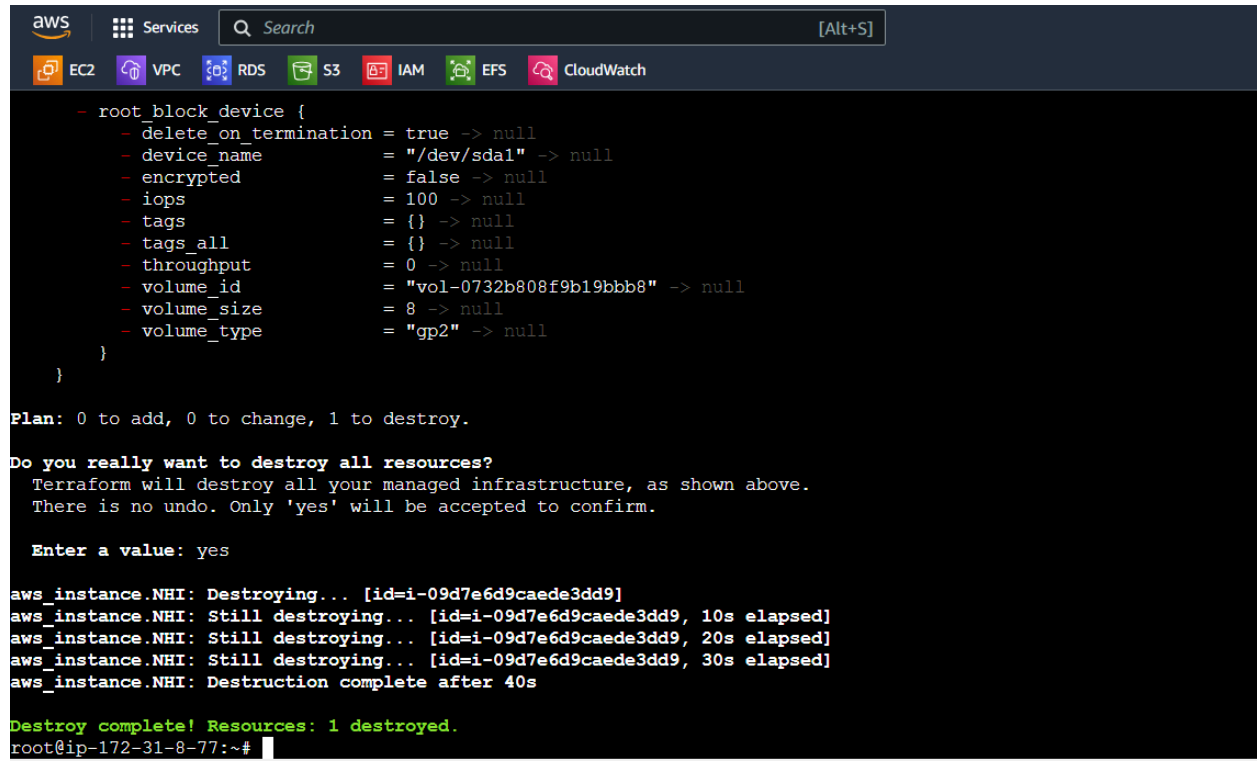


## Terraform Assignment - 2

You have been asked to:

- Destroy the previous deployment



```
- root_block_device {
-   delete_on_termination = true -> null
-   device_name           = "/dev/sda1" -> null
-   encrypted             = false -> null
-   iops                  = 100 -> null
-   tags                  = {} -> null
-   tags_all              = {} -> null
-   throughput            = 0 -> null
-   volume_id             = "vol-0732b808f9b19bbb8" -> null
-   volume_size           = 8 -> null
-   volume_type           = "gp2" -> null
}

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.NHI: Destroying... [id=i-09d7e6d9caede3dd9]
aws_instance.NHI: Still destroying... [id=i-09d7e6d9caede3dd9, 10s elapsed]
aws_instance.NHI: Still destroying... [id=i-09d7e6d9caede3dd9, 20s elapsed]
aws_instance.NHI: Still destroying... [id=i-09d7e6d9caede3dd9, 30s elapsed]
aws_instance.NHI: Destruction complete after 40s

Destroy complete! Resources: 1 destroyed.
root@ip-172-31-8-77:~#
```

- Create a new EC2 instance with an Elastic IP

```
provider "aws" {
  access_key = "AKIAVNPLXOGUOSZZY7UF"
  secret_key = "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"
  region     = "ap-south-1"
}

resource "aws_instance" "NHI" {
  tags = {
    Name = "pointer"
    env  = "NHI"
  }
}
```

```

}
ami      = "ami-03bb6d83c60fc5f7c"
instance_type = "t2.micro"
key_name = "Newword"
security_groups = ["default"]
}
resource "aws_eip" "lb" {
  domain = "vpc"
}

```

```
root@ip-172-31-8-77:~# terraform init
```

```
Initializing the backend...
```

```
Initializing provider plugins...
```

```
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.40.0
```

```
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.
```

```
root@ip-172-31-8-77:~# terraform plan
```

```
Plan: 2 to add, 0 to change, 0 to destroy.
```

```
Do you want to perform these actions?
```

```
Terraform will perform the actions described above.
```

```
Only 'yes' will be accepted to approve.
```

```
Enter a value: yes
```

```
aws_eip.lb: Creating...
```

```
aws_instance.NHI: Creating...
```

```
aws_eip.lb: Creation complete after 0s [id=eipalloc-012be792a06d7da71]
```

```
aws_instance.NHI: Still creating... [10s elapsed]
```

```
aws_instance.NHI: Still creating... [20s elapsed]
```

```
aws_instance.NHI: Still creating... [30s elapsed]
```

```
aws_instance.NHI: Creation complete after 32s [id=i-03ef6190f89289132]
```

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

- ▼ Images
  - AMIs
  - AMI Catalog

- ▼ Elastic Block Store
  - Volumes
  - Snapshots
  - Lifecycle Manager

- ▼ Network & Security
  - Security Groups
  - Elastic IPs**

Elastic IP addresses (1)

Actions ▼

Allocate Elastic IP address

Find resources by attribute or tag

< 1 >

<input type="checkbox"/>	Name ▼	Allocated IPv4 addr... ▼	Type ▼	Allocation ID
<input type="checkbox"/>	-	<a href="#">13.201.198.137</a>	Public IP	eipalloc-012be792a06d7da71

View IP address usage and recommendations to release unused IPs with [Public IP insights](#)