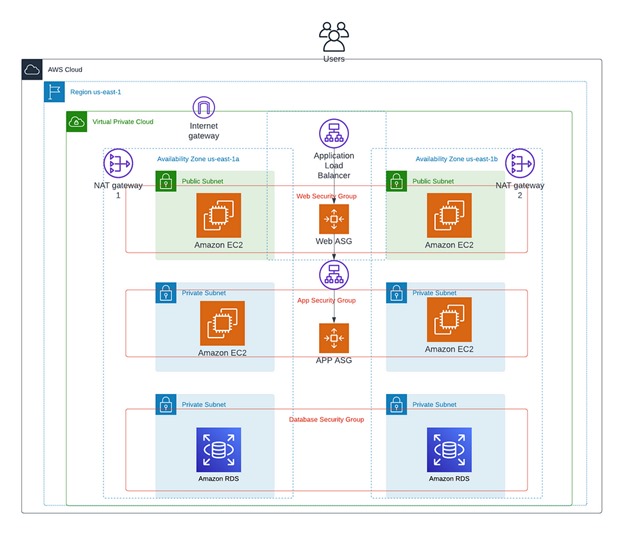
PROJECT-1

AWS 3-TIER ARCHITECTURE

A 3-tier architecture consists of Web tier, App tier and database tier where the three tiers work together to run a website.



In a virtual private cloud(vpc) we must create the entire project. The 3-tier Architecture consists of the following components:

1. A VPC vpc-project (us-east-1)
2. Two public subnets

* public-1a (us-east-1a)
* public-1b (us-east-1b)

1. Four private subnets

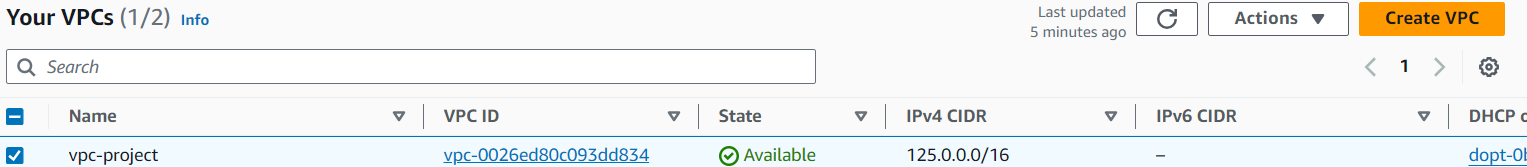
* private-1a (us-east-1a)
* private-1a-db (us-east-1a)
* private-1b (us-east-1b)
* private-1b-db (us-east-1b)

