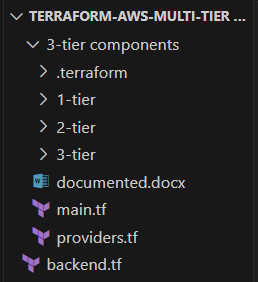
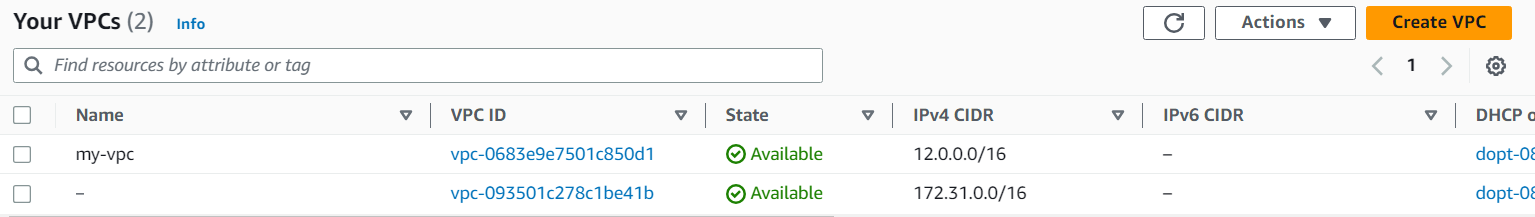
**TERRFORM-AWS-MULTI-TIER ARCHITECTURE SETUP**

