

TERRAFORM ASSIGNMENT-3

Name: Vikram

Assignment: Multi-Region EC2 Deployment (Ohio & N. Virginia)

Problem Statement

1. Destroy the previous deployment
2. Create two EC2 instances
 - One in Ohio (us-east-2)
 - One in N. Virginia (us-east-1)
3. Rename the instances:
 - Ohio → hello-ohio
 - Virginia → hello-virginia

Step 1: Destroy Previous Deployment

`terraform destroy -auto-approve`

```
ubuntu@ip-172-31-4-222:~/terraform-assignment2$ terraform destroy
aws_instance.new_ec2: Refreshing state... [id=i-043705974a254af64]
aws_eip.elastic_ip: Refreshing state... [id=eipalloc-086410904857b2574]

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

Terraform will perform the following actions:

# aws_eip.elastic_ip will be destroyed
- resource "aws_eip" "elastic_ip" {
  - allocation_id      = "eipalloc-086410904857b2574" -> null
  - arn                = "arn:aws:ec2:us-east-2:062250062838:elastic-ip/eipalloc-086410904857b2574" -> null
  - association_id     = "eipassoc-03b01e5e1c55c470" -> null
  - domain             = "vpc" -> null
  - id                = "eipalloc-086410904857b2574" -> null
  - instance           = "i-043705974a254af64" -> null
  - network_border_group = "us-east-2" -> null
  - network_interface  = "eni-006e43a7728927b75" -> null
}
```

```
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_eip.elastic_ip: Destroying... [id=eipalloc-086410904857b2574]
aws_eip.elastic_ip: Destruction complete after 2s
aws_instance.new_ec2: Destroying... [id=i-043705974a254af64]
aws_instance.new_ec2: Still destroying... [id=i-043705974a254af64, 00m10s elapsed]
aws_instance.new_ec2: Still destroying... [id=i-043705974a254af64, 00m20s elapsed]
aws_instance.new_ec2: Destruction complete after 30s

Destroy complete! Resources: 2 destroyed.
ubuntu@ip-172-31-4-222:~/terraform-assignment2$
```

Instances								
Instances (1/3) Info								
Find Instance by attribute or tag (case-sensitive)								
All states								
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	terraform	i-0ca0c7a99fa1c8500	Running	t3.small	3/3 checks passed	View alarms +	us-east-2a	ec2-18-118
<input checked="" type="checkbox"/>	Terraform-EC2...	i-043705974a254af64	Terminated	t3.micro	-	View alarms +	us-east-2c	-

Step 2: Create New Project

`mkdir terraform-assignment3`

`cd terraform-assignment3`

```
ubuntu@ip-172-31-4-222:~$ mkdir terraform-assignment3
cd terraform-assignment3
ubuntu@ip-172-31-4-222:~/terraform-assignment3$
```

Create main file:

`nano main.tf`

main.tf

```
provider "aws" {
```

```
    alias = "virginia"
```

```
    region = "us-east-1"
```

```
}
```

```
provider "aws" {
```

```
    alias = "ohio"
```

```
    region = "us-east-2"
```

```
}
```

```
# -----
```

```
# Virginia EC2 Instance
```

```
# -----
```

```
resource "aws_instance" "virginia_ec2" {
```

```
    provider    = aws.virginia
```

```
    ami         = "ami-0ecb62995f68bb549"
```

```
    instance_type = "t3.micro"
```

```
    subnet_id    = "subnet-0c0f78808cbf67247"
```

```
    tags = {
```

```

    Name = "hello-virginia"
}
}

# -----
# Ohio EC2 Instance
# -----

resource "aws_instance" "ohio_ec2" {

    provider    = aws.ohio

    ami         = "ami-0f5fcdcbd140e4ab7"

    instance_type = "t3.micro"

    tags = {

        Name = "hello-ohio"

    }
}

```

Save and exit

```
ubuntu@ip-172-31-4-222:~/terraform-assignment3$ nano main.tf
```

```

GNU nano 7.2
provider "aws" {
    alias    = "virginia"
    region  = "us-east-1"
}

provider "aws" {
    alias    = "ohio"
    region  = "us-east-2"
}

# -----
# Virginia EC2 Instance
# -----
resource "aws_instance" "virginia_ec2" {
    provider      = aws.virginia
    ami           = "ami-0ecb62995f68bb549"
    instance_type = "t3.micro"
    subnet_id     = "subnet-0c0f78808cbf67247"

    tags = {
        Name = "hello-virginia"
    }
}

^G Help      ^O Write Out  ^W Where Is   ^K Cut
^X Exit      ^R Read File  ^\ Replace    ^U Paste

