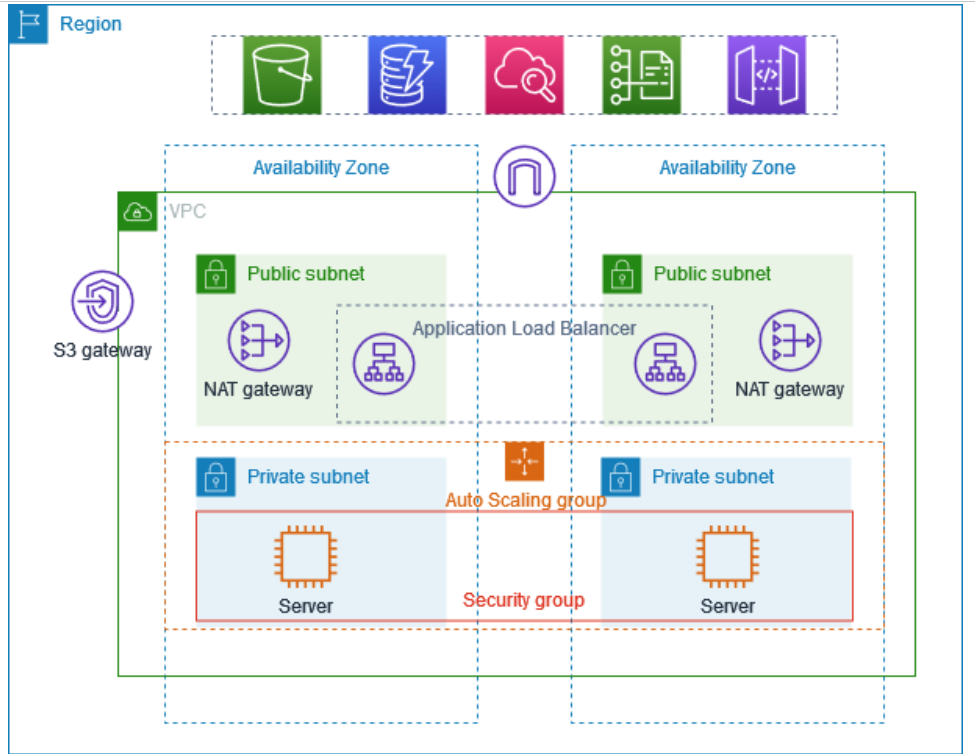
VPC WITH PUBLIC PRIVATE SUBNET IN PRODUCTION

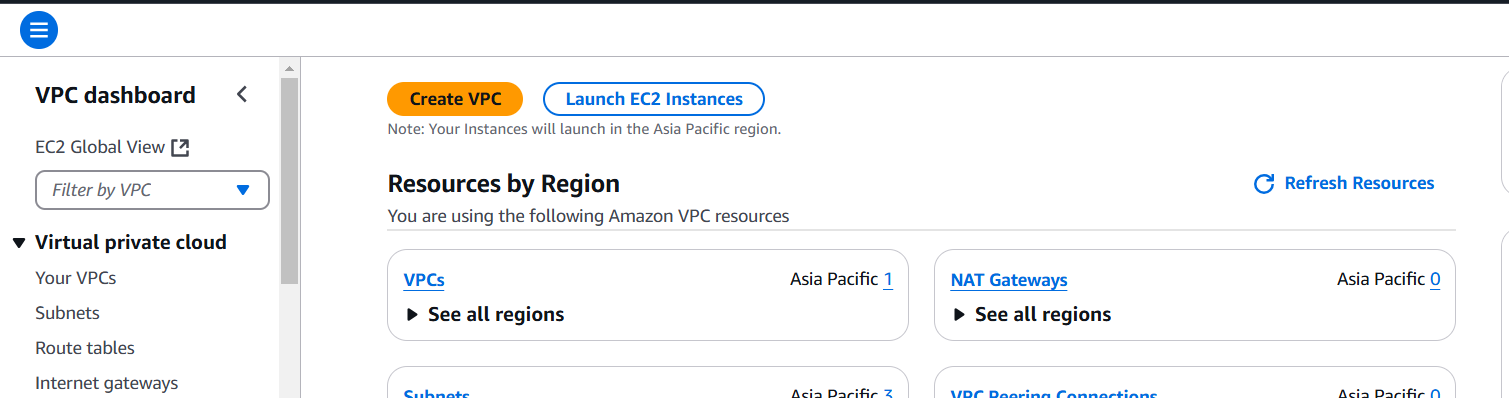
This example demonstrates how to create a VPC that you can use for servers in a production environment. To improve resiliency, you deploy the servers in two Availability Zones, by using an Auto Scaling group and an Application Load Balancer. For additional security, you deploy the servers in private subnets. The servers receive requests through the load balancer. The servers can connect to the internet by using a NAT gateway. To improve resiliency, you deploy the NAT gateway in both Availability Zones.

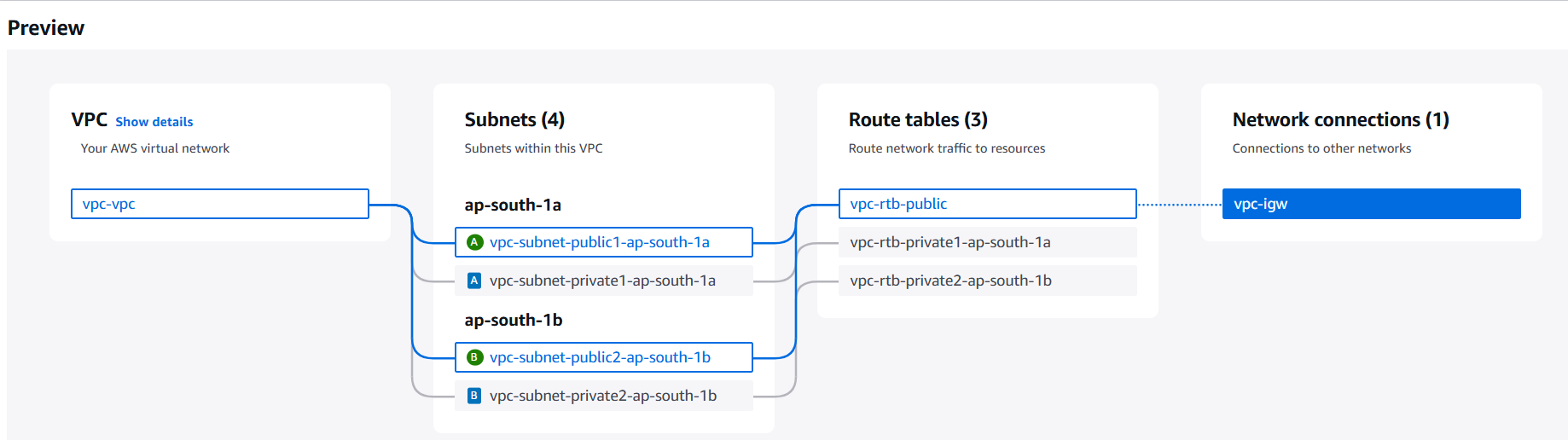
The following diagram provides an overview of the resources included in this example. The VPC has public subnets and private subnets in two Availability Zones. Each public subnet contains a NAT gateway and a load balancer. The servers run in the private subnets, are launched and terminated by using an Auto Scaling group, and receive traffic from the load balancer. The servers can connect to the internet by using the NAT gateway.



