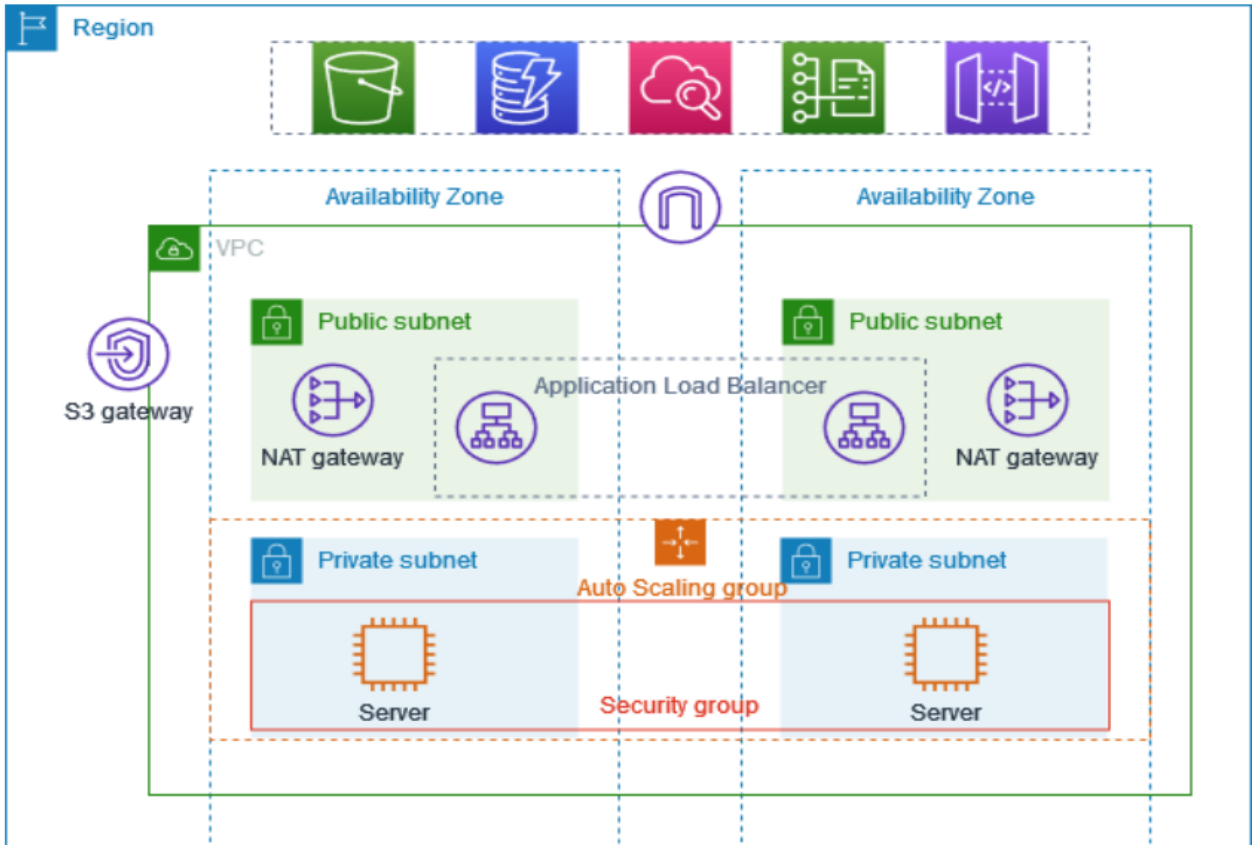
Project Name: VPC with public-private subnet in Production

About the Project

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 the resiliency, you deploy the NAT gateway in both Availability Zones.



Overview

The VPC has public and private subnets in two AZs.

Each public subnet contains a NAT gateway and a load balancer node.