1. Two route tables

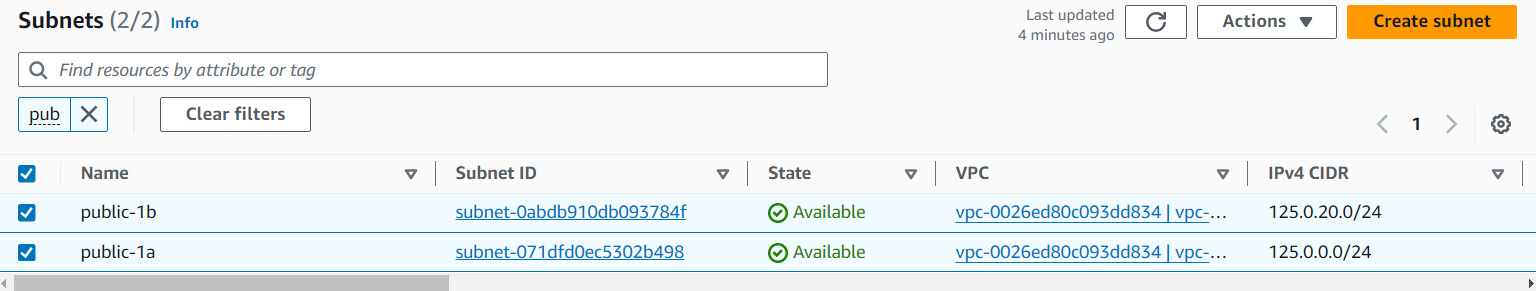
* my-rtb-public
* my-rtb-private

1. One Internet gateway my-igw
2. One NAT gateway my-ngw

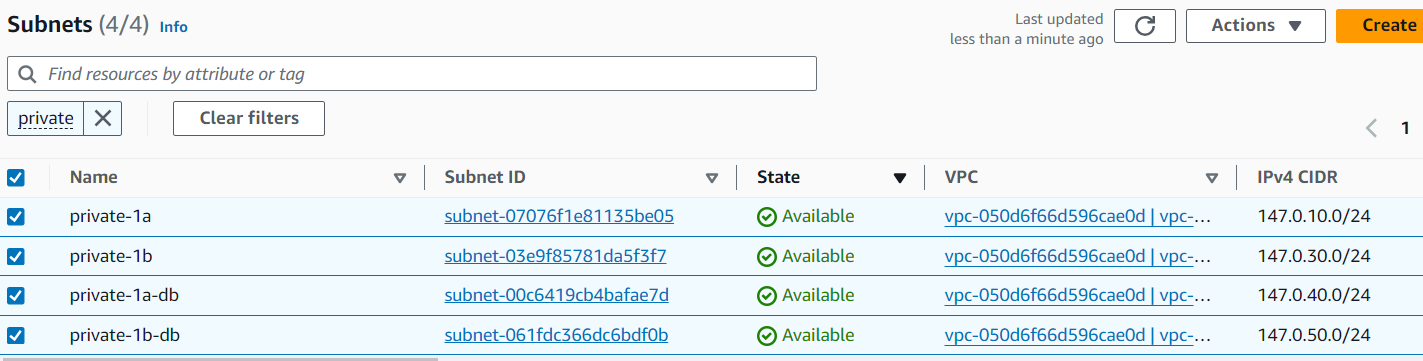
Start with creating a VPC vpc-project with CIDR 125.0.0.0/16 in the region us-east-1.



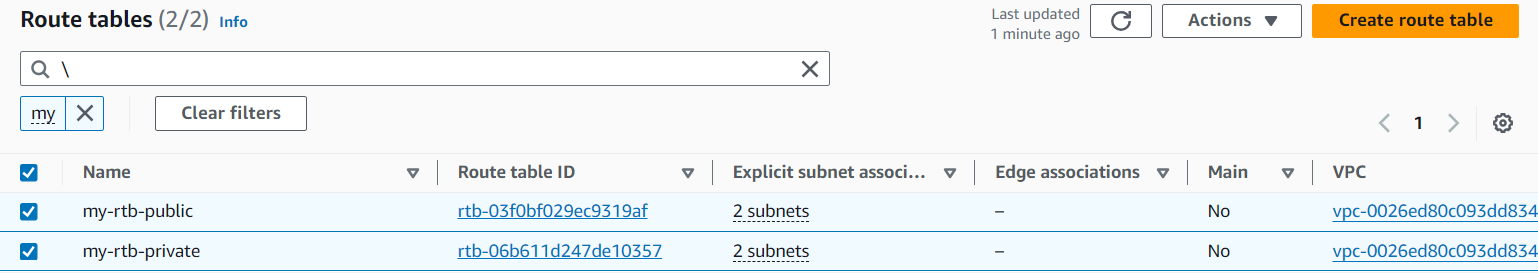
After creating vpc create two public subnets public-1a (us-east-1a) and public-1b (us-east-1b).



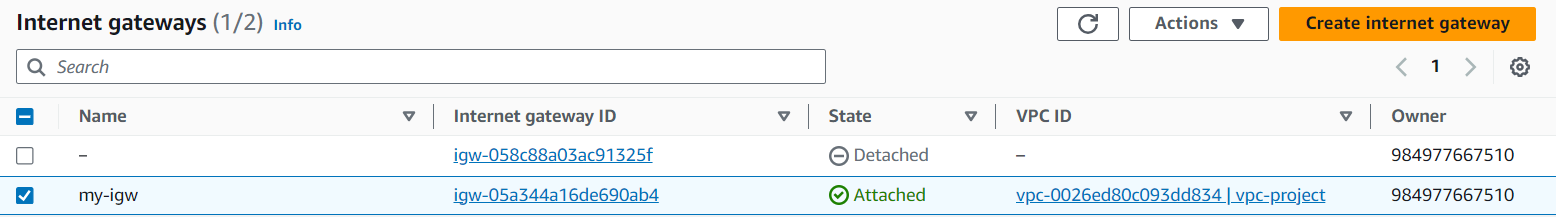
Create four private subnets private-1a and private-1a-db in the region us-east-1a, private-1b and private-1b-db in the region us-east-1b.

