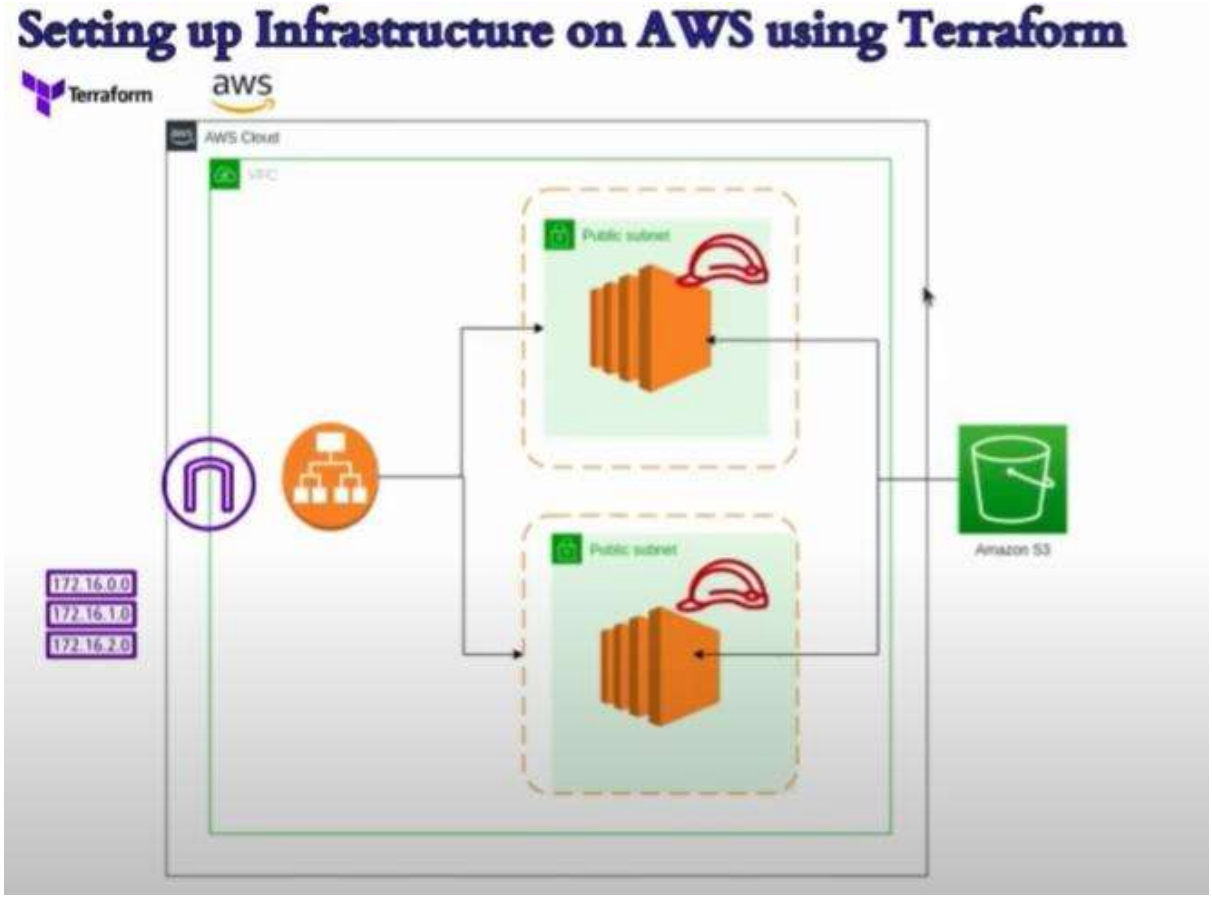
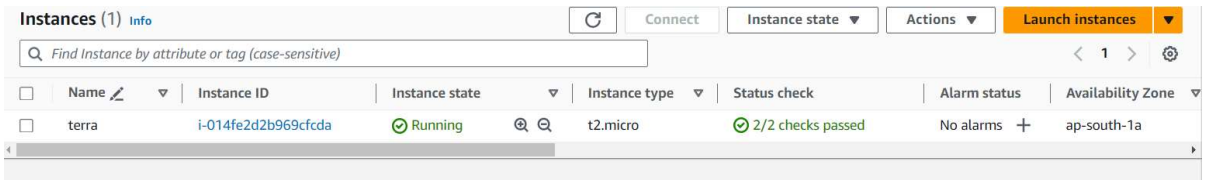
AWS – VPC






```
ubuntu@ip-172-31-46-83:~/TERRAFORM$ wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d
/hashicorp.list
sudo apt update && sudo apt install terraform
```

```
ubuntu@ip-172-31-46-83:~/TERRAFORM$ ls
main.tf  provider.tf  terraform.tfstate  terraform.tfstate.backup  userdata.sh  userdata1.sh
```