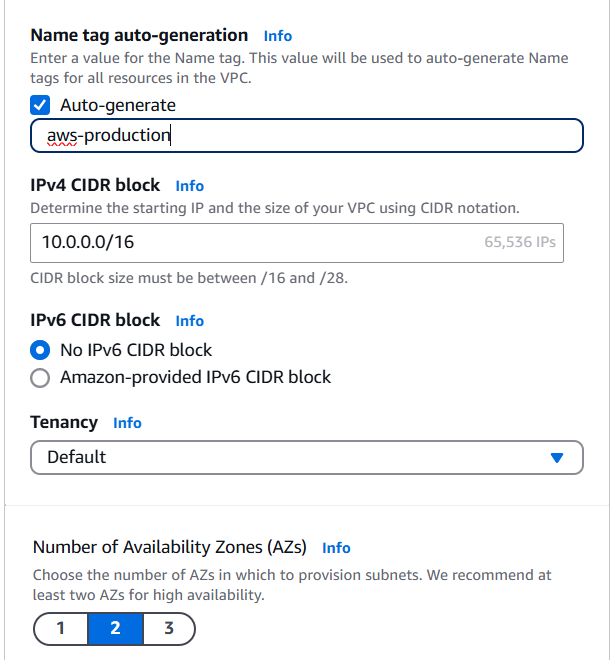
>Goto vpc > create vpc

>Create vpc and more



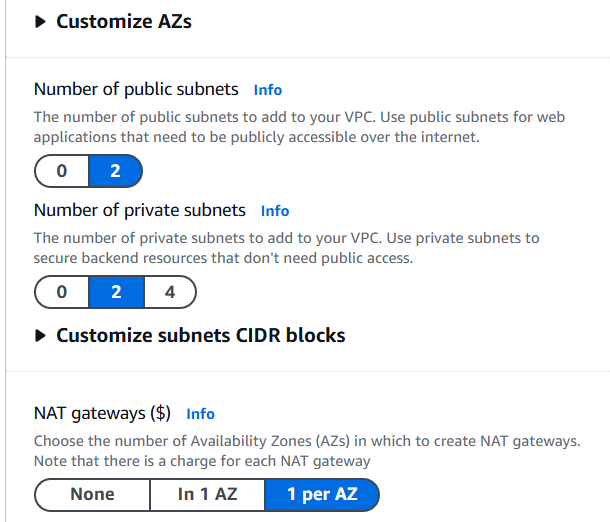


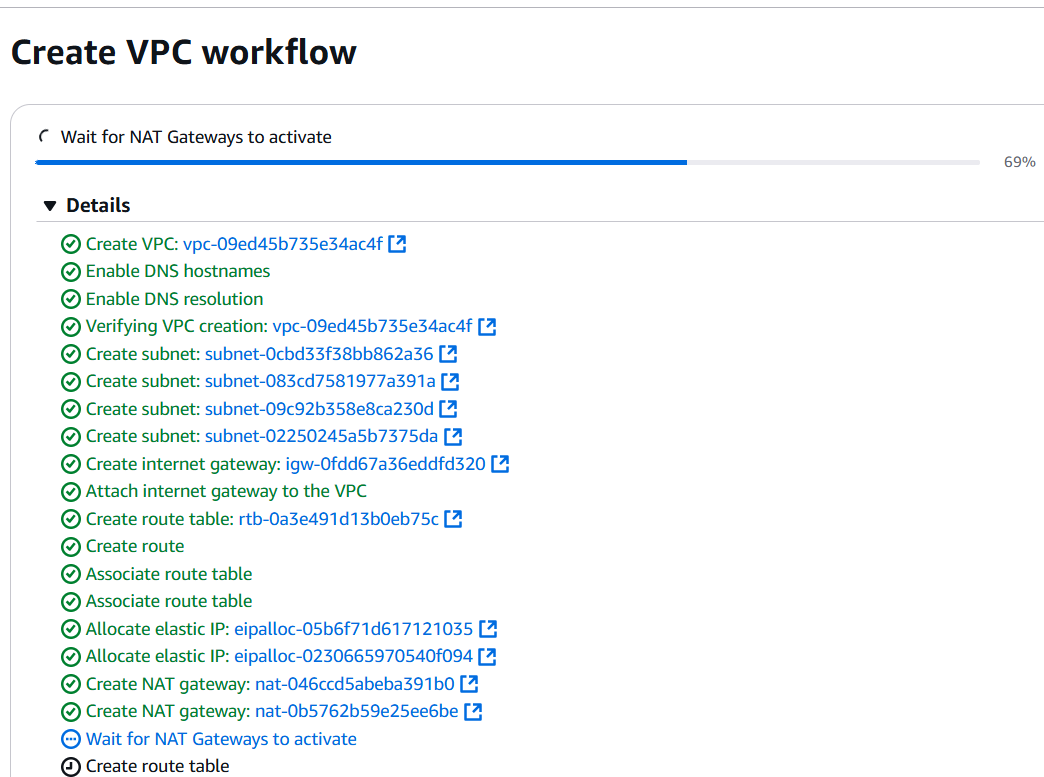
>name > number of AZ >2



>Number of public subnets > 2 > Number of private subnets > 2

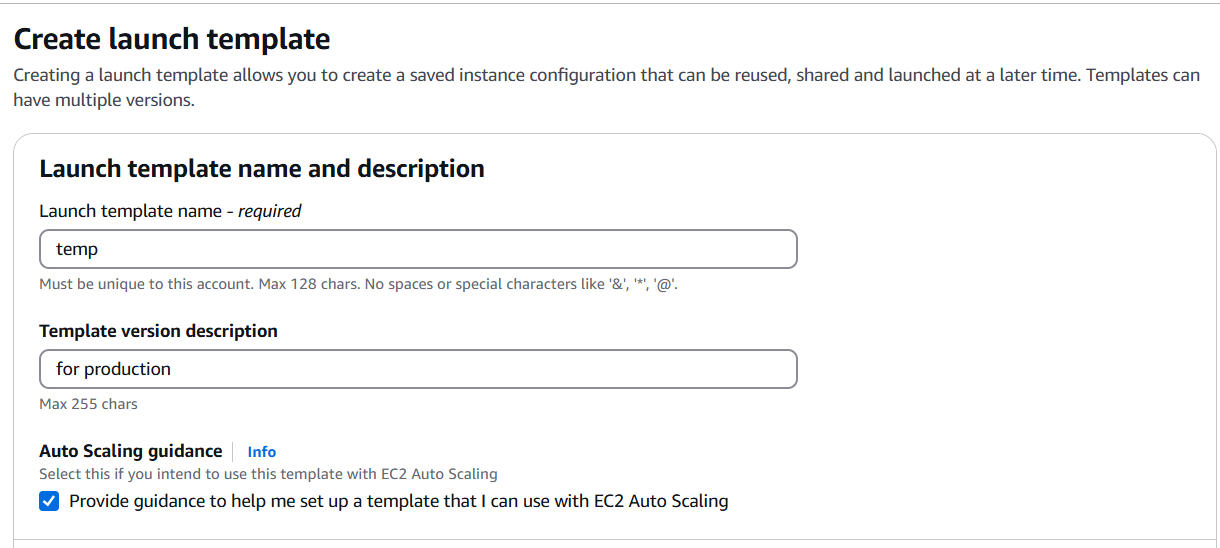
> Number of NAT Gateway > 1 per AZ

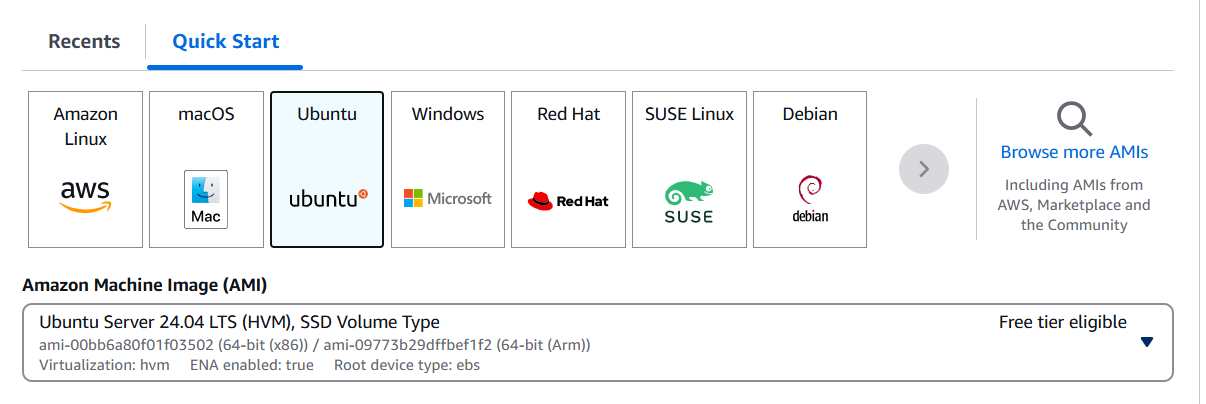
>Create

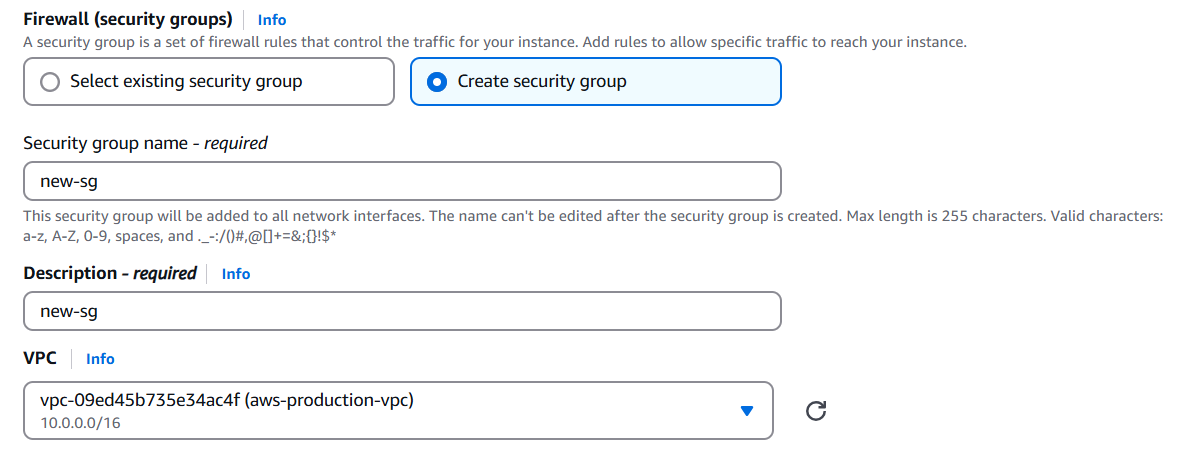


You can see all components are create automatically.