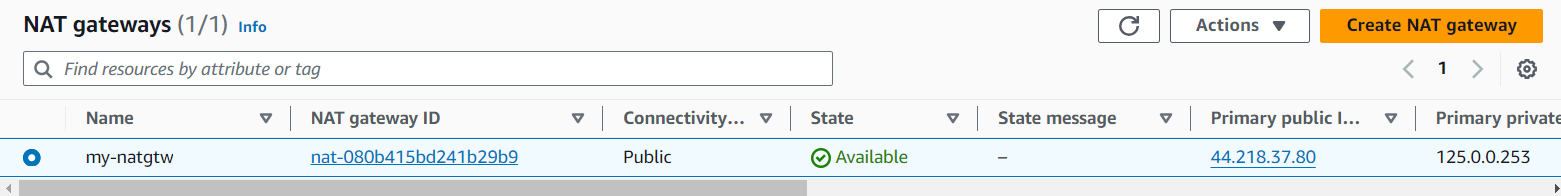


Now create two route tables my-rtb-public and my-rtb-private. Associate public subnets to the public route table and private subnets to private route table.

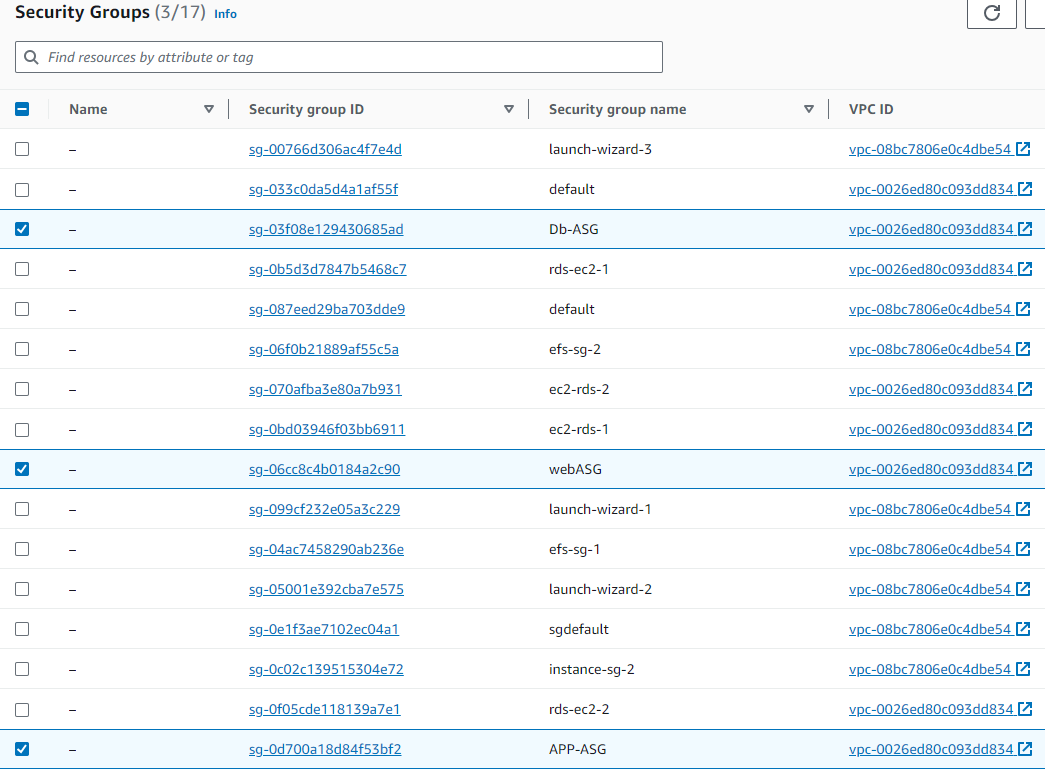


Create internet gateway and NAT gateway. Associate my-igw to public route table and my-ngw to private route table.