Github code link:

```
ubuntu@ip-172-31-46-83:~/TERRAFORM$ terraform validate
Success! The configuration is valid.
```

```
ubuntu@ip-172-31-46-83:~/TERRAFORM$ 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_iam_instance_profile.instance_profile will be created
  + resource "aws_iam_instance_profile" "instance_profile" {
      + arn         = (known after apply)
      + create_date = (known after apply)
      + id          = (known after apply)
      + name        = "instanceprofile"
      + name_prefix = (known after apply)
      + path        = "/"
      + role        = "instance_role"
      + tags_all    = (known after apply)
      + unique_id   = (known after apply)
    }
```

```
Plan: 20 to add, 0 to change, 0 to destroy.

Changes to Outputs:
  + loadbalancerdns = (known after apply)
```

```
ubuntu@ip-172-31-46-83:~/TERRAFORM$ 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_iam_instance_profile.instance_profile will be created
  + resource "aws_iam_instance_profile" "instance_profile" {
```

```
aws_instance.terra-2: Creation complete after 41s [id=i-03cb770c493c9df77]
aws_lb_target_group_attachment.tg-attach-2: Creating...
aws_lb_target_group_attachment.tg-attach-1: Creation complete after 0s [id=arn:aws:elasticloadbalancing:ap-south-1:687157172064:targetgroup/tf-lb-tg/9570a3847ec5043b-20231213061315302500000004]
aws_lb_target_group_attachment.tg-attach-2: Creation complete after 0s [id=arn:aws:elasticloadbalancing:ap-south-1:687157172064:targetgroup/tf-lb-tg/9570a3847ec5043b-20231213061315413400000005]
aws_lb.lb: Still creating... [50s elapsed]
aws_lb.lb: Still creating... [1m0s elapsed]
aws_lb.lb: Still creating... [1m10s elapsed]
aws_lb.lb: Still creating... [1m20s elapsed]
aws_lb.lb: Still creating... [1m30s elapsed]
aws_lb.lb: Still creating... [1m40s elapsed]
aws_lb.lb: Still creating... [1m50s elapsed]
aws_lb.lb: Still creating... [2m0s elapsed]
aws_lb.lb: Still creating... [2m10s elapsed]
aws_lb.lb: Still creating... [2m20s elapsed]
aws_lb.lb: Still creating... [2m30s elapsed]
aws_lb.lb: Still creating... [2m40s elapsed]
aws_lb.lb: Creation complete after 2m42s [id=arn:aws:elasticloadbalancing:ap-south-1:687157172064:loadbalancer/app/LB-T/7b53293300eebe05]
aws_lb_listener.lb_listner: Creating...
aws_lb_listener.lb_listner: Creation complete after 0s [id=arn:aws:elasticloadbalancing:ap-south-1:687157172064:listener/app/LB-T/7b53293300eebe05/c016a475c99b3ce0]

Apply complete! Resources: 20 added, 0 changed, 0 destroyed.

Outputs:

loadbalancerdns = "LB-T-1412627750.ap-south-1.elb.amazonaws.com"
ubuntu@ip-172-31-46-83:~/TERRAFORM$
```