```

i-0ca0c7a99fa1c8500 (terraform)

PublicIPs: 18.118.31.136 PrivateIPs: 172.31.4.222

Step 3: Initialize Terraform

terraform init

```
ubuntu@ip-172-31-4-222:~/terraform-assignment3$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v6.27.0

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-172-31-4-222:~/terraform-assignment3$
```

```
ubuntu@ip-172-31-4-222:~/terraform-assignment3$ terraform validate
Success! The configuration is valid.

ubuntu@ip-172-31-4-222:~/terraform-assignment3$
```

Step 4: Verify the Plan

terraform plan

```
ubuntu@ip-172-31-4-222:~/terraform-assignment3$ terraform plan

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.ohio_ec2 will be created
+ resource "aws_instance" "ohio_ec2" {
  + ami              = "ami-0f5fcd9bd140e4ab7"
  + arn              = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone = (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)
  + force_destroy     = false
  + 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)
}
```

Step 5: Apply the Deployment

terraform apply -auto-approve

```
ubuntu@ip-172-31-4-222:~/terraform-assignment3$ terraform apply -auto-approve
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.ohio_ec2 will be created
+ resource "aws_instance" "ohio_ec2" {
  + ami                    = "ami-0f5fcd140e4ab7"
  + arn                   = (known after apply)
  + associate_public_ip_address = (known after apply)
  + availability_zone      = (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)
  + force_destroy          = false
  + 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)
}
```

```
Plan: 2 to add, 0 to change, 0 to destroy.
aws_instance.virginia_ec2: Creating...
aws_instance.ohio_ec2: Creating...
aws_instance.virginia_ec2: Still creating... [00m10s elapsed]
aws_instance.ohio_ec2: Still creating... [00m10s elapsed]
aws_instance.ohio_ec2: Creation complete after 12s [id=i-0ebc12e4ce38ef131]
aws_instance.virginia_ec2: Creation complete after 12s [id=i-023c4915d5e60e9b3]

Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
ubuntu@ip-172-31-4-222:~/terraform-assignment3$
```

i-0ca0c7a99fa1c8500 (terraform)

Step 6: Verification

In AWS Console:

N. Virginia Region (us-east-1)

✓ Instance Name → [hello-virginia](#)

Ohio Region (us-east-2)

✓ Instance Name → [hello-ohio](#)

The screenshot shows the AWS Management Console interface. At the top, there's a search bar and navigation tabs. The main section is titled 'Instances (1/1) Info'. Below this, there's a table with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4. The table contains one entry: 'hello-virginia' with ID 'i-023c4915d5e60e9b3', state 'Running', type 't3.micro', status 'Initializing', and public IP 'ec2-34-228-238-68'. Below the table, there's a detailed view for the selected instance 'i-023c4915d5e60e9b3 (hello-virginia)'. This view includes tabs for 'Details', 'Status and alarms', 'Monitoring', 'Security', 'Networking', 'Storage', and 'Tags'. The 'Details' tab is active, showing the 'Instance summary' with fields for Instance ID, Public IPv4 address, Private IPv4 addresses, Instance state (Running), and Public DNS.

Search

[Alt+S] Ask Amazon Q

United States (Ohio)

Account ID: 0622-5006-2838

root

Instances

Instances (1/2) Info

Last updated less than a minute ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

Running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
<input type="checkbox"/>	terraform	i-0ca0c7a99fa1c8500	Running	t3.small	3/3 checks passed	View alarms +	us-east-2a	ec2-18-118
<input checked="" type="checkbox"/>	hello-ohio	i-0ebc12e4ce38ef131	Running	t3.micro	Initializing	View alarms +	us-east-2c	ec2-3-19-14

Unselect instance: hello-ohio

i-0ebc12e4ce38ef131 (hello-ohio)

AMI ID

ami-0f5fcdffd140e4ab7

Monitoring

disabled

Platform details

Linux/UNIX

AMI name

ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20251022

Allowed image

-

Termination protection

Disabled

Stop protection

Disabled

Launch time

Thu Dec 25 2025 13:00:18 GMT+0530 (India Standard Time)

AMI location

amazon/ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-server-20251022

Conclusion

- ✓ Destroyed old deployment
- ✓ Created EC2 in two different regions
- ✓ Correctly named both instances
- ✓ Successfully managed multi-region infrastructure with Terraform