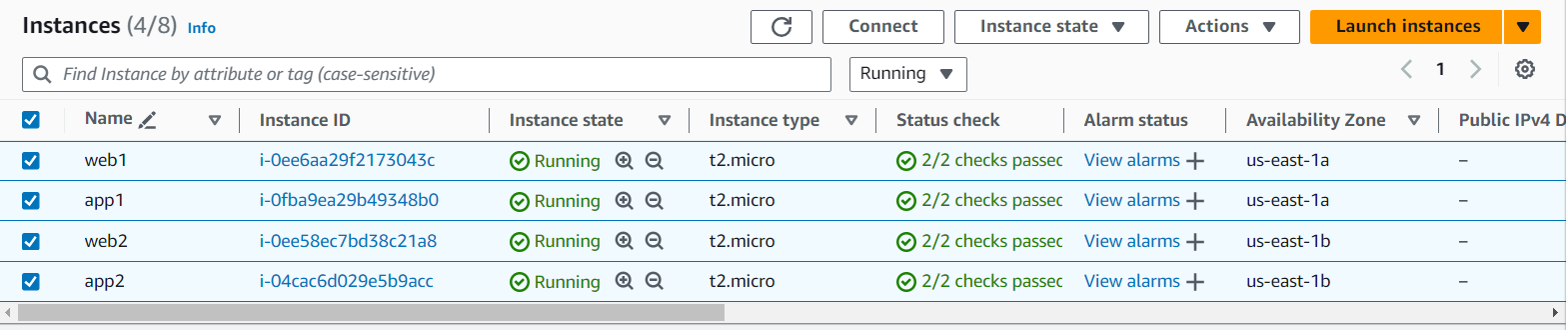




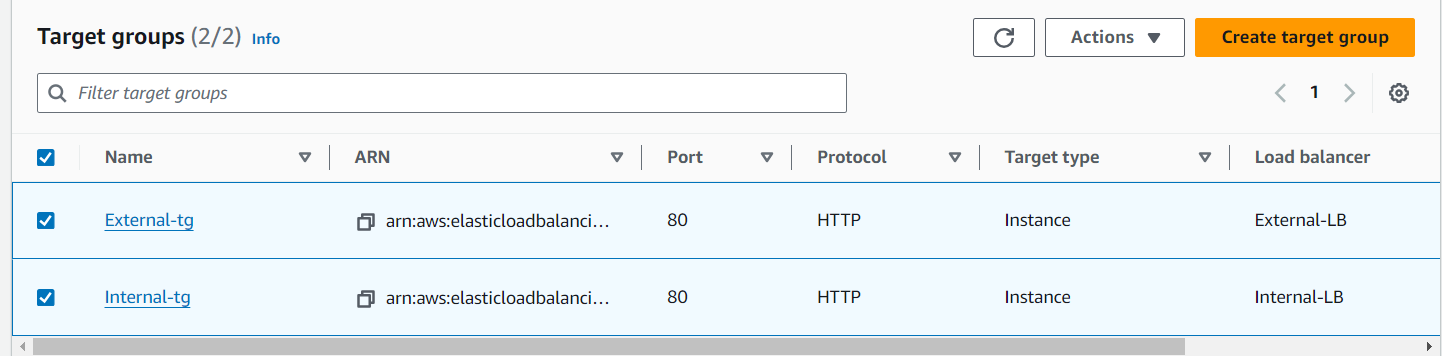
Create three Security groups Web-ASG, App-ASG and Db-ASG.



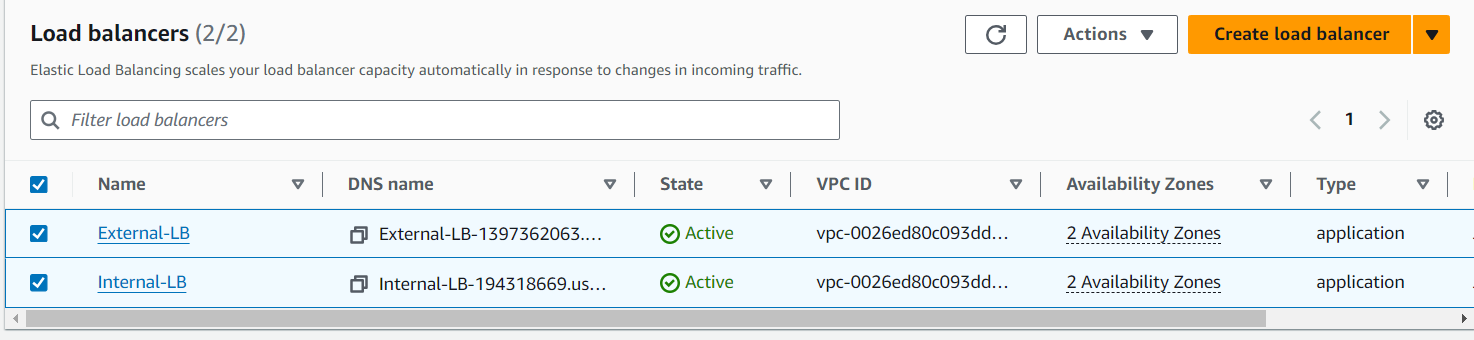
Create four instances Web1 and App1 in us-east-1a, Web2 and App2 in us-east-1b. Where Web1 and Web2 are web tier, App1 and App2 are app tier.