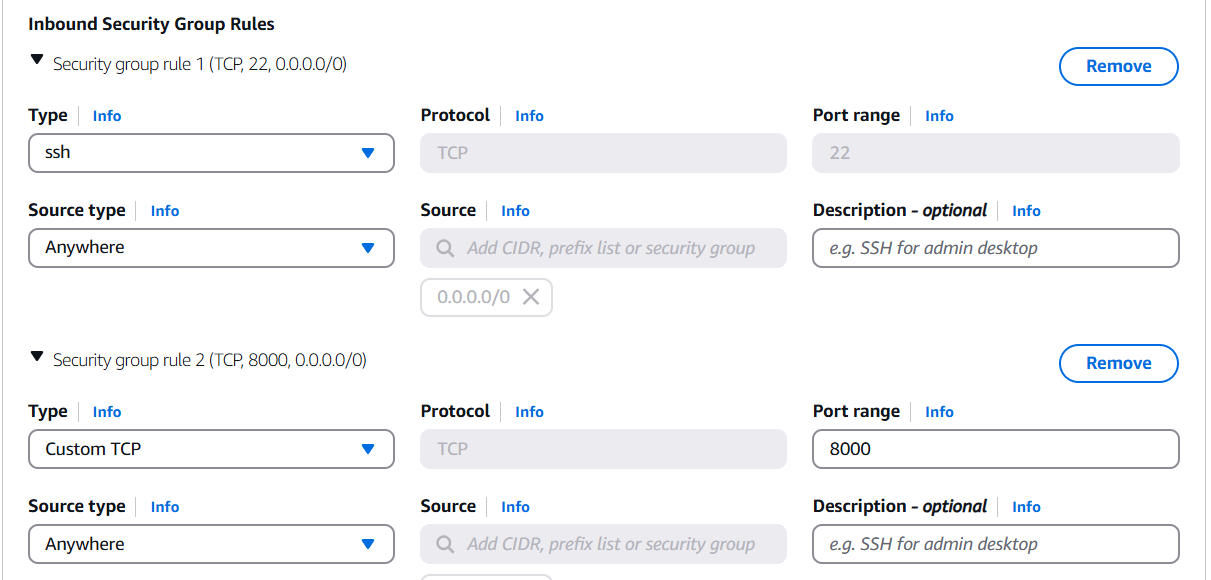
>Create template

>select ubuntu image

>Create new security group in production vpc

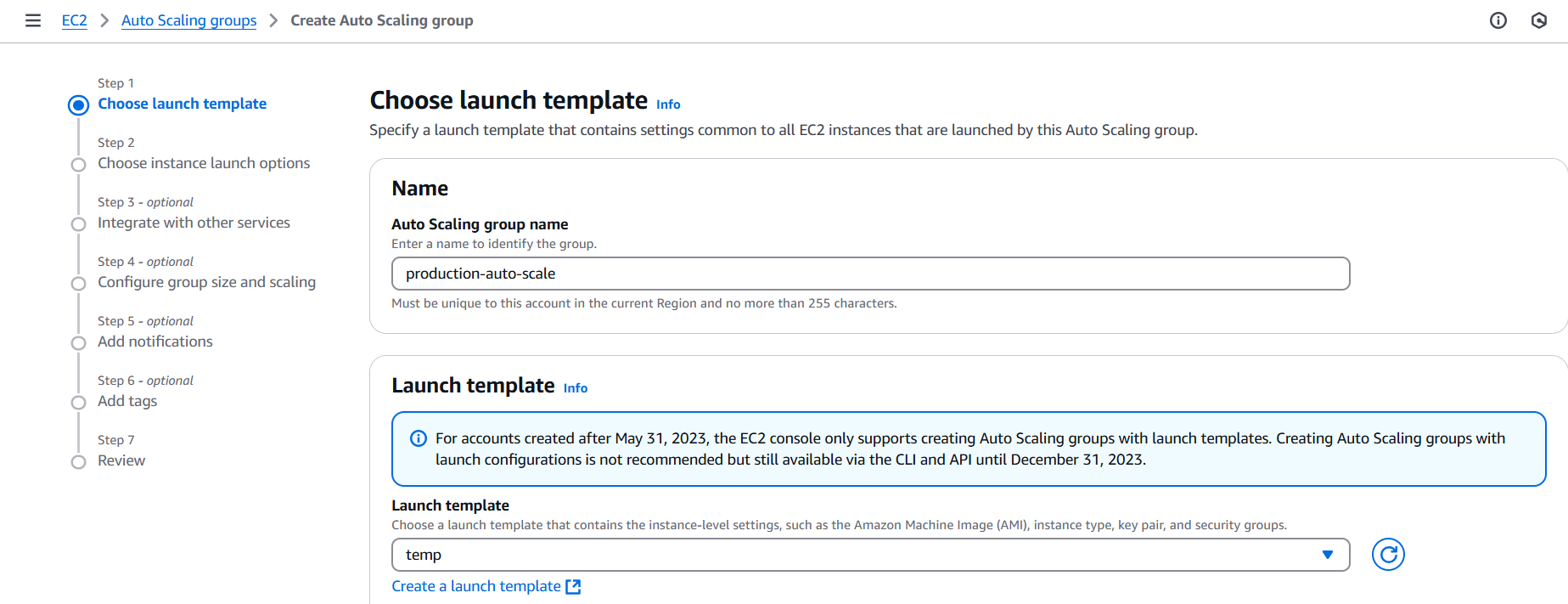


>in security group allow SSH and allow 8000 port

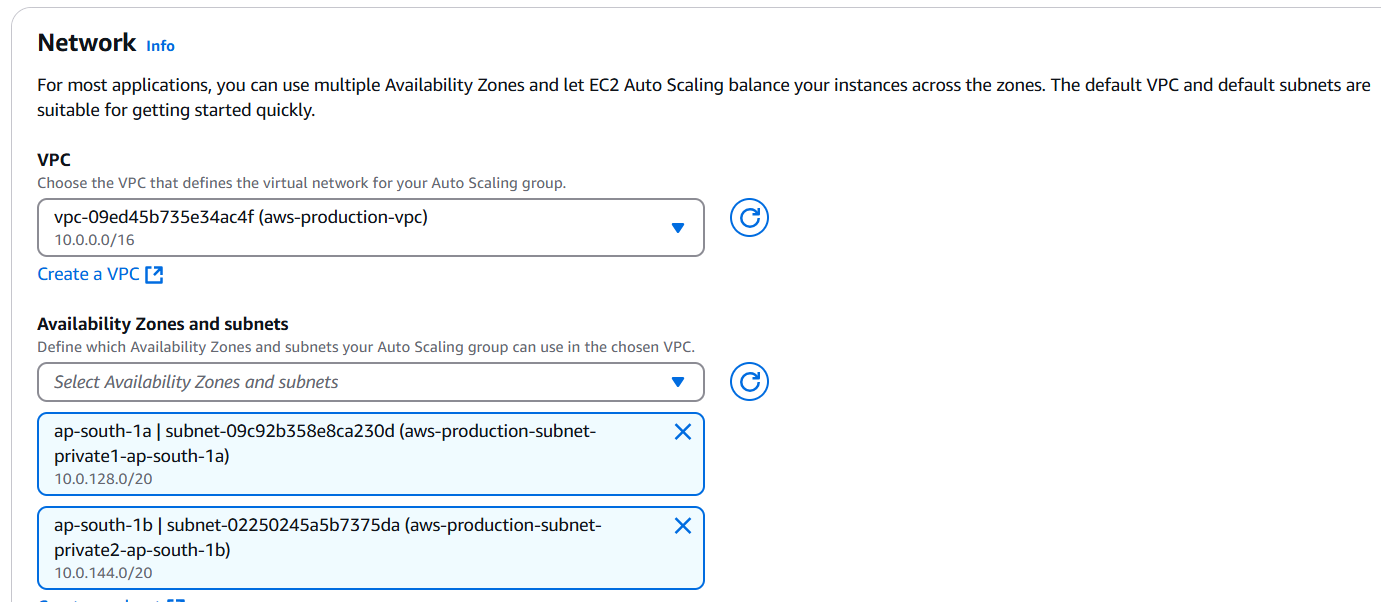