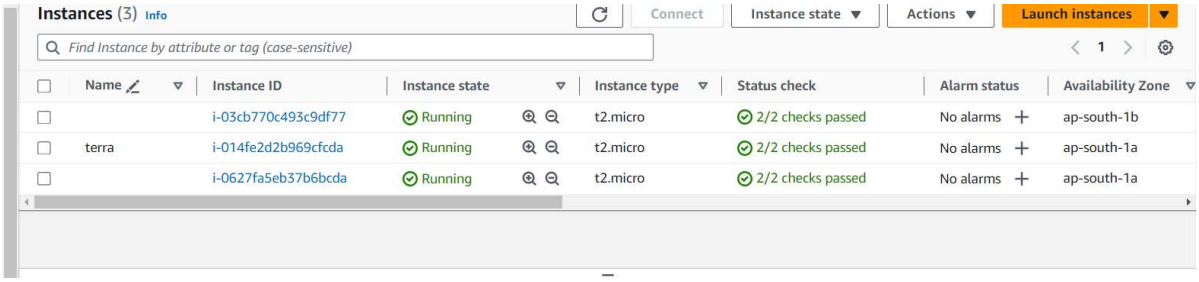
Terraform Project Server 1

Instance ID: i-03cb770c493c9df77

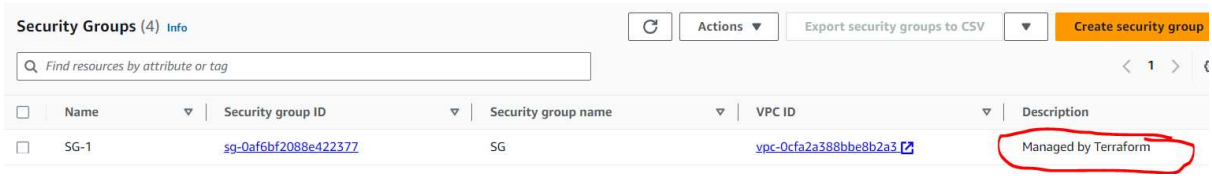
When I click on refresh button LB distribute traffic between 2 subnets.

Resources Created by Terraform:

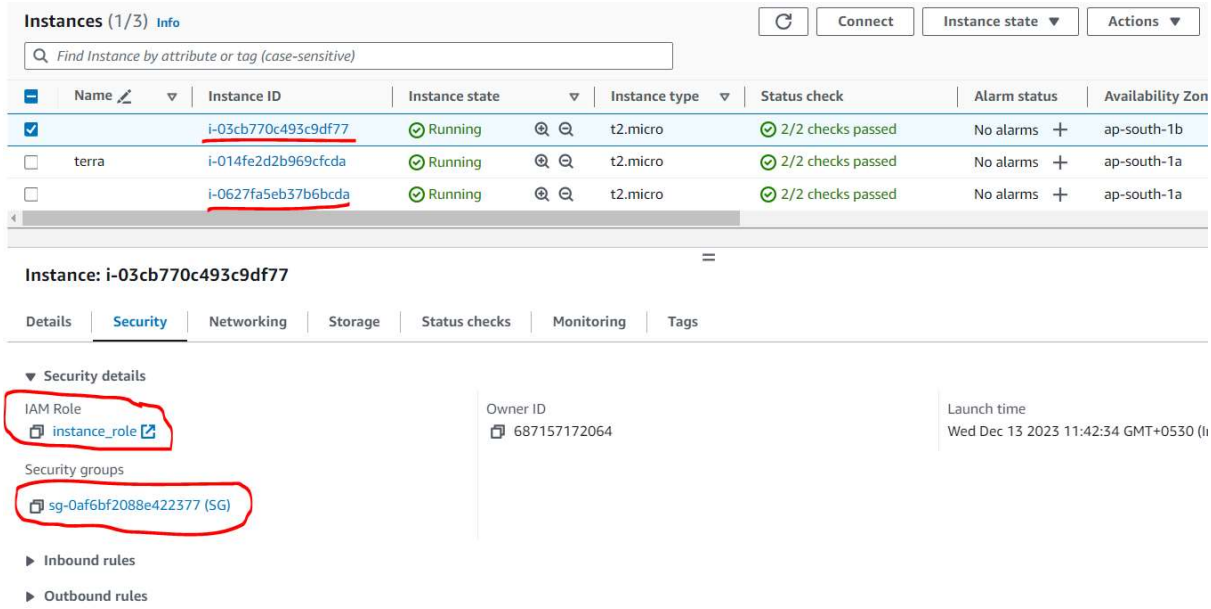
EC2:



