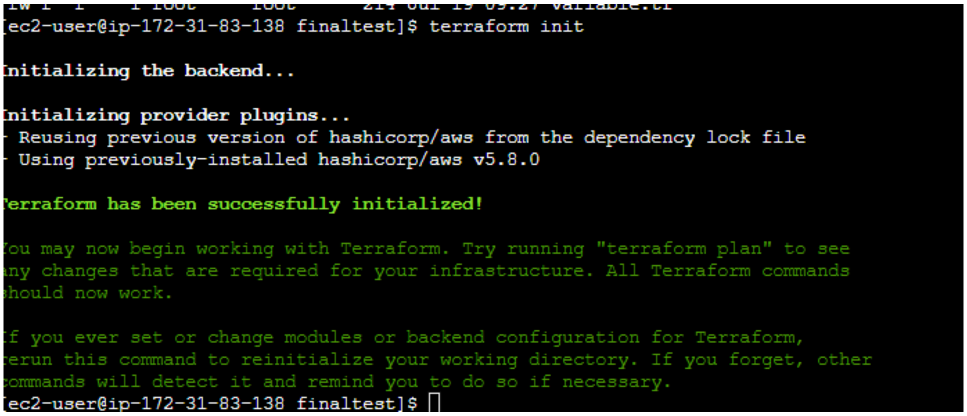
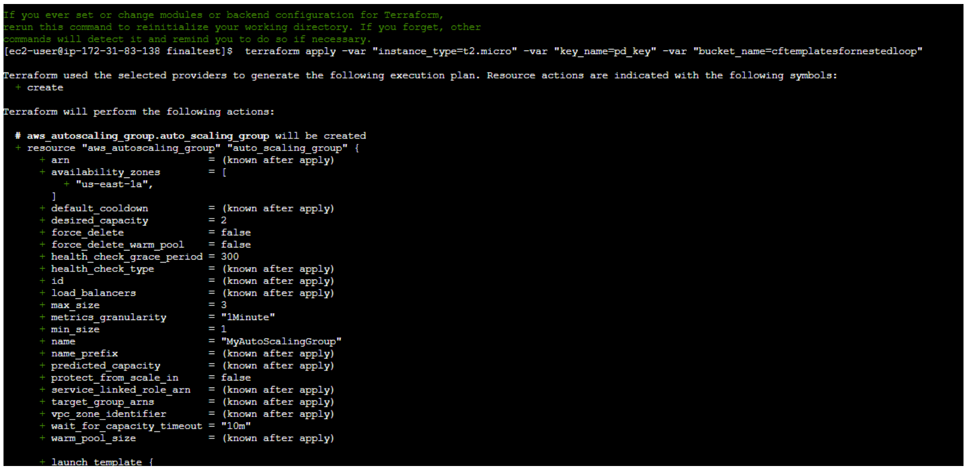
Steps Performed:

1.terraform init

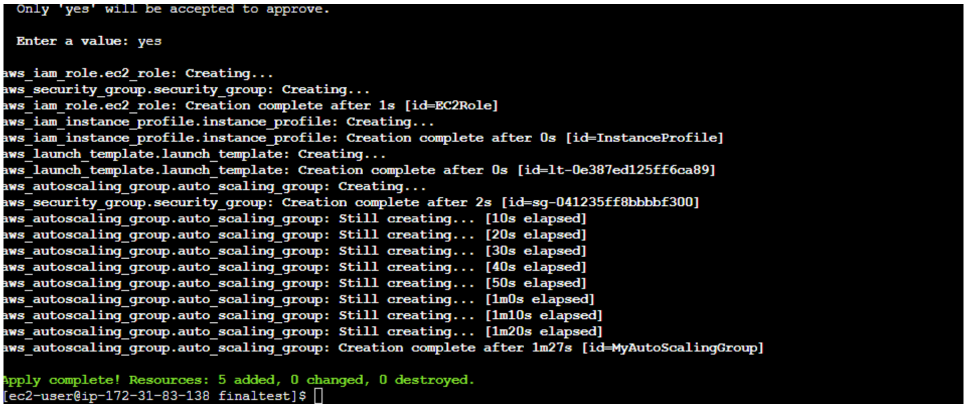


```
rw r  r    1 root     root      214 Jul 19 09:27 variable.tf
[ec2-user@ip-172-31-83-138 finaltest]$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.8.0

Terraform has been successfully initialized!

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

f you ever set or change modules or backend configuration for Terraform,
erun this command to reinitialize your working directory. If you forget, other
ommands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-83-138 finaltest]$ 
```

2.terraform apply-var "instance_type=t2.micro" -var "key_name=pd_key" -var "bucket_name=cftemplatesfornestedloop"