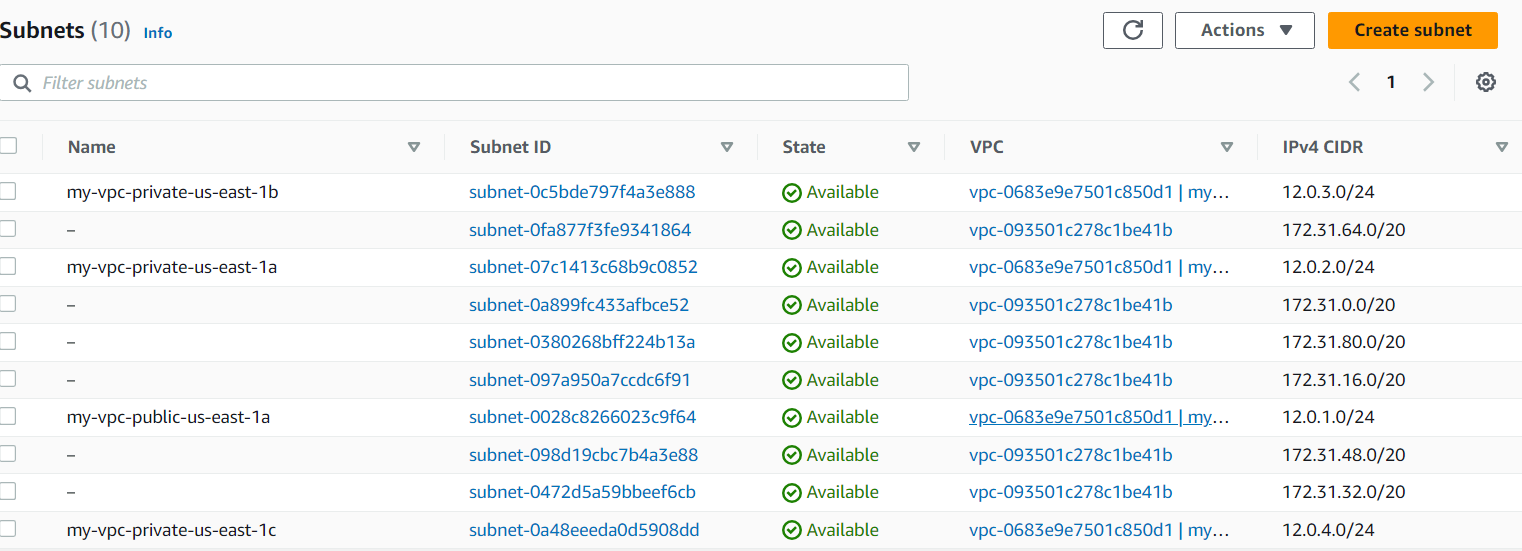
**FILE ARCHITECTURE**

****

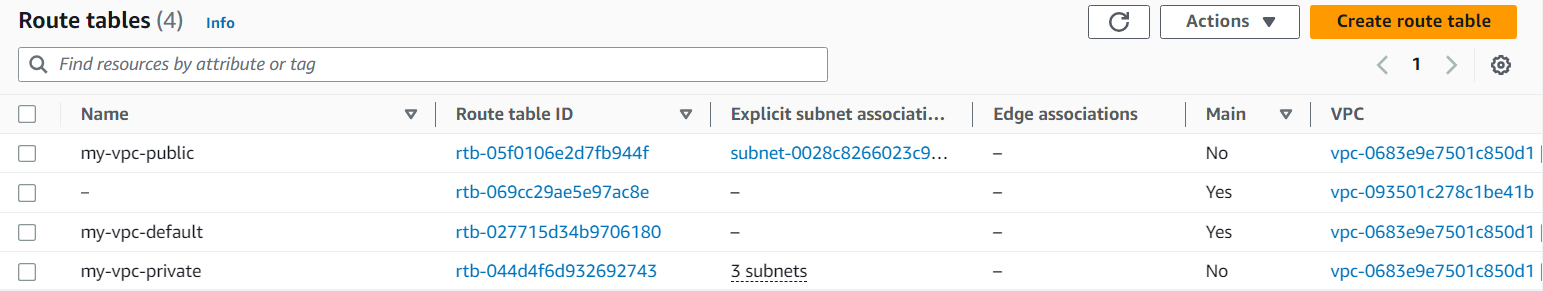
**VPC IN AWS via TERRAFORM (specifically used terraform registry module for creating vpc)**