>done

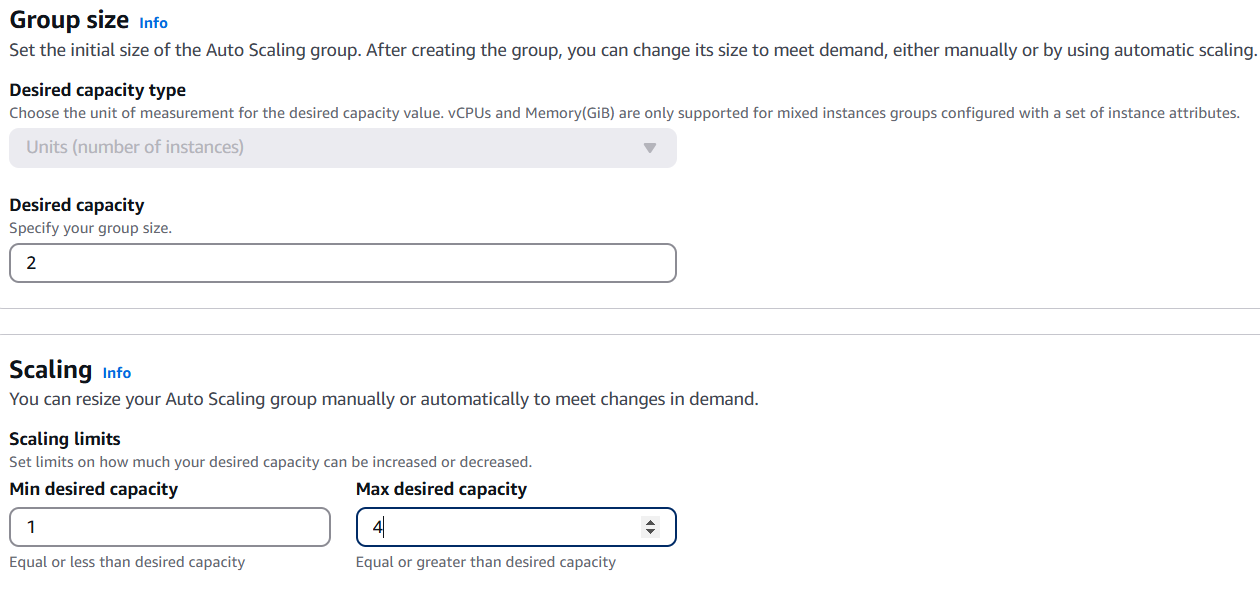
>Create Auto-scaling



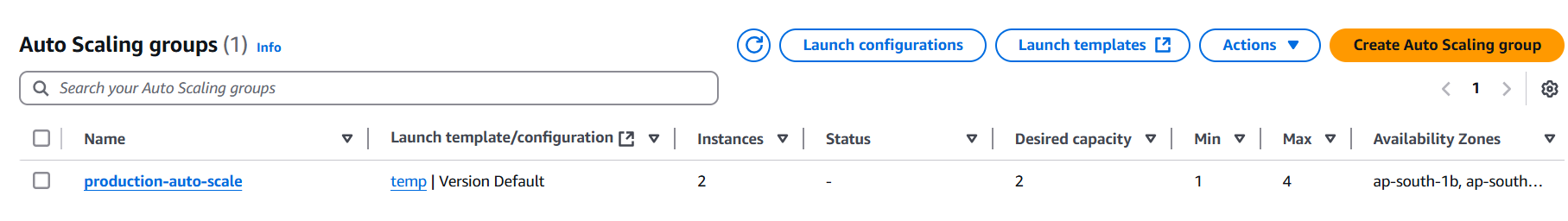
>name > select temp



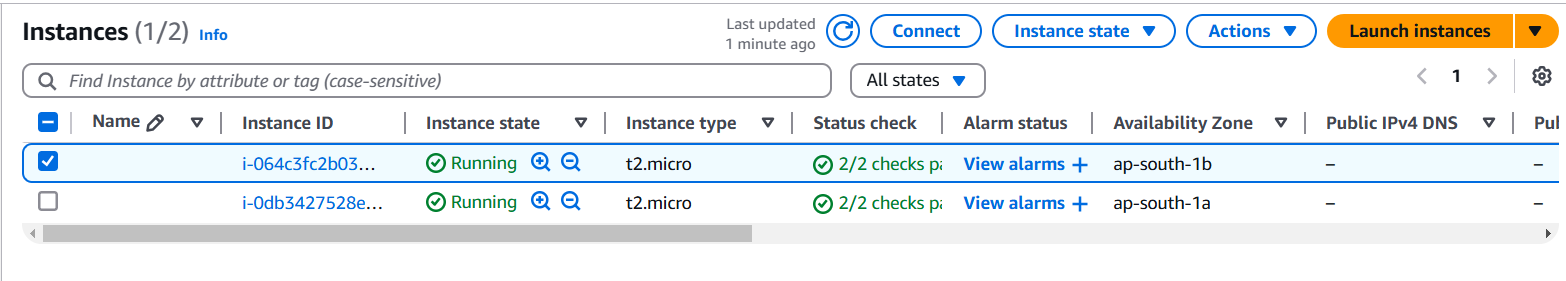
>select production vpc >select 2 private subnet