Create 2 target Groups and Create 2 Application Load balancer for External and internal Layer, attach 2 Public Ec2 server mention pending as below.

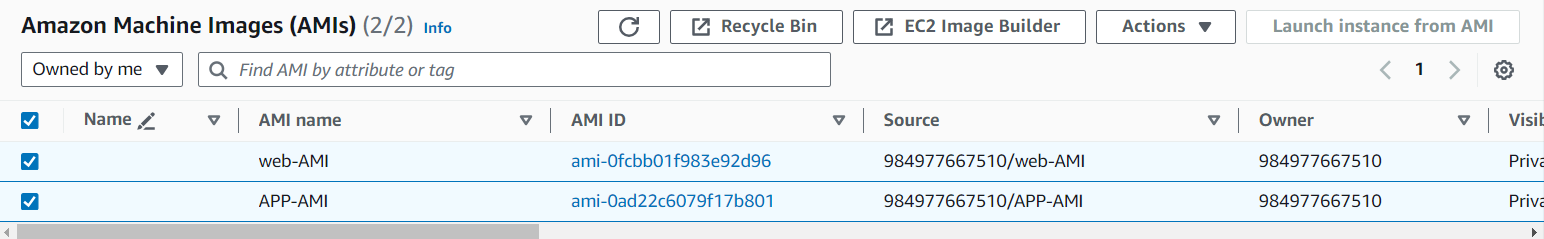


Now attach the target groups to the load balancer.

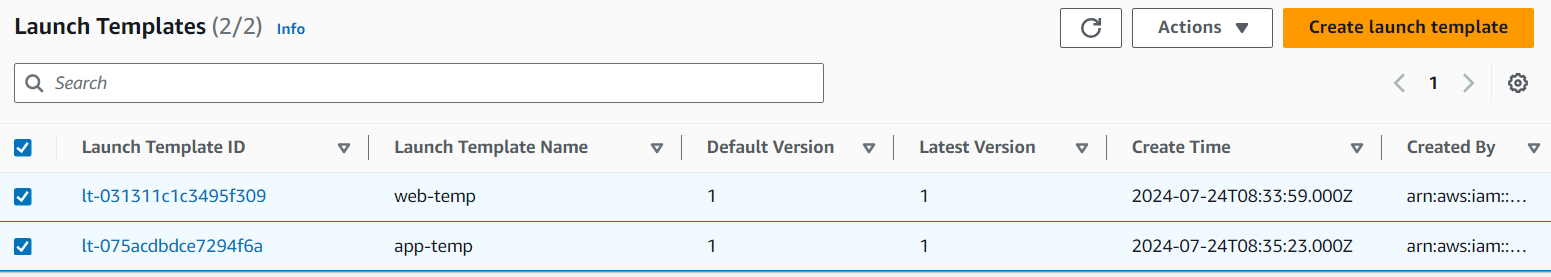


Copy DNS and paste it in the browser, to see load is equally distributed or not, if the load is equally distributed, then when we refresh the browser, we see pieces of information of index.html of nginx in each server. 

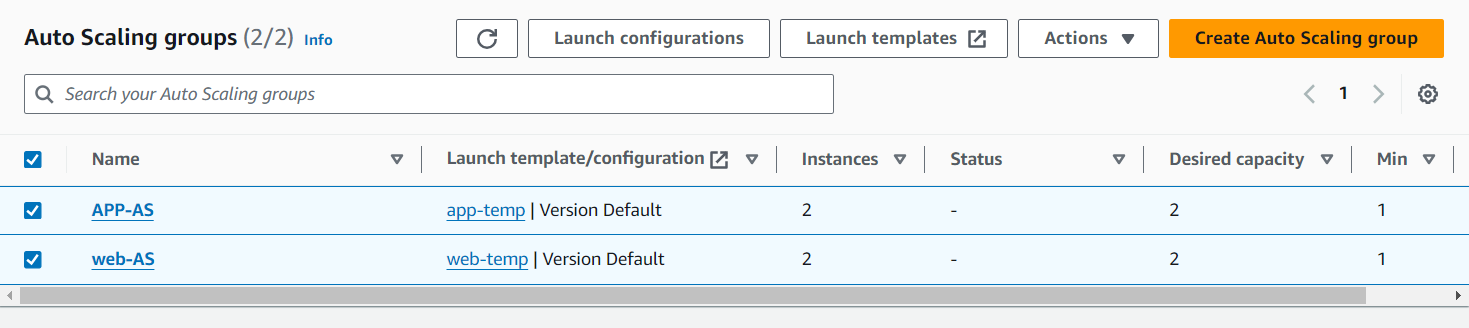
Create an Amazon Machine Image (AMI) by selecting an instance from each of Web-tier and App-tier.