****

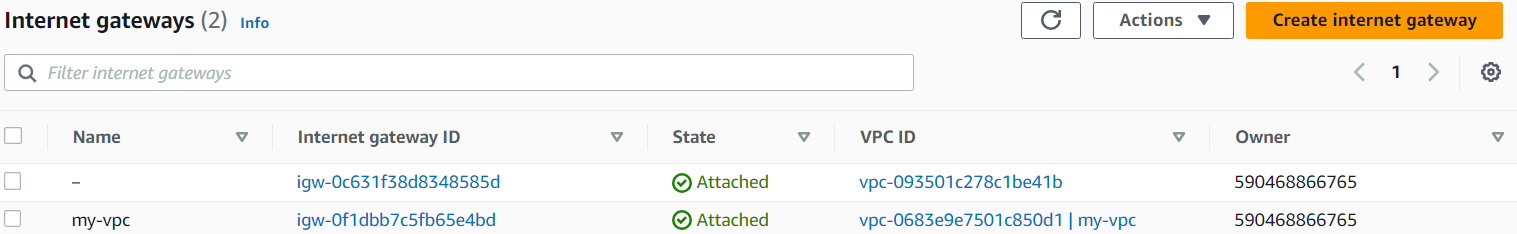
**SUBNETS CREATED**

****

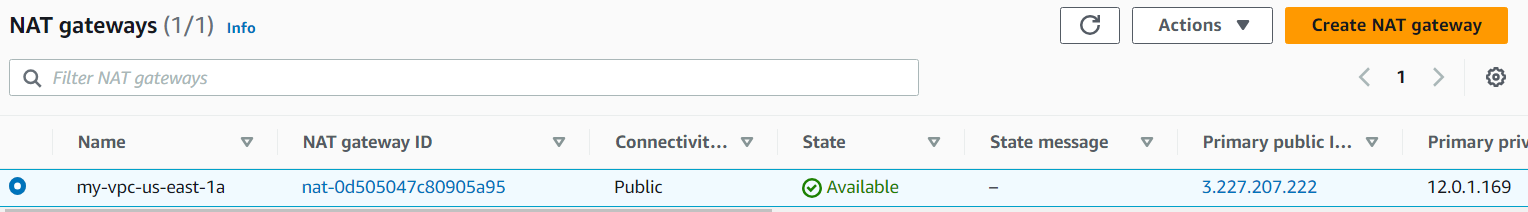
**ROUTE TABLES**

****

**IGW**

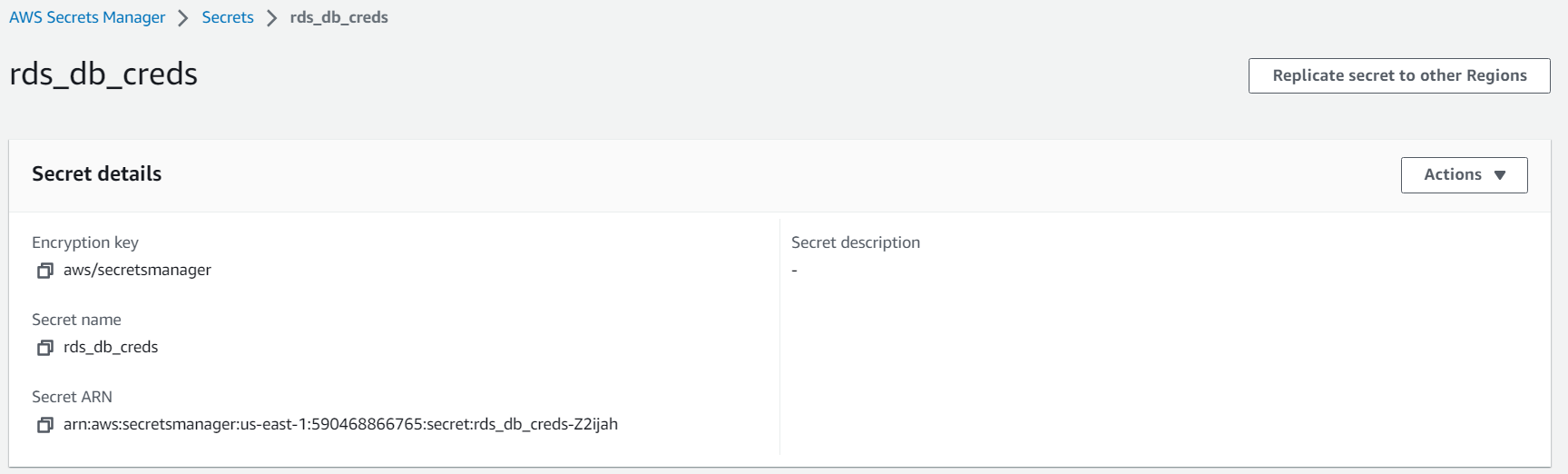
****

**NAT GW**

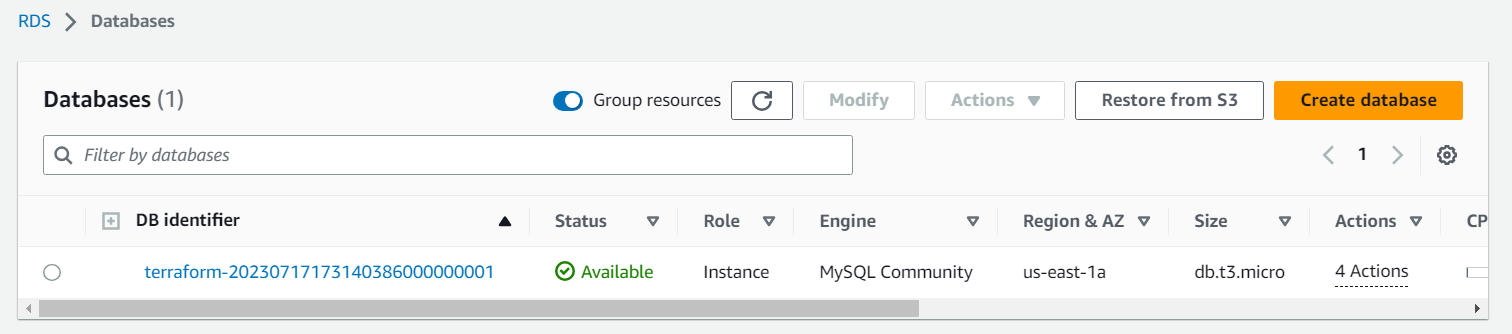
****

**Tier-3(used modular approach)**

**AWS secrets manager service is used to store username and password of database.**

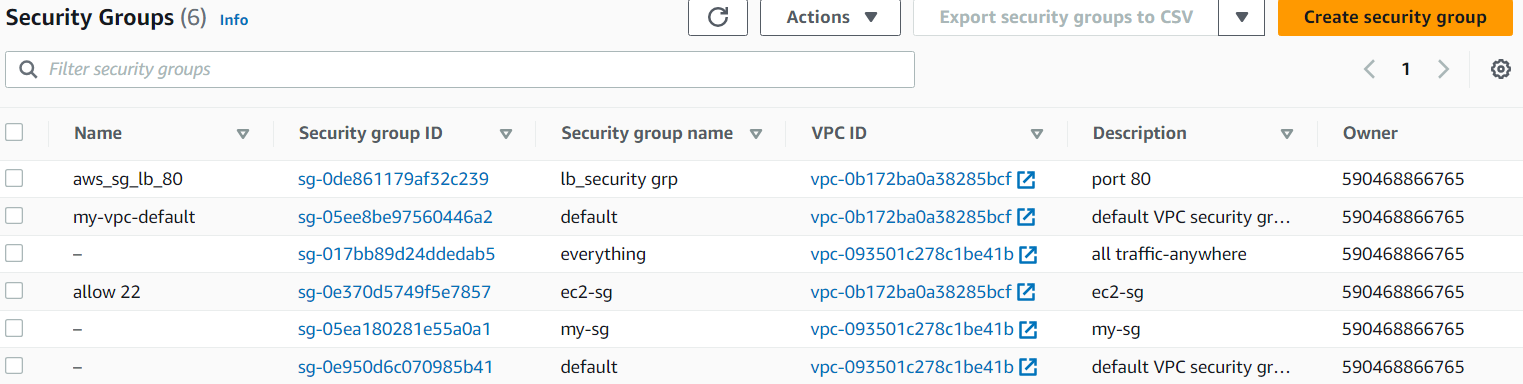
****

**Created database**

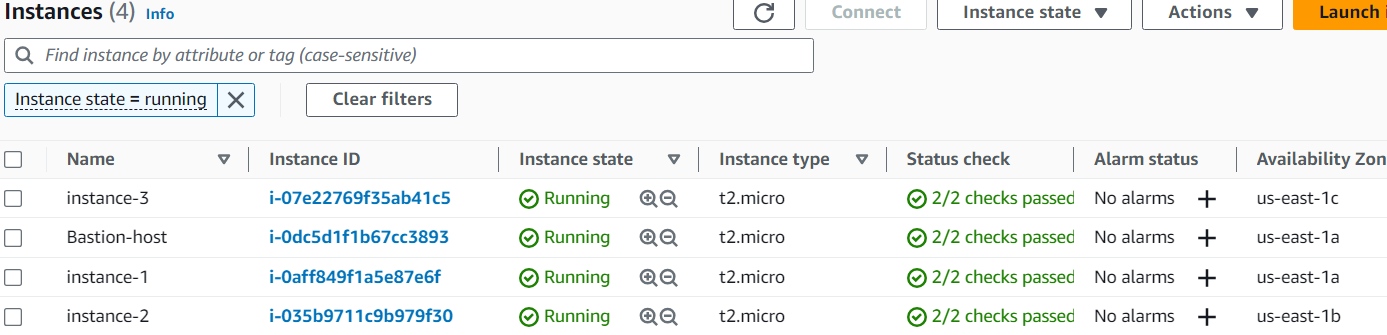