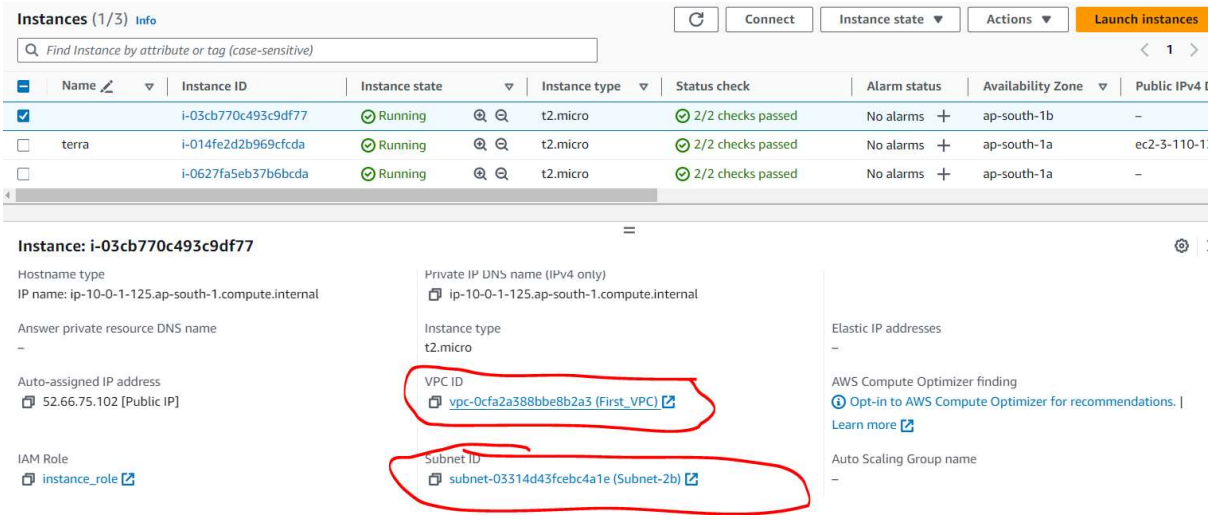
SECURITY GROUP:



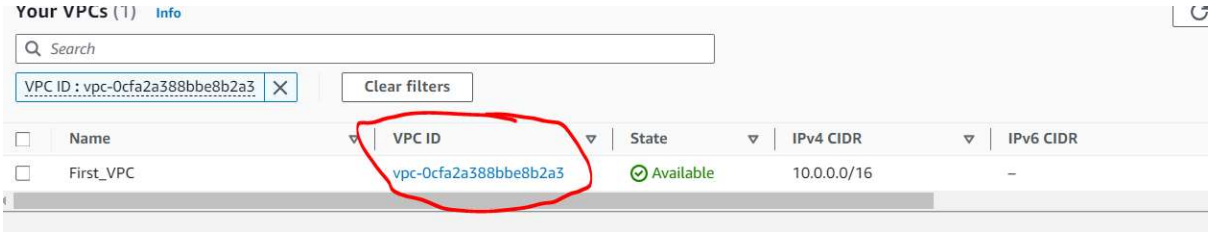
Security group and IAM Role attached to ec2-instances.