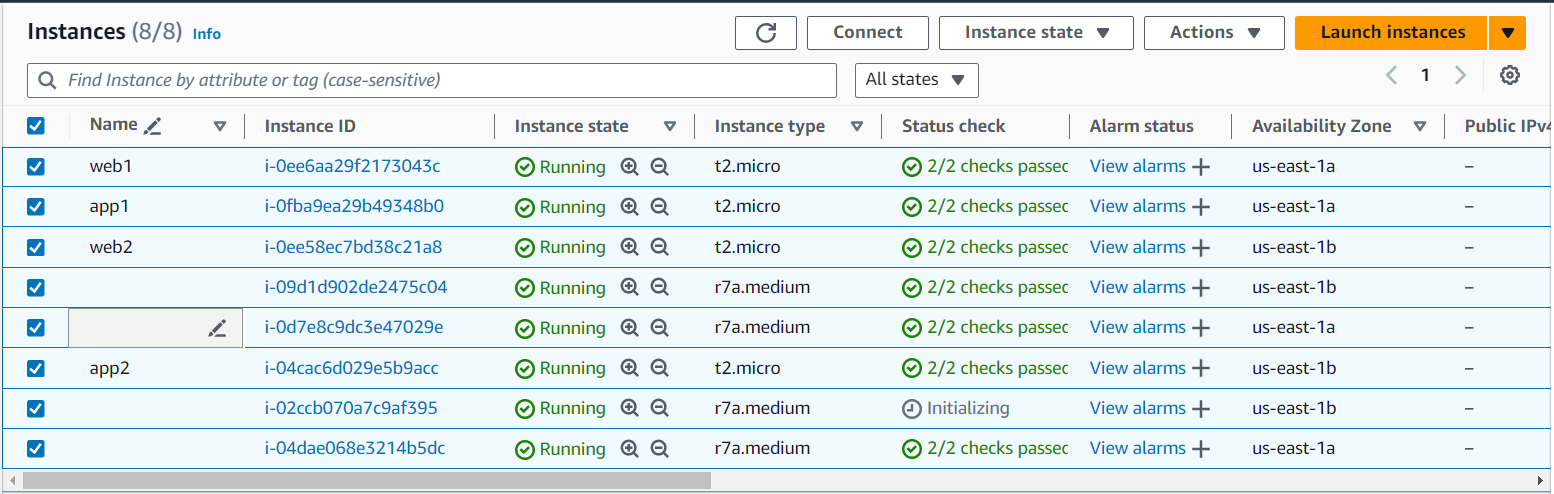
Now create launch template for web and app tier.



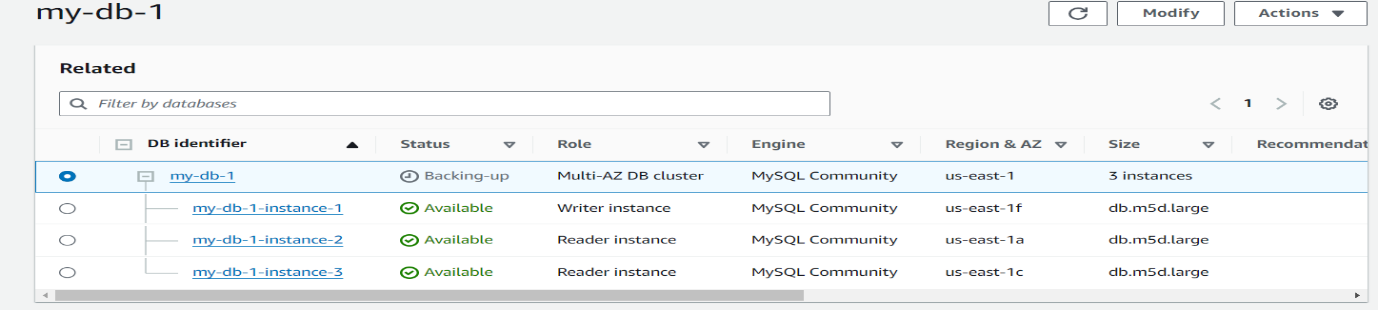
Now create Auto scaling groups for Web and App tier.