****

**Tier-2(used modular approach)**

**Created security group (allow 22) for ec2 (added security group of lb so that the ec2 can access the traffic from load balancer(lb)**

****

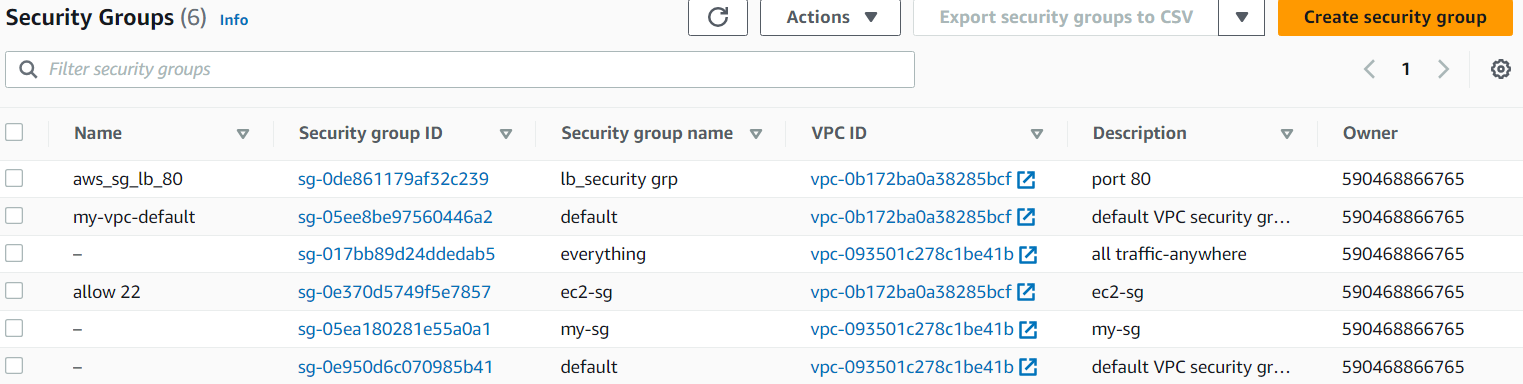
**Created 4 ec2 instances – 3 in private network and 1 in public network (acts as bastion host)**

****

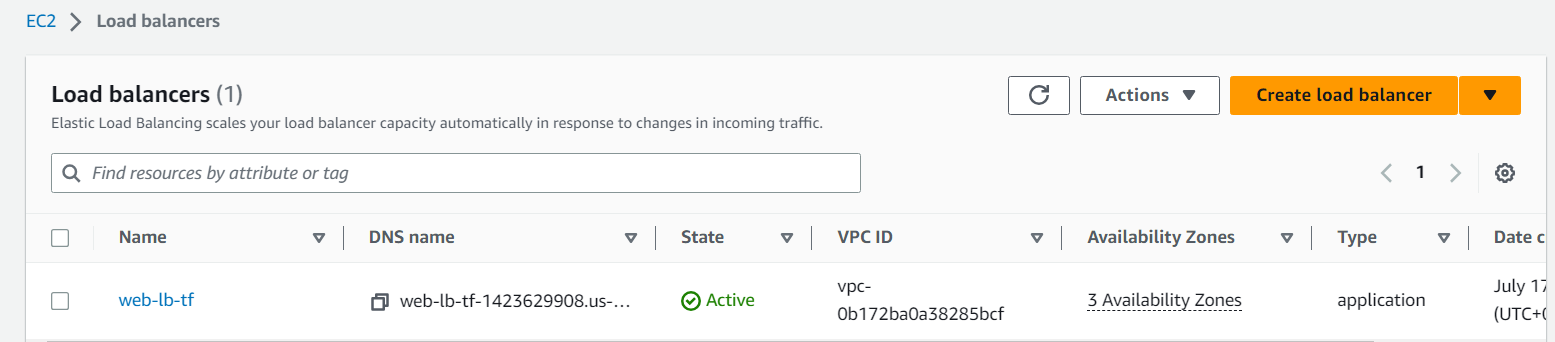
**Here we need to setup our website manually through bastion host until then the target group you will see all the instances flagged as unhealthy. Hint-target grp flag an instance as healthy only when response comes from port 80.**

**Tier – 1(used modular approach)**

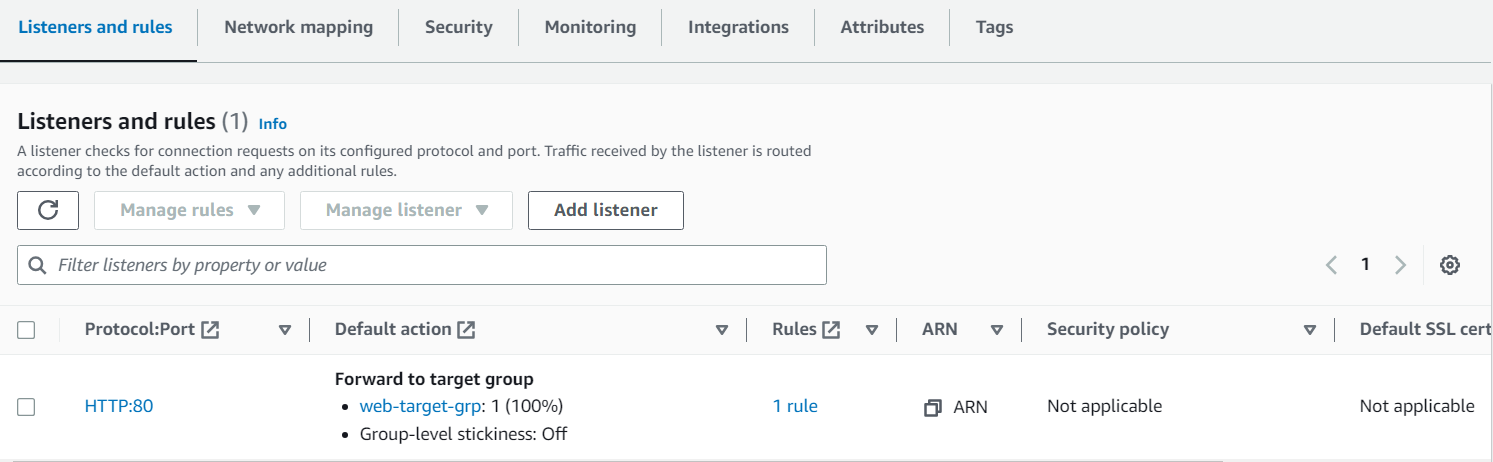
**Created security group for load balancer (aws\_sg\_lb\_80)**

****

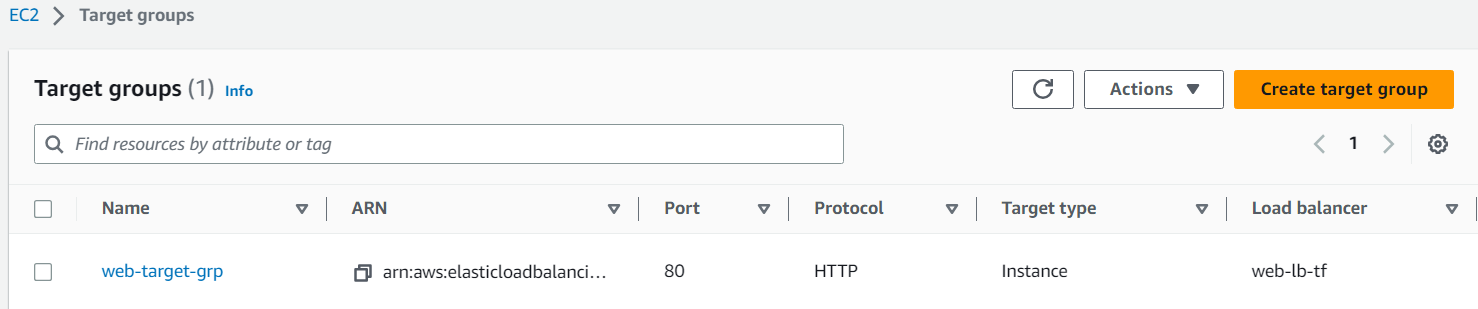
**Created load balancer**

****

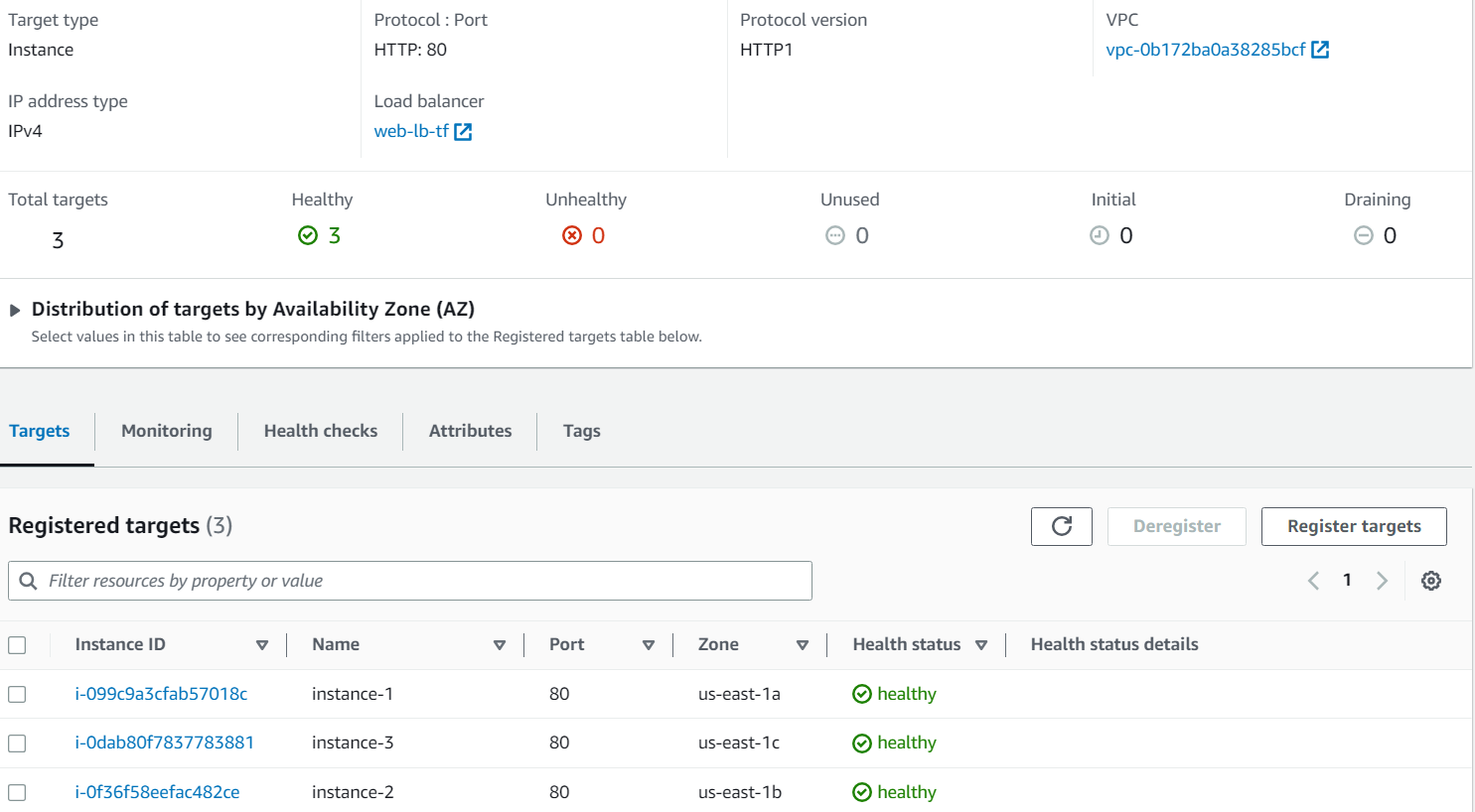
**Created listener on port 80 for load Balancer on which it will listen to the incoming requests**

****

**Created target grp for lb**

****

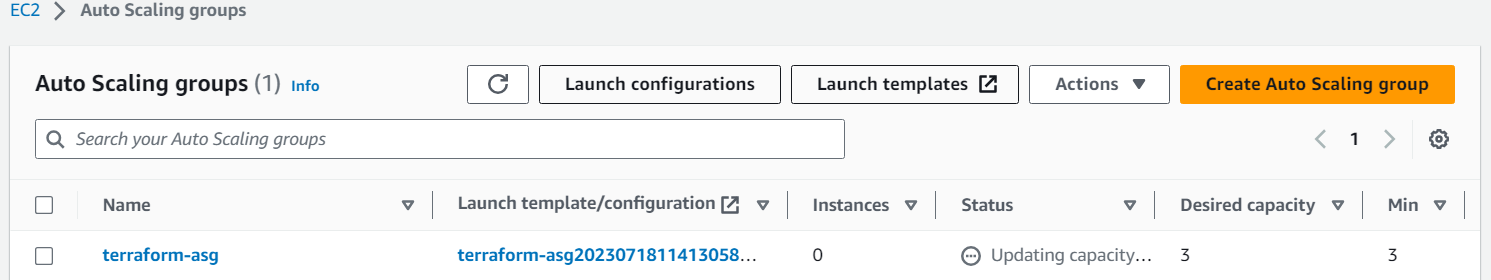
**Attached instances to the target group**

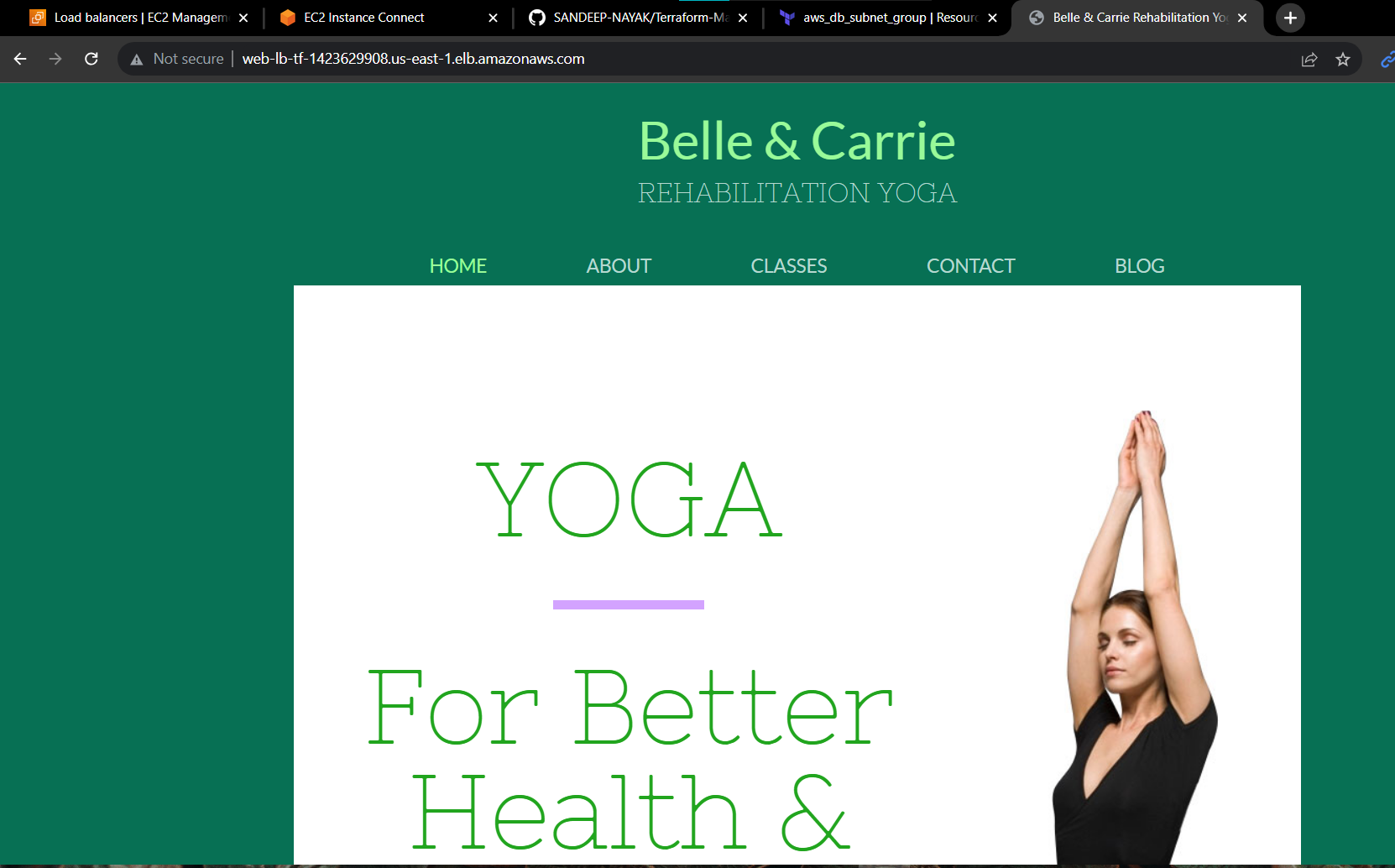
****

**Key take away from here is that:**

**You have 3 instances in 3 different private subnets in 3 different zones. Since private instances cannot be accessed directly via internet coz of no public IP so they cannot be directly attached with the lb and hence you need 3 different public subnets in 3 different zones just like above and you have to map them with the subnets attribute of lb which will lead to the detection of them and hence leading to their addition to target group via attachment resource. This specific pattern is known as “load Balancer Sandwich”.**

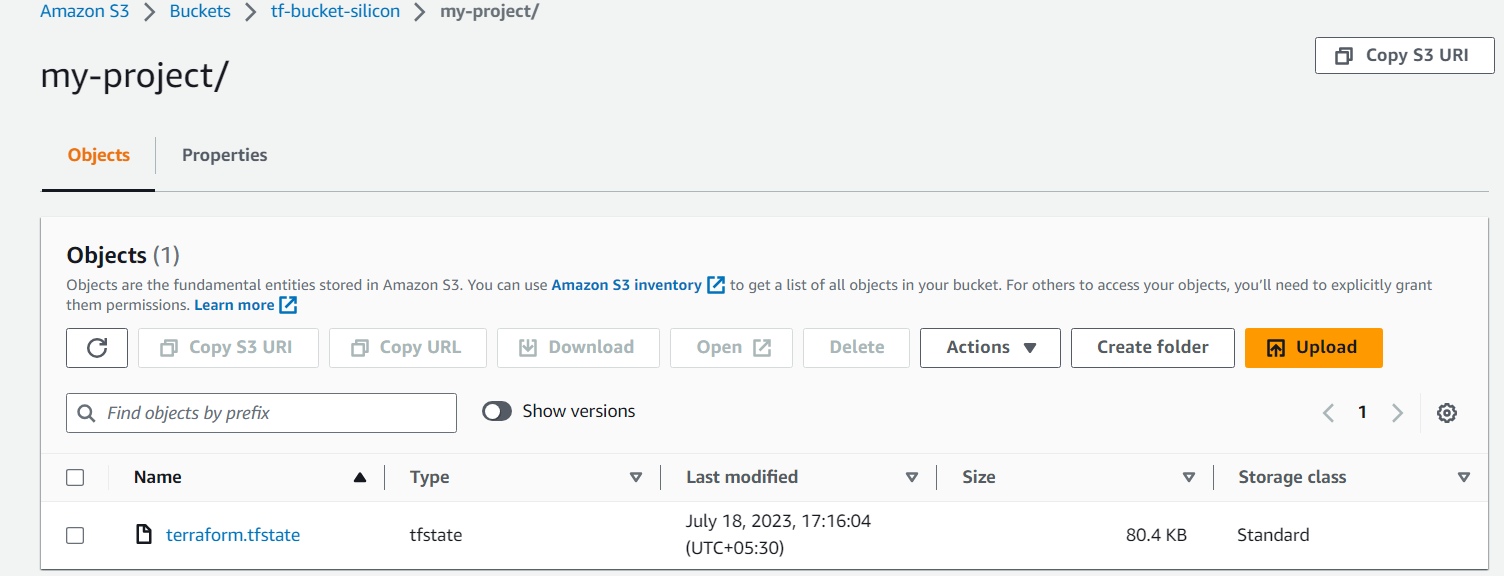
**Created Autoscaling group to scale up and scale down as per the incoming traffic.**

****

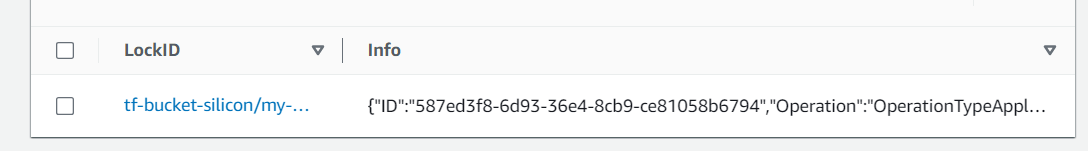
****

**backend.tf**

* **stored terraform.tfstate file remotely using s3 bucket as a best practice.**

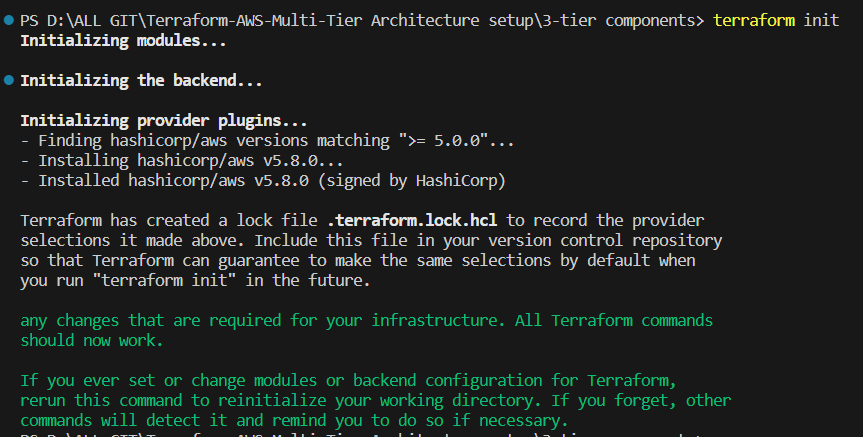


* **implemented state locking using DynamoDB by creating a table.**

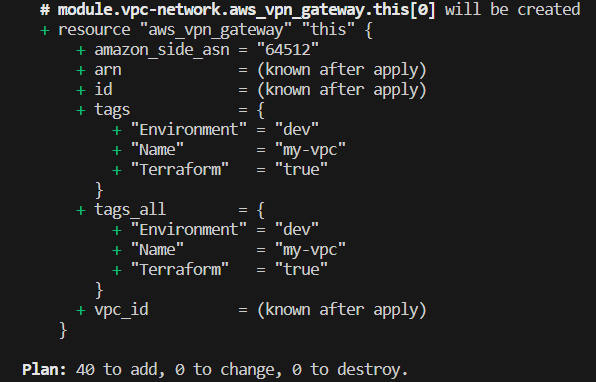
****

**Terraform commands used:**

* **Terraform init**

****

* **Terraform plan**

****

* **Terraform fmt**
* **Terraform validate**
* ****
* **Terraform state list**
* **Terraform apply**
* **Terraform destroy**