xr-x 2 root root 4096 Mar  2 23:44 ./
drwxr-xr-x 8 root root 4096 Mar  2 23:44 ../
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment02# touch assignment02.tf
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment02# sudo nano assignment02.tf
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment02#
```

```
provider "aws" {
    region = "us-east-2"
    access_key = ""
    secret_key = ""
}

resource "aws_instance" "assignment-2" {
    ami = ""
    instance_type = "t2.micro"
    key_name = ""
    tags = {
        Name = "assignment-2"
    }
}

resource "aws_eip" "eip" {
    vpc = true
}

resource "aws_eip_association" "eip_assoc" {
    instance_id = aws_instance.assignment-2.id
    allocation_id = aws_eip.eip.id
}
```

Now we need to run the code in terraform.

NOTE:EVERY TIME YOU NEED TO CHANGE THE DIRECTORY NEED TO RUN THE terraform init COMMAND IN SAME PLACE.

```
root@DESKTOP-V0HL7L3:/home/hitesh/terraform/assignment02# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.89.0...
```

```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment02# terraform init
Initializing the backend...
Initializing provider plugins...
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v5.89.0...
- Installed hashicorp/aws v5.89.0 (signed by HashiCorp)
Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.
```

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.

```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment02#
```

Now Need To Run terraform apply command.

**terraform apply -auto-approve**

**Now This time we need to run this command**

```
root@DESKTOP-VOHL7L3:/home/hitesh/terraform/assignment02# terraform apply -auto-approve
aws_eip.eip: Refreshing state... [id=eipalloc-0d4785987e2430b07]
```

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:  
+ create

Terraform will perform the following actions:

```
# aws_eip_association.eip_assoc will be created
+ resource "aws_eip_association" "eip_assoc" {
  + allocation_id = "eipalloc-0d4785987e2430b07"
  + id            = (known after apply)
  + instance_id  = (known after apply)
  + network_interface_id = (known after apply)
  + private_ip_address = (known after apply)
  + public_ip     = (known after apply)
}

# aws_instance.assignment-2 will be created
+ resource "aws_instance" "assignment-2" {
  + ami                    = "ami-04b4f1a9cf54c11d0"
  + arn                   = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone     = (known after apply)
  + cpu_core_count       = (known after apply)
  + cpu_threads_per_core = (known after apply)
  + disable_api_stop     = (known after apply)
  + disable_api_termination = (known after apply)
  + ebs_optimized        = (known after apply)
  + enable_primary_ipv6  = (known after apply)
  + get_password_data    = false
  + host_id              = (known after apply)
  + host_resource_group_arn = (known after apply)
  + iam_instance_profile  = (known after apply)
  + id                   = (known after apply)
  + instance_initiated_shutdown_behavior = (known after apply)
```

```

+ ebs_block_device (known after apply)
+ enclave_options (known after apply)
+ ephemeral_block_device (known after apply)
+ instance_market_options (known after apply)
+ maintenance_options (known after apply)
+ metadata_options (known after apply)
+ network_interface (known after apply)
+ private_dns_name_options (known after apply)
+ root_block_device (known after apply)
}

```

Plan: 3 to add, 0 to change, 0 to destroy.

aws\_eip.eip: Creating...

aws\_instance.assignment-2: Creating...

aws\_eip.eip: Creation complete after 3s [id=eipalloc-0c43a91aa2afe20c1]

aws\_instance.assignment-2: Still creating... [10s elapsed]

aws\_instance.assignment-2: Creation complete after 17s [id=i-03f89ecb6bd126a3c]

aws\_eip\_association.eip\_assoc: Creating...

aws\_eip\_association.eip\_assoc: Creation complete after 2s [id=eipassoc-0e689dfe32bc72015]

**Warning:** Argument is deprecated

with aws\_eip.eip,  
on assignment02.tf line 15, in resource "aws\_eip" "eip":  
15: vpc = true

use domain attribute instead

(and 2 more similar warnings elsewhere)

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.

root@DESKTOP-V0HL7L3:/home/hitesh/terraform/assignment02#

Deployment Has been completed with elastic ip.

The screenshot shows the AWS Management Console 'Instances' page. A table lists the instance 'assignment-2' with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP4 DNS
assignment-2	i-03f89ecb6bd126a3c	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-44-221-169-248.co...

The screenshot shows the AWS Management Console 'Elastic IP addresses' page. A table lists the Elastic IP address with the following details:

Name	Allocated IPv4 address	Type	Allocation ID	Reverse DNS record	Associated instance ID
-	44.221.169.248	Public IP	eipalloc-0c43a91aa2afe20c1	-	i-03f89ecb6bd126a3c