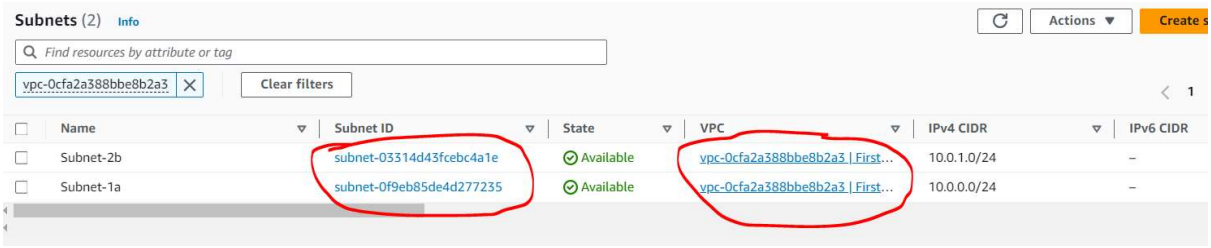
VPC & Subnets:



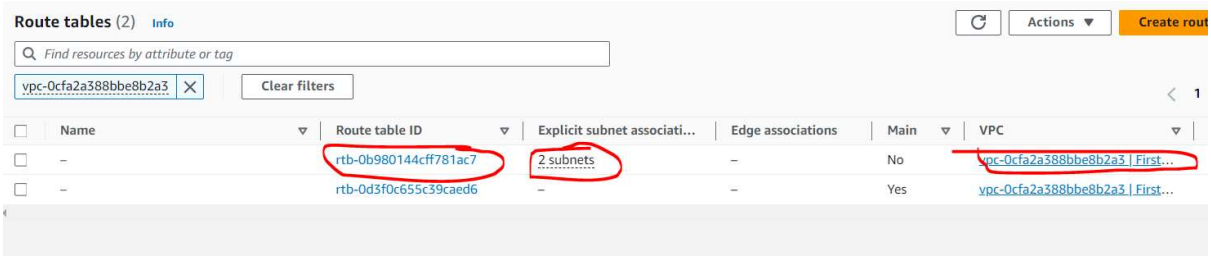
VPC:



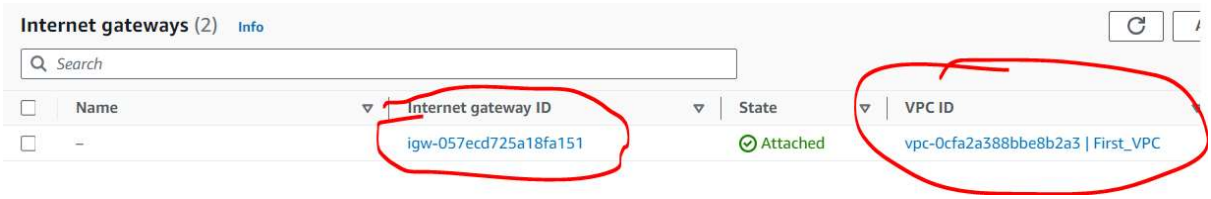
Subnets:



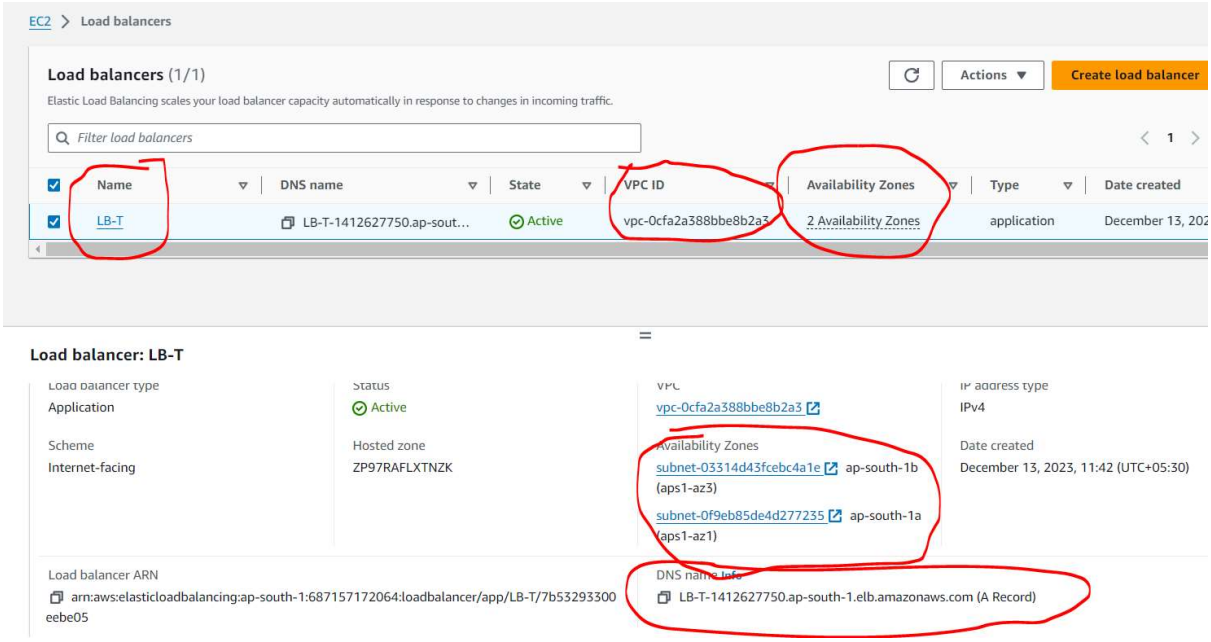
Route-Table:



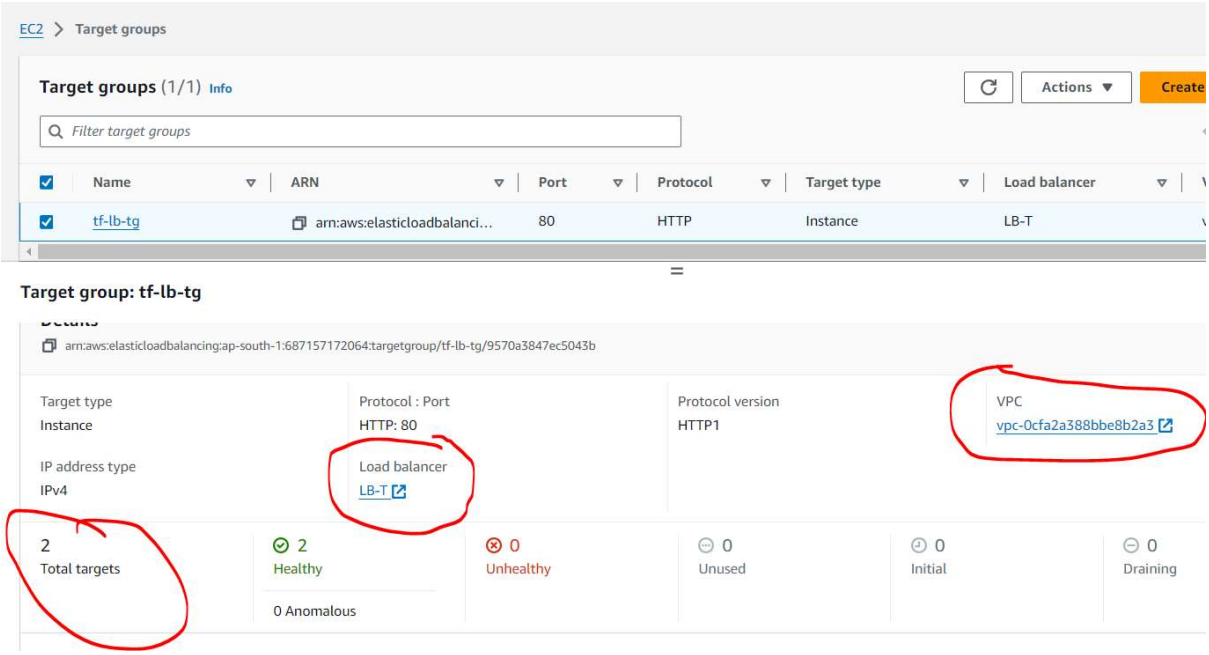
IGW:



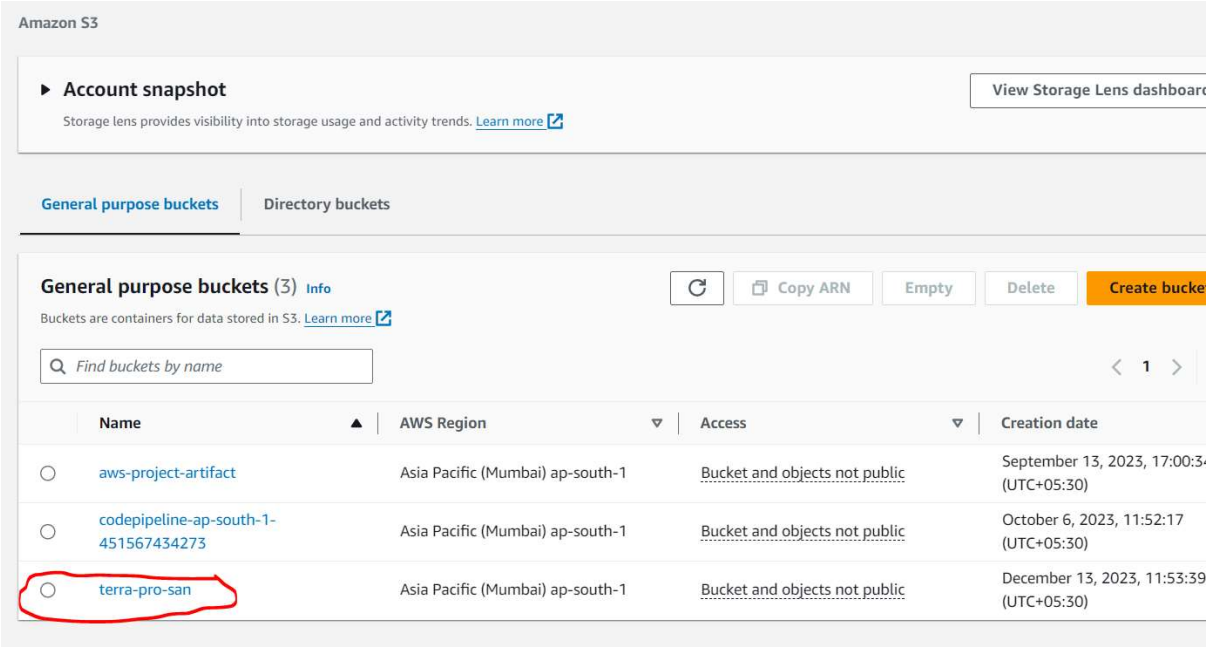
Load Balancer:



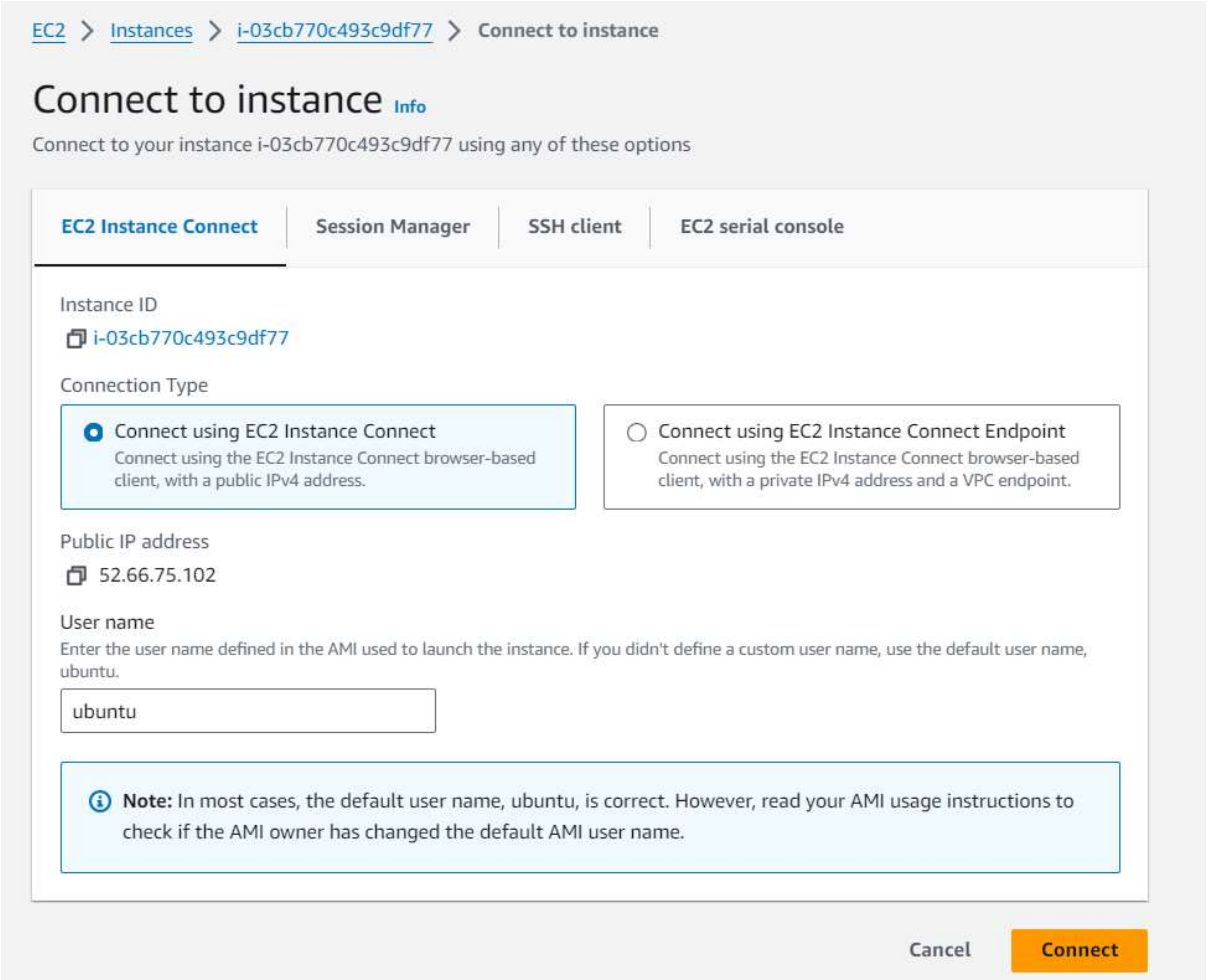
Target Group:



S3 Bucket:



Connect one of the instance to check instance able to access s3 bucket or not (as per attached IAM ROLE) :




```
ubuntu@ip-10-0-1-125:~$ ls
ubuntu@ip-10-0-1-125:~$ aws s3 ls
2023-09-13 11:30:34 aws-project-artifact
2023-10-06 06:22:17 codepipeline-ap-south-1-451567434273
2023-12-13 06:23:39 terra-pro-san
ubuntu@ip-10-0-1-125:~$ sudo aws s3 cp s3://terra-pro-san/TERRAFORM.png /var/www/html/TERRAFORM.png
download: s3://terra-pro-san/TERRAFORM.png to ../../var/www/html/TERRAFORM.png
ubuntu@ip-10-0-1-125:~$ cd ../..//var/www/html
ubuntu@ip-10-0-1-125:/var/www/html$ ls
TERRAFORM.png  index.html
ubuntu@ip-10-0-1-125:/var/www/html$
```

Terraform Destroy: