

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
 - o One in Ohio (us-east-2)
 - o One in N. Virginia (us-east-1)
 3. Rename the instances:
 - o Ohio → hello-ohio
 - o 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-03b01e5ee1c55c470" -> 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							
Actions		Last updated	Connect	Instance state	Actions	Launch instances	
		less than a minute ago		All states			
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
	<input type="checkbox"/> terraform	i-0ca0c7a99fa1c8500	Running Q Q	t3.small	3/3 checks passed View alarms +	us-east-2a	ec2-18-118
	<input checked="" type="checkbox"/> Terraform-EC2...	i-043705974a254af64	Terminated Q Q	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-0f5fcdfbd140e4ab7"
    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)

Public IPs: 18.118.31.136 Private IPs: 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
    + arn
    + associate_public_ip_address
    + availability_zone
    + disable_api_stop
    + disable_api_termination
    + ebs_optimized
    + enable_primary_ipv6
    + force_destroy
    + get_password_data
    + host_id
    + host_resource_group_arn
    + iam_instance_profile
    + id
        = "ami-0f5fcdfbd140e4ab7"
        = (known after apply)
        = false
        = false
        = (known after apply)
        = (known after apply)
        = (known after apply)
        = (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
  + arn
  + associate_public_ip_address
  + availability_zone
  + disable_api_stop
  + disable_api_termination
  + ebs_optimized
  + enable_primary_ipv6
  + force_destroy
  + get_password_data
  + host_id
  + host_resource_group_arn
  + iam_instance_profile
  + id
  + ...
  + ...
}
```

```
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 CloudWatch Metrics interface. A single metric named 'Hello' is displayed with a value of 1. The metric has a timestamp of 2023-06-28T14:00:00Z. The chart shows a single data point at time 0.

The screenshot shows the AWS CloudWatch Metrics Insights interface. A search bar at the top contains the query: `CloudWatch Metrics Insights metrics`. Below the search bar, there are two tabs: `Metrics` and `Logs`, with `Metrics` selected. The main area displays a table of metrics with the following columns: Metric Name, Value, Unit, and Last Value. The table includes rows for `CloudWatch Metrics Insights metrics` and `CloudWatch Metrics Insights metrics` (1 hour ago). At the bottom of the interface, there is a section titled "Metrics from CloudWatch Metrics Insights" with a table showing metrics like `CloudWatch Metrics Insights metrics` and `CloudWatch Metrics Insights metrics`.

Conclusion

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