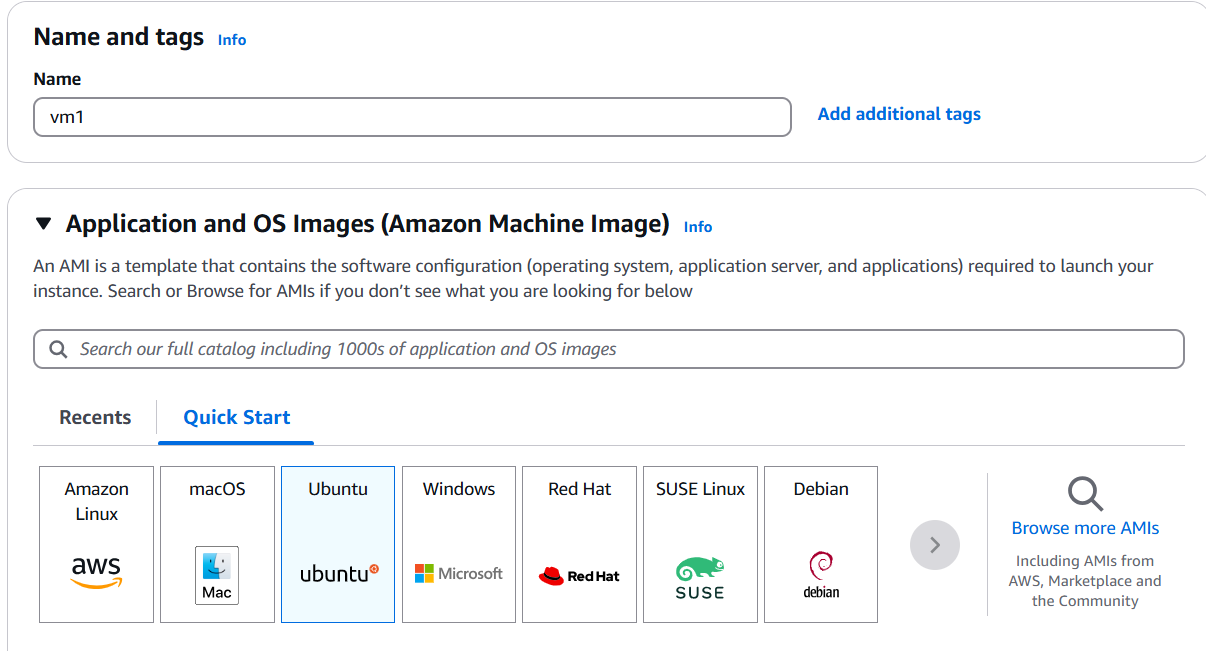


>desired capacity=2 >min desired capacity=1 >max desired capacity=4

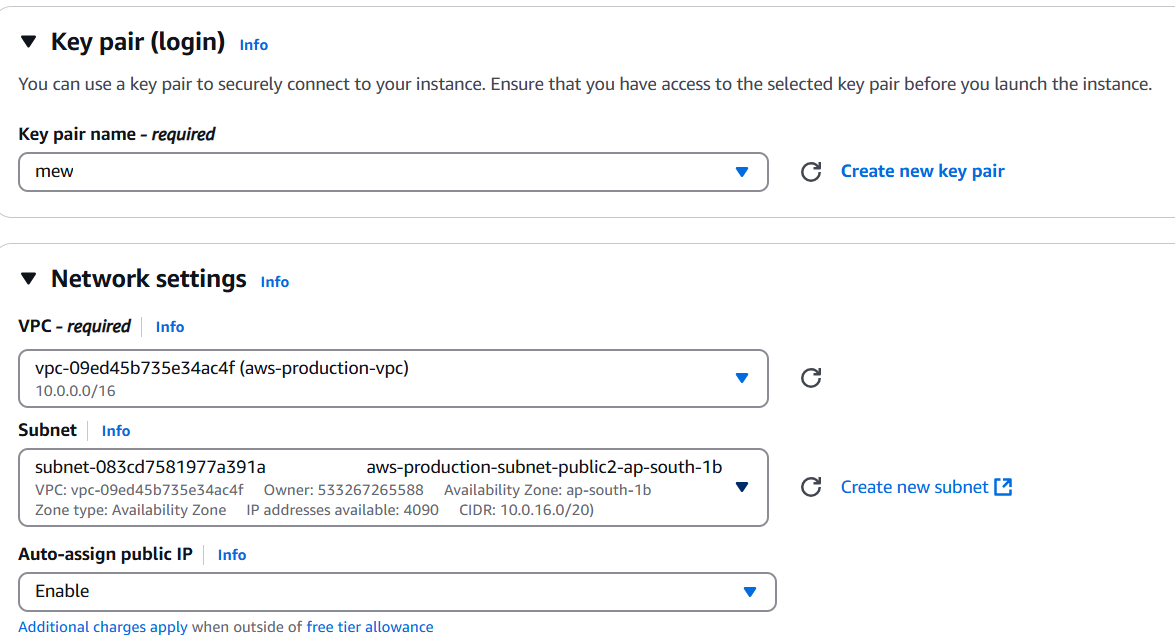


You can see auto scaling group are create

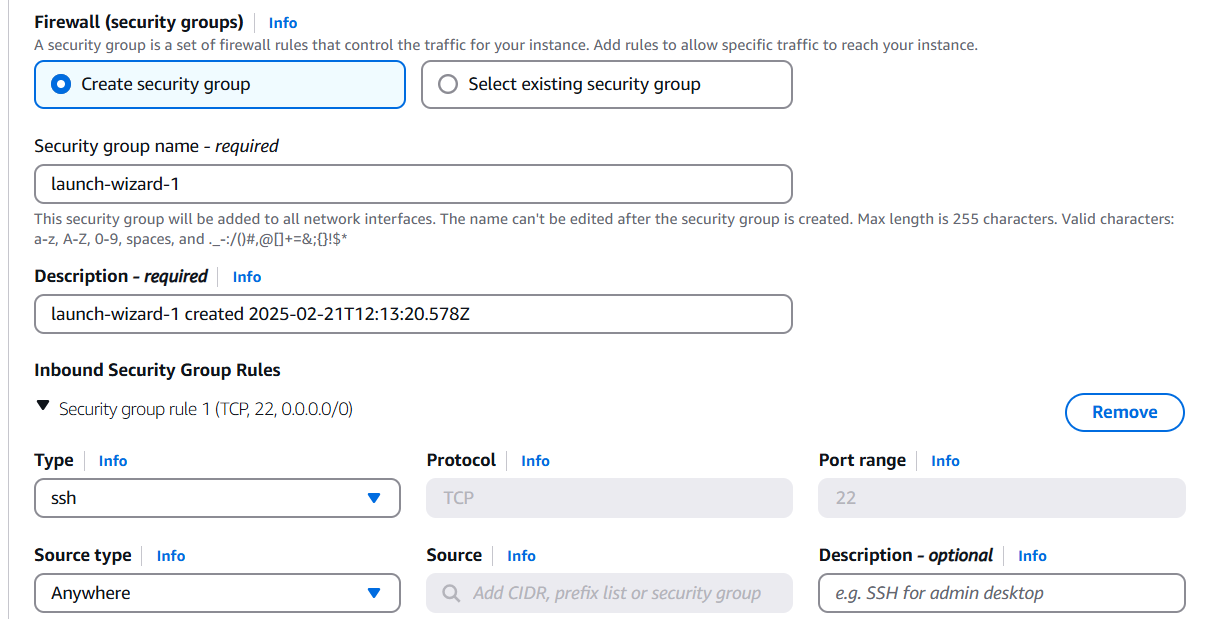
You can see 2 instance are deploy automatically.

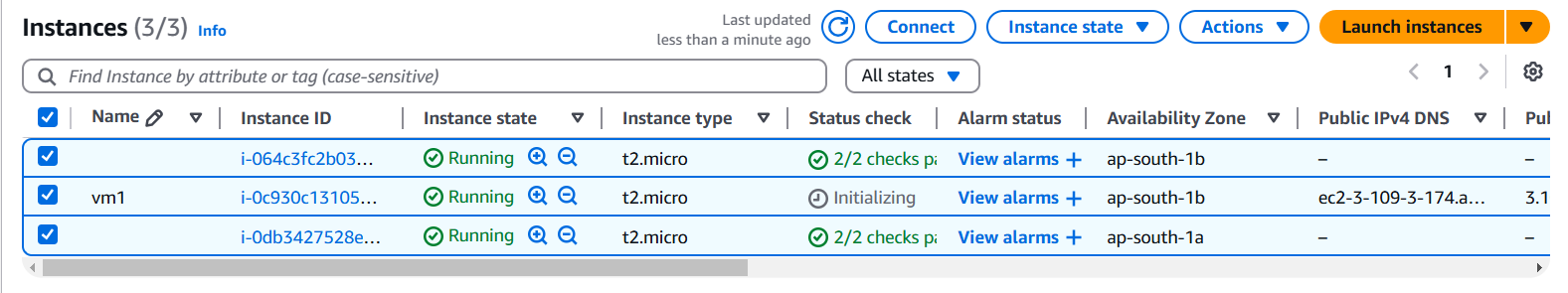


>Also create 1 instance for jump server



>With same production Vpc with public subnet and enable public Ip

>create security group with SSH rule

you can see new instances are created

>scp -i /users\downloads\abc.pem c:\ users\downloads\abc.pem [ubuntu@1.24.23.5](mailto:ubuntu@1.24.23.5): /home/ubuntu



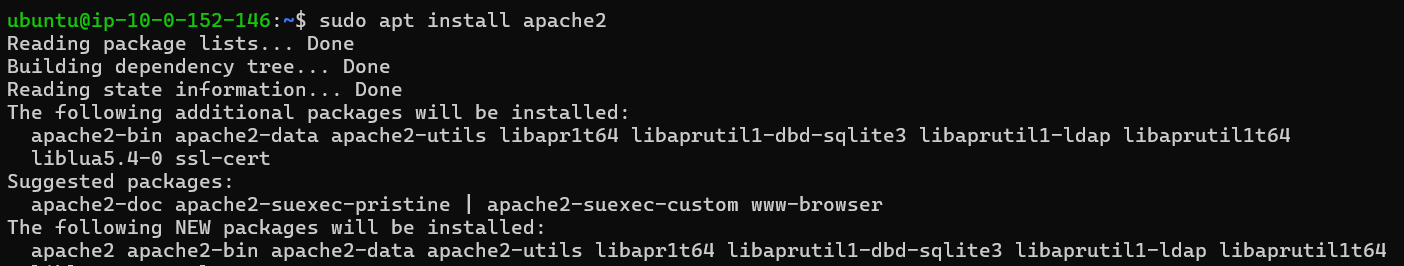
Copy in ubuntu machine



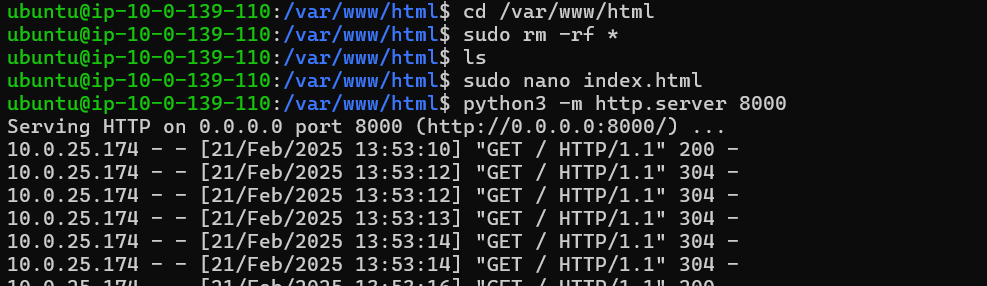
>ssh -i abc.pem ubuntu@1.24.23.5



you can see you access the machine



>sudo apt install apache2



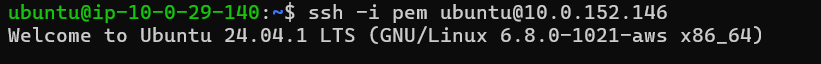
>cd /var/www/html/

>sudo rm -rf \*

>sudo nano index.html link - [krishna-20802](https://github.com/krishna-20802)/[-VPC-with-Public-Private-Subnet-in-Production](https://github.com/krishna-20802/-VPC-with-Public-Private-Subnet-in-Production)

>python3 -m http.server 8000

>get ssh in another instance

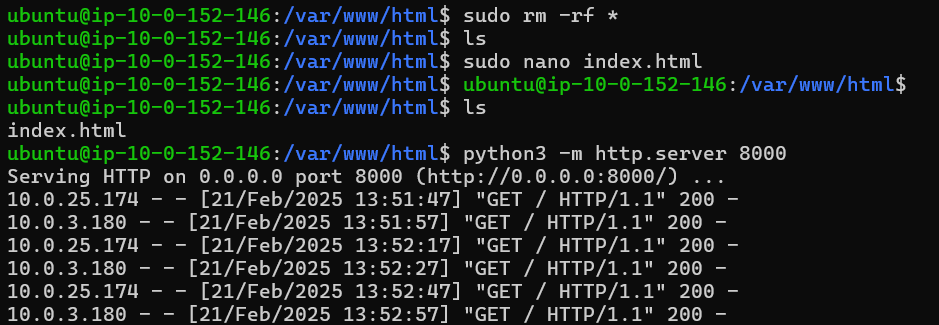


>cd /var/www/html/

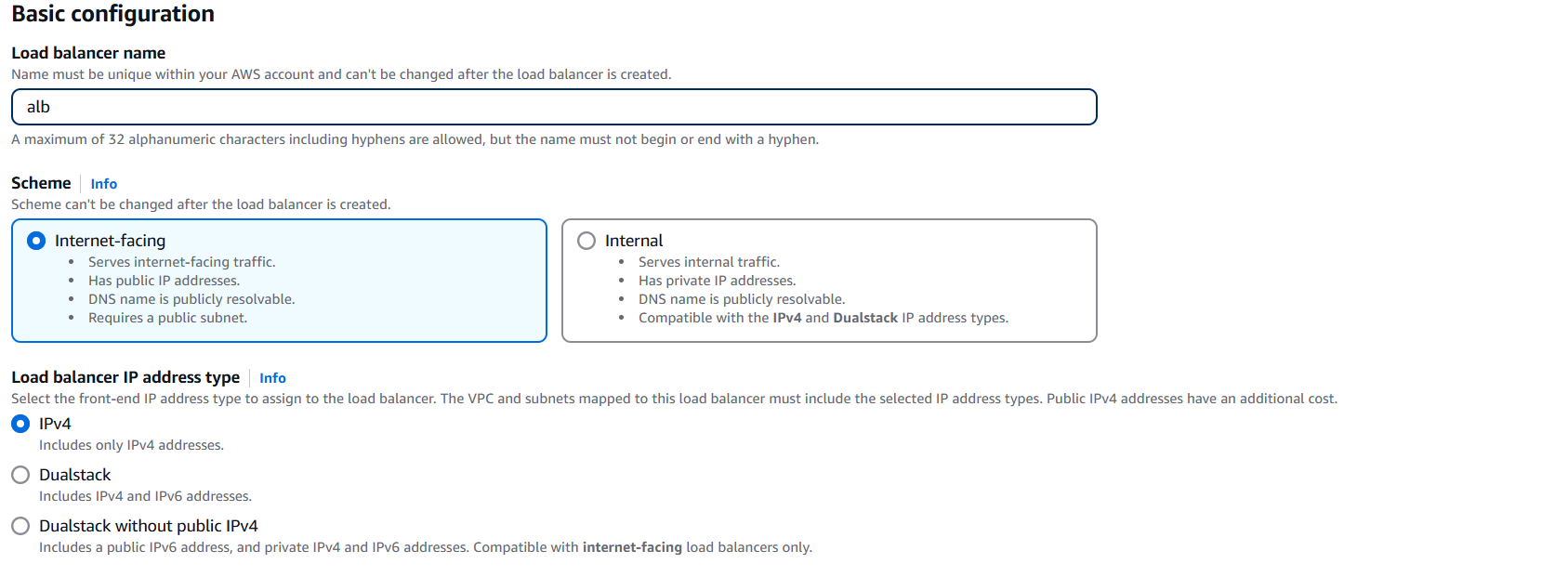
>sudo rm -rf \*

>sudo nano index.html

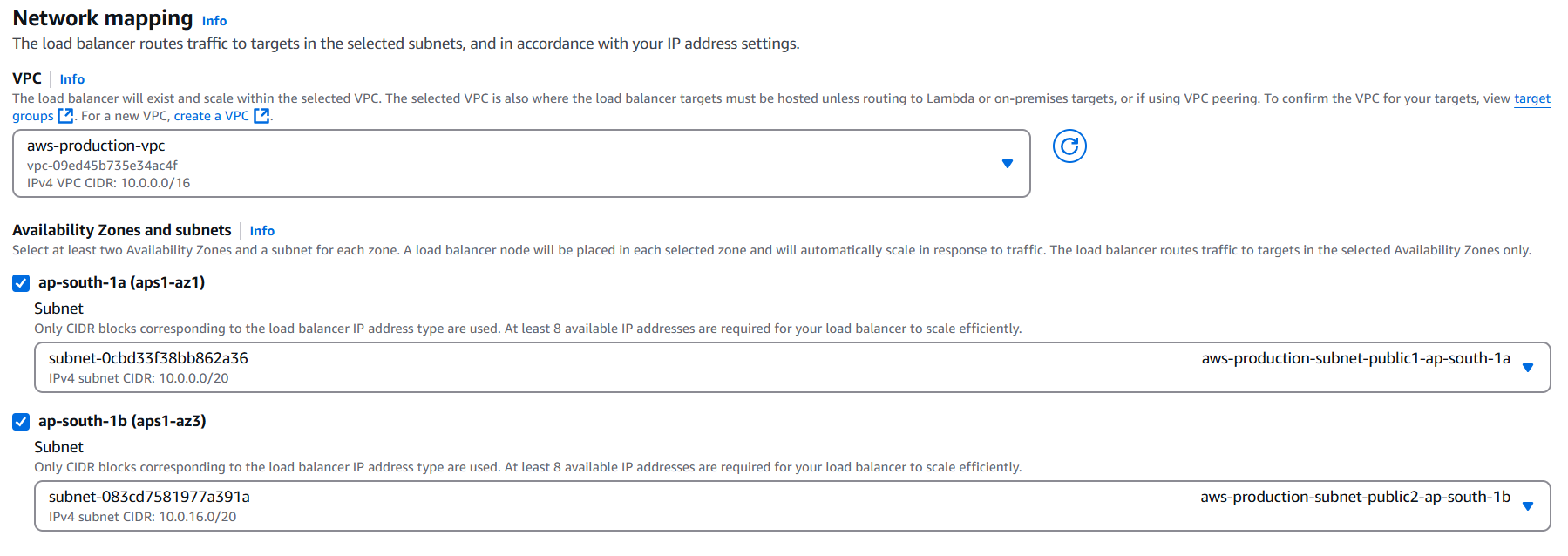
>python3 -m http.server 8000



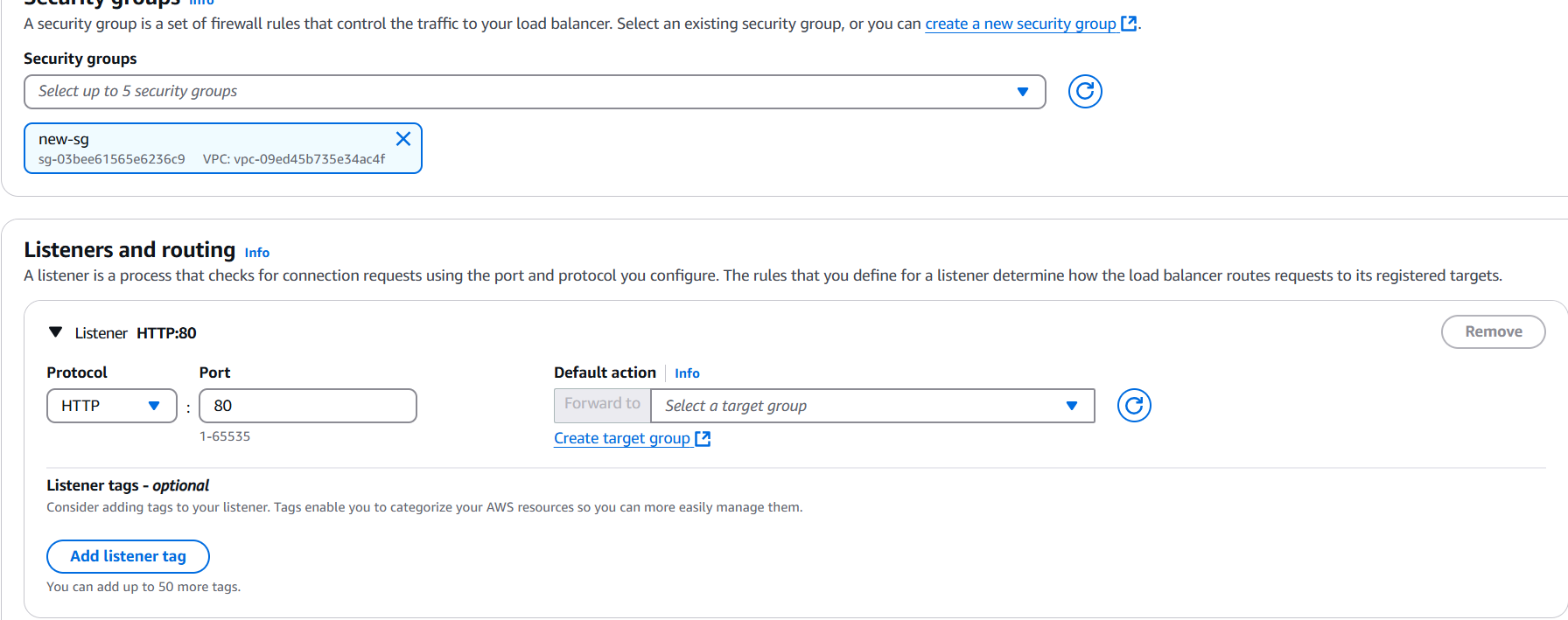
>Create Application Load Balancer with internet facing



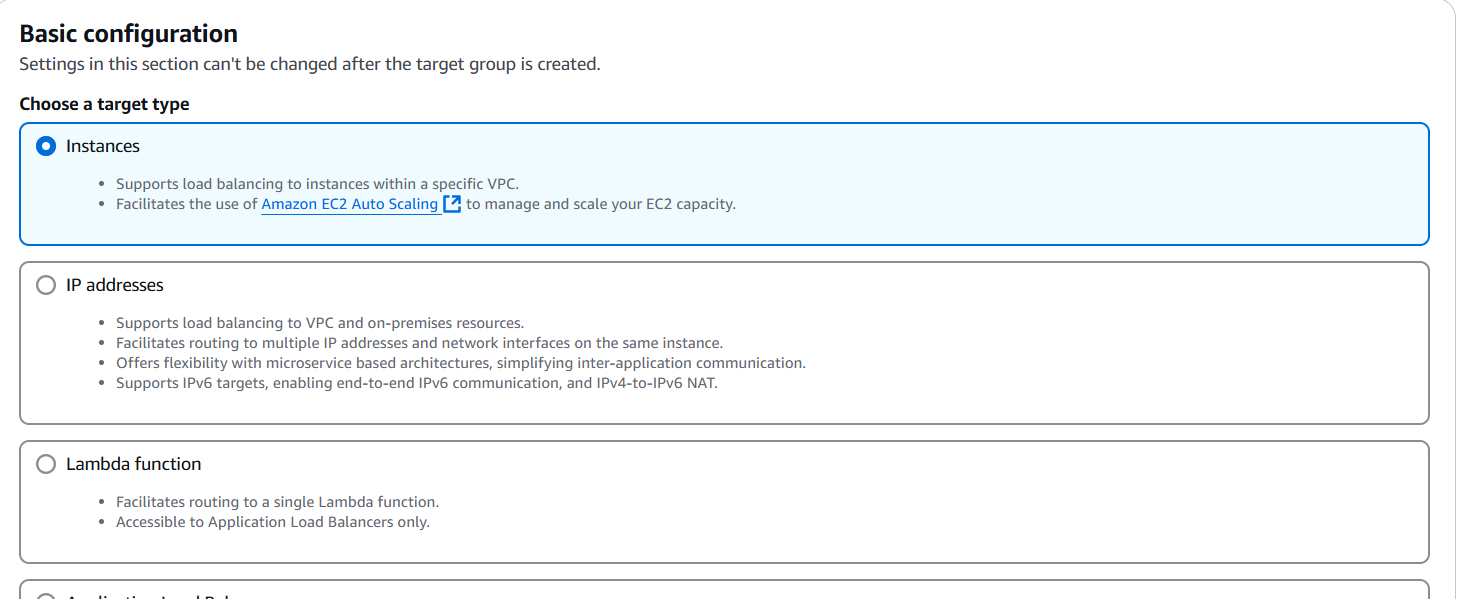
>select production vpc and public subnet

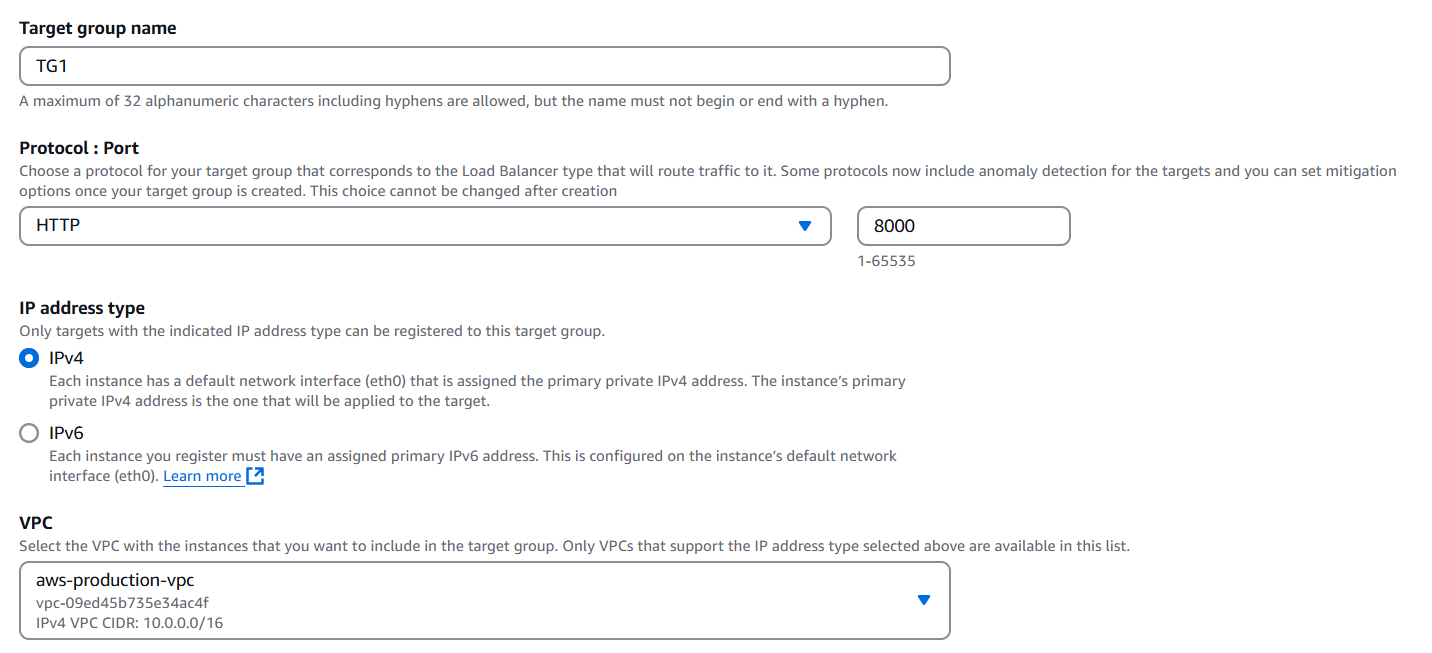


>Select Security group

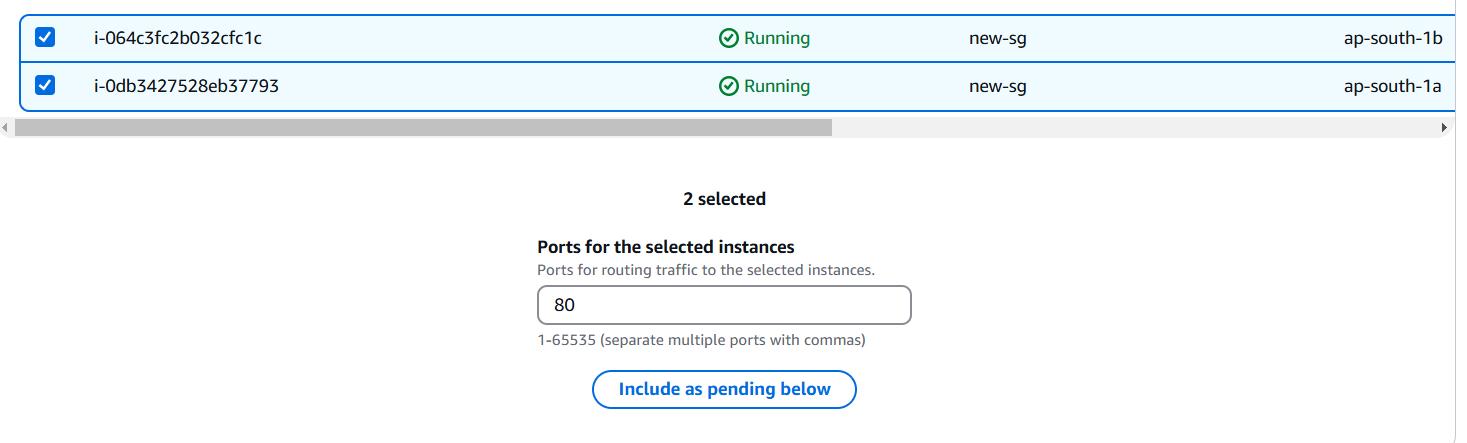


>create target group with port 8000





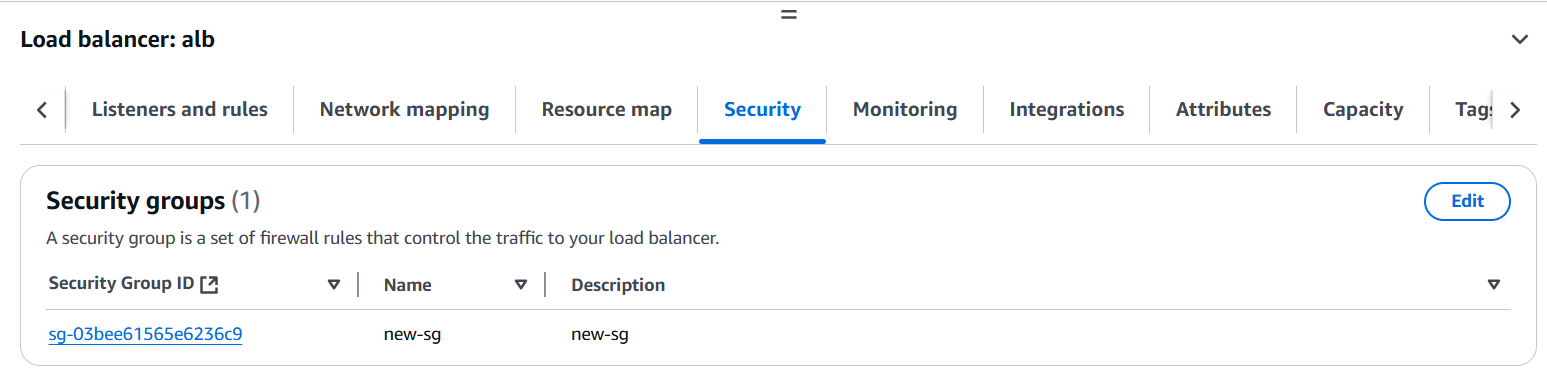
>select 2 and include as pending below

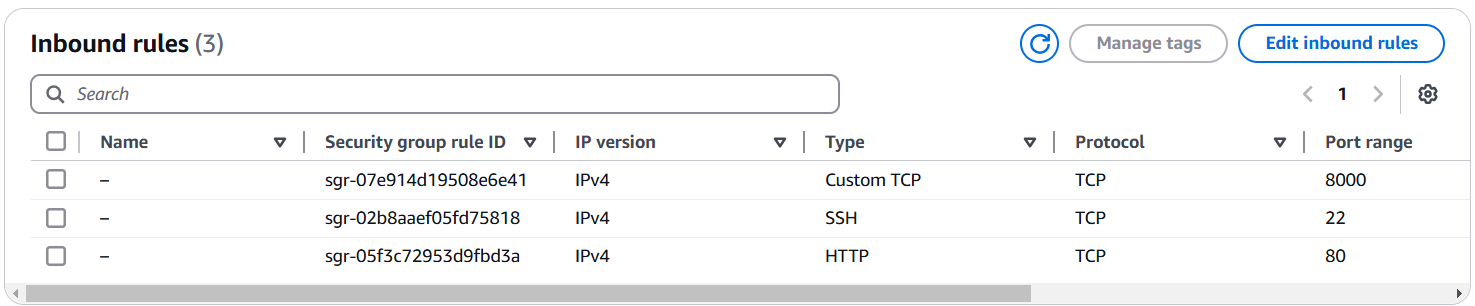


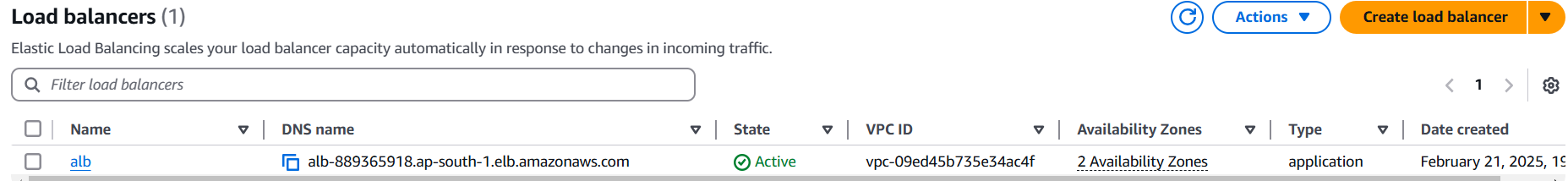
>copy the alb dns and paste it on chrome