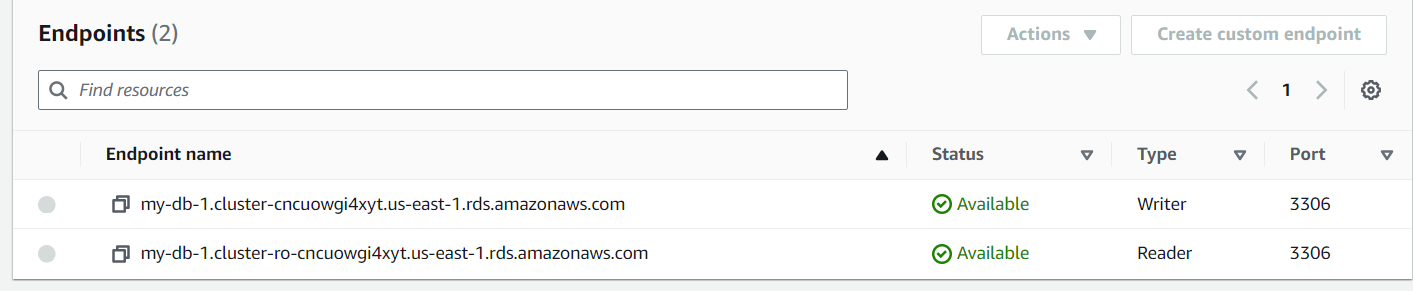
We can clearly see that the minimum number of instances are created automatically through Auto scaling in desired Availability zones.



Create RDS and attach Private EC2 (App-1) to RDS Database.

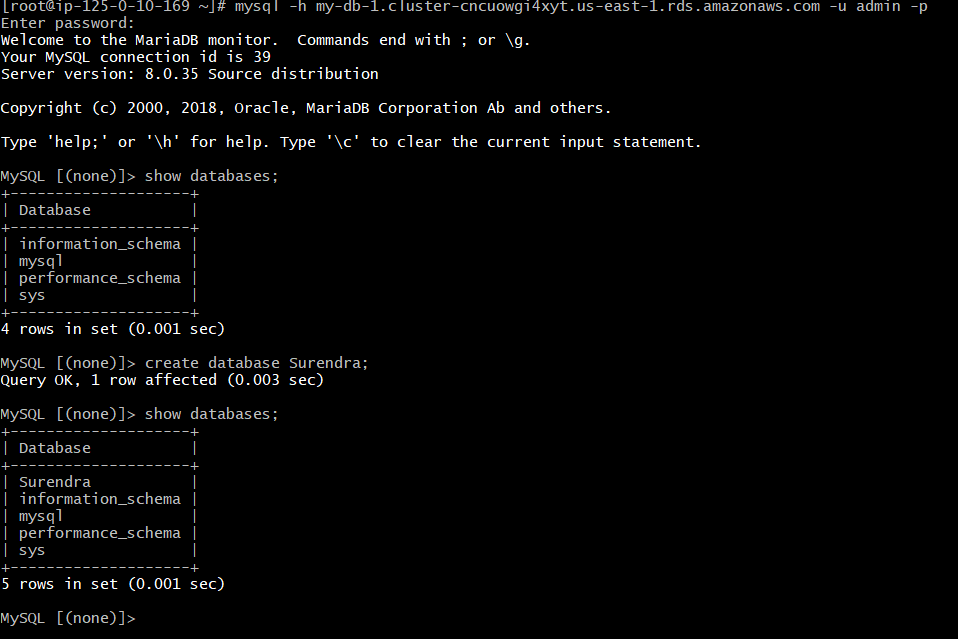


These are the Endpoints.



Login to Public EC2 server connect from public server to private server and install MySQL in private server.

Copy endpoint URL and login into database in private server.

Database is successfully created.