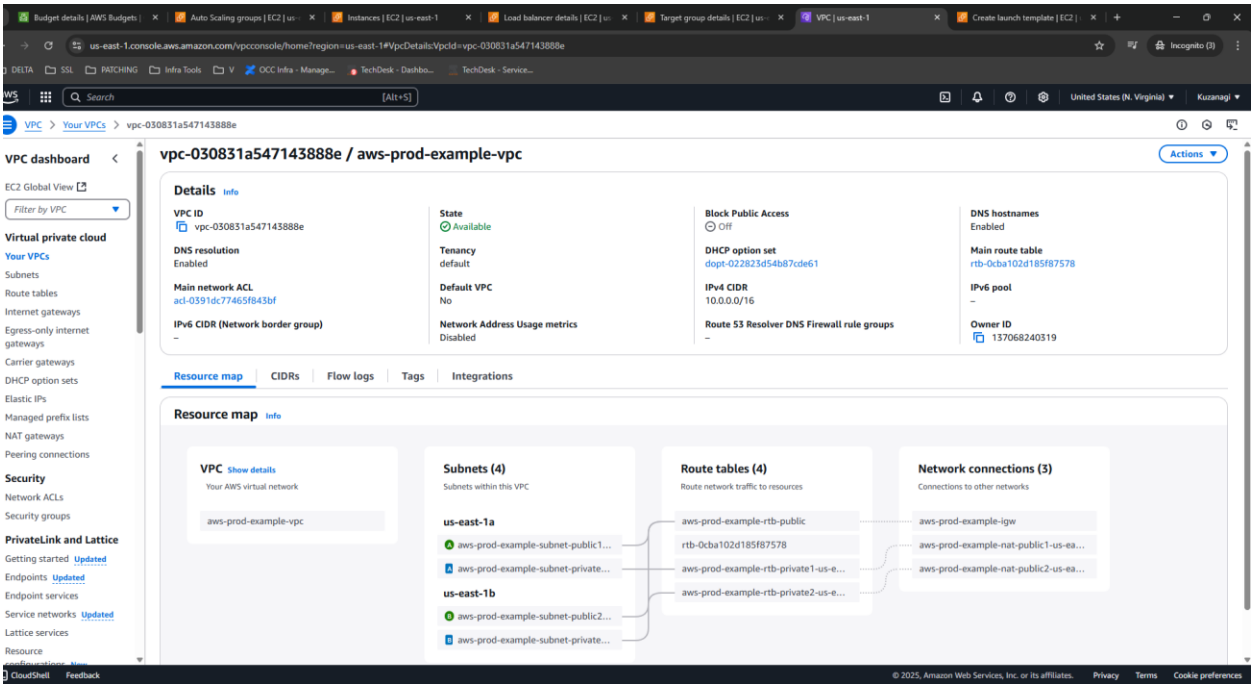
The servers run in the private subnets which are launched and terminated by using an Auto Scaling group and receive traffic from the load balancer.

The servers can connect to the internet by using the NAT gateway. We launch a bastion instance for connecting to the servers in the private subnets for their configuration which includes creating the html landing page and running an http server using python.

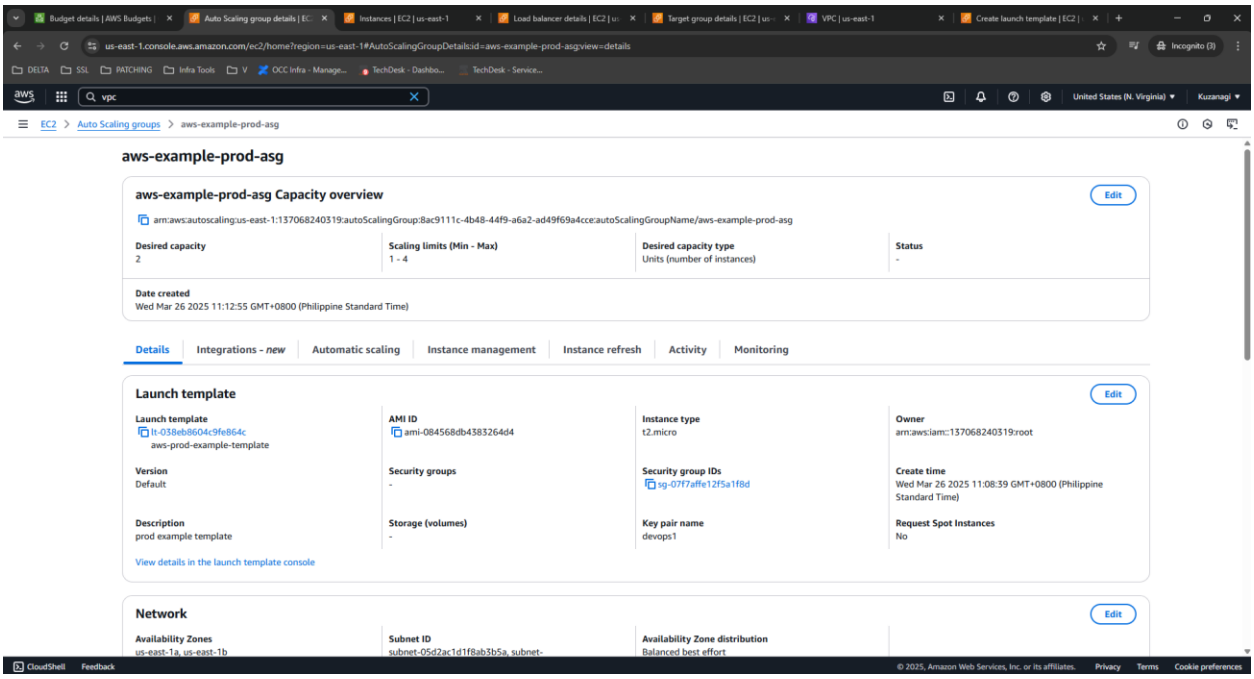
Instructions:

Step-by-step instructions can be followed on [Day-7 | AWS Project Used In Production | Complete Implementation]