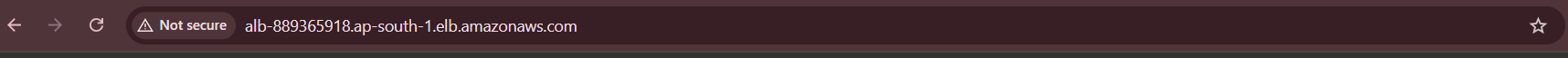
You don’t see any website

>go to alb security click on that

>add inbound rule 8000 port

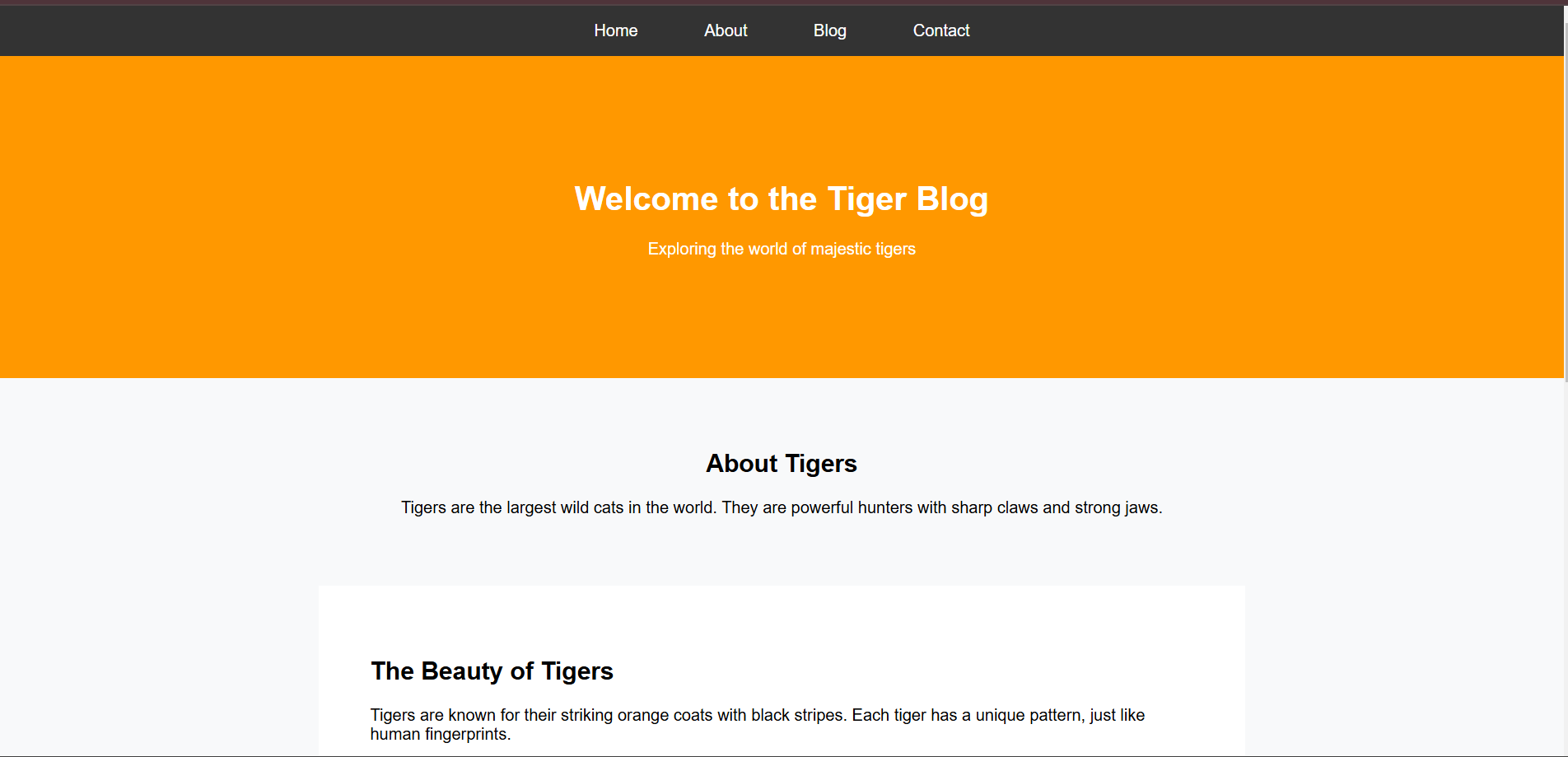
>copy the alb DNS



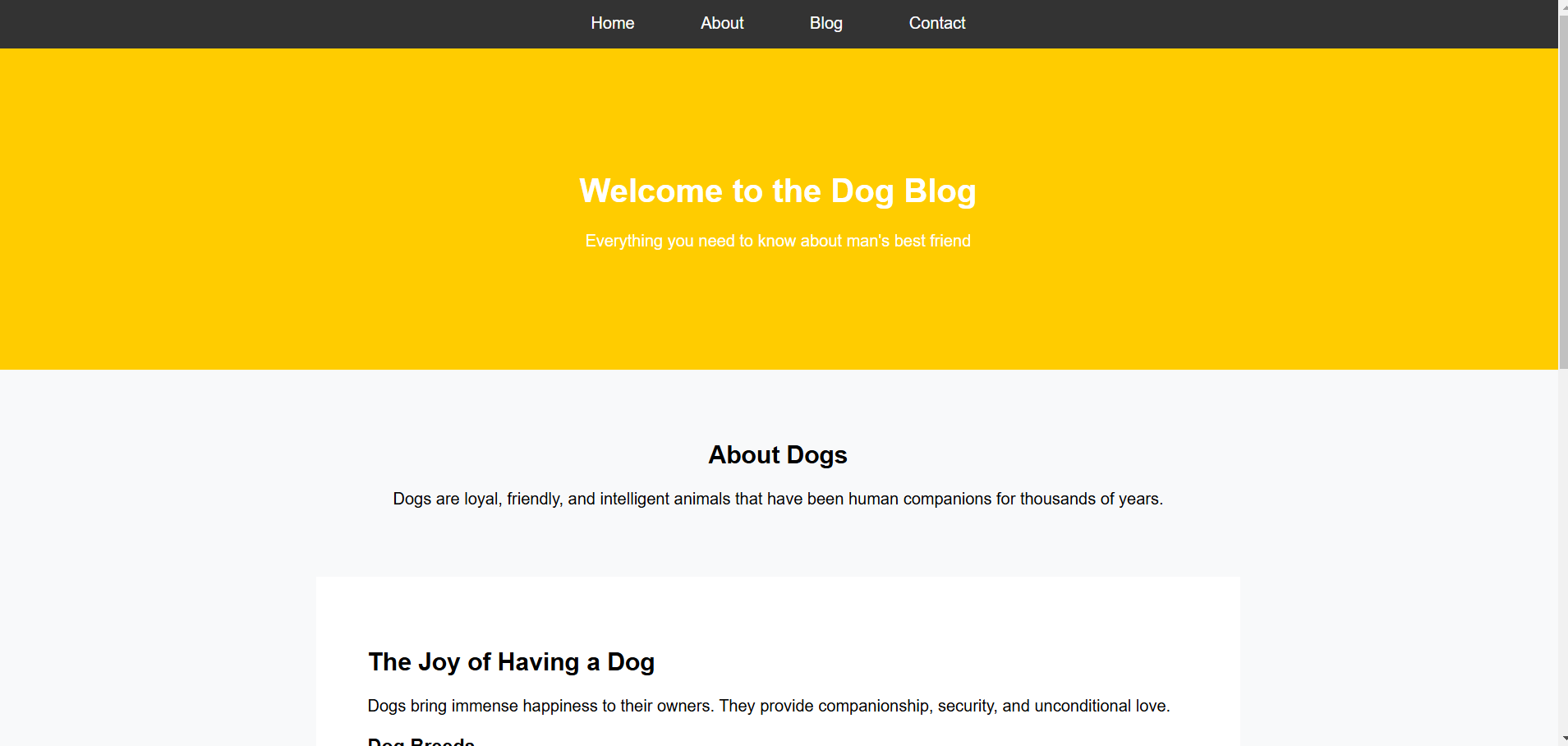


>paste it on chrome

You see your web-site is LIVE



>Refresh the website you can see your another website



Well done!!!!!

Our project is done!!!!!!

Your web site work on private subnet.

We deploy our instance in private subnet. The private subnet does not face internet directly.

In This Project I Use EC2, VPC, Internet Gateway, Route Table, Nat Gateway, Elastic IP, HTML Code, Auto-Scaling Group, Target Group, Application Load Balancer, Launch Templates.