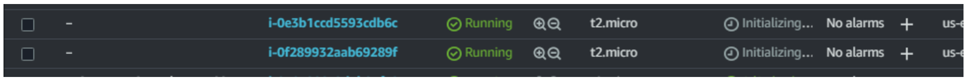


```
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-83-138 finaltest]$  terraform apply -var "instance_type=t2.micro" -var "key_name=pd_key" -var "bucket_name=cftemplatesfornestedloop"

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_autoscaling_group.auto_scaling_group will be created
  + resource "aws_autoscaling_group" "auto_scaling_group" {
      + arn                       = (known after apply)
      + availability_zones        = [
          + "us-east-1a",
        ]
      + default_cooldown          = (known after apply)
      + desired_capacity          = 2
      + force_delete              = false
      + force_delete_warm_pool    = false
      + health_check_grace_period = 300
      + health_check_type         = (known after apply)
      + id                        = (known after apply)
      + load_balancers            = (known after apply)
      + max_size                  = 3
      + metrics_granularity       = "1Minute"
      + min_size                  = 1
      + name                      = "MyAutoScalingGroup"
      + name_prefix               = (known after apply)
      + predicted_capacity        = (known after apply)
      + protect_from_scale_in     = false
      + service_linked_role_arn   = (known after apply)
      + target_group_arns         = (known after apply)
      + vpc_zone_identifier       = (known after apply)
      + wait_for_capacity_timeout = "10m"
      + warm_pool_size            = (known after apply)

      + launch_template {
```



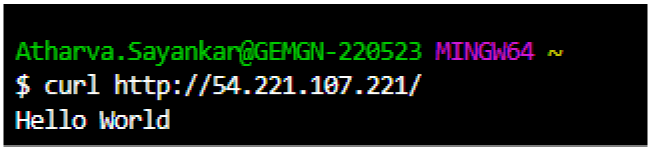
```
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_iam_role.ec2_role: Creating...
aws_security_group.security_group: Creating...
aws_iam_role.ec2_role: Creation complete after 1s [id=EC2Role]
aws_iam_instance_profile.instance_profile: Creating...
aws_iam_instance_profile.instance_profile: Creation complete after 0s [id=InstanceProfile]
aws_launch_template.launch_template: Creating...
aws_launch_template.launch_template: Creation complete after 0s [id=lt-0e387ed125ff6ca89]
aws_autoscaling_group.auto_scaling_group: Creating...
aws_security_group.security_group: Creation complete after 2s [id=sg-041235ff8bbbbf300]
aws_autoscaling_group.auto_scaling_group: Still creating... [10s elapsed]
aws_autoscaling_group.auto_scaling_group: Still creating... [20s elapsed]
aws_autoscaling_group.auto_scaling_group: Still creating... [30s elapsed]
aws_autoscaling_group.auto_scaling_group: Still creating... [40s elapsed]
aws_autoscaling_group.auto_scaling_group: Still creating... [50s elapsed]
aws_autoscaling_group.auto_scaling_group: Still creating... [1m0s elapsed]
aws_autoscaling_group.auto_scaling_group: Still creating... [1m10s elapsed]
aws_autoscaling_group.auto_scaling_group: Still creating... [1m20s elapsed]
aws_autoscaling_group.auto_scaling_group: Creation complete after 1m27s [id=MyAutoScalingGroup]

Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-83-138 finaltest]$ 
```

Instances created successfully:



□	-	i-0e3b1ccd5593cdb6c	Running		t2.micro	Initializing...	No alarms	+	us-e
□	-	i-0f289932aab69289f	Running		t2.micro	Initializing...	No alarms	+	us-e

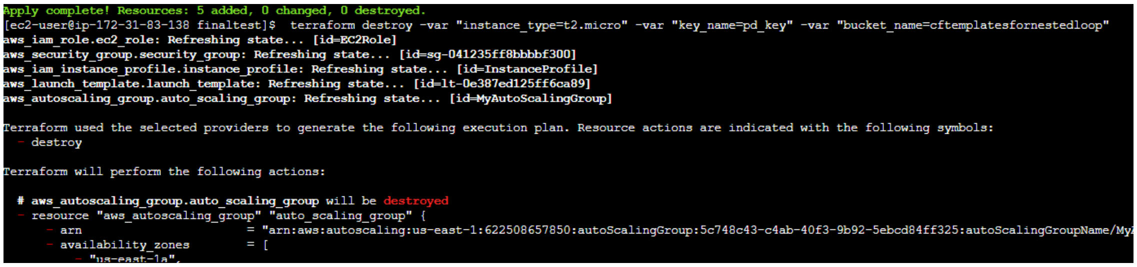
curl the instance IP:



```
Atharva.Sayankar@GEMGN-220523 MINGW64 ~
$ curl http://54.221.107.221/
Hello World
```

Cleanup:

terraform destroy -var "instance_type=t2.micro" -var "key_name=pd_key" -var "bucket_name=cftemplatesfornestedloop"



```
Apply complete! Resources: 5 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-83-138 finaltest]$  terraform destroy -var "instance_type=t2.micro" -var "key_name=pd_key" -var "bucket_name=cftemplatesfornestedloop"
aws_iam_role.ec2_role: Refreshing state... [id=EC2Role]
aws_security_group.security_group: Refreshing state... [id=sg-041235ff8bbbbf300]
aws_iam_instance_profile.instance_profile: Refreshing state... [id=InstanceProfile]
aws_launch_template.launch_template: Refreshing state... [id=lt-0e387ed125ff6ca89]
aws_autoscaling_group.auto_scaling_group: Refreshing state... [id=MyAutoScalingGroup]

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_autoscaling_group.auto_scaling_group will be destroyed
  - resource "aws_autoscaling_group" "auto_scaling_group" {
      - arn                       = "arn:aws:autoscaling:us-east-1:622508657850:autoScalingGroup:5c748c43-c4ab-40f3-9b92-5ebcd84ff325:autoScalingGroupName/My
      - availability_zones        = [
          - "us-east-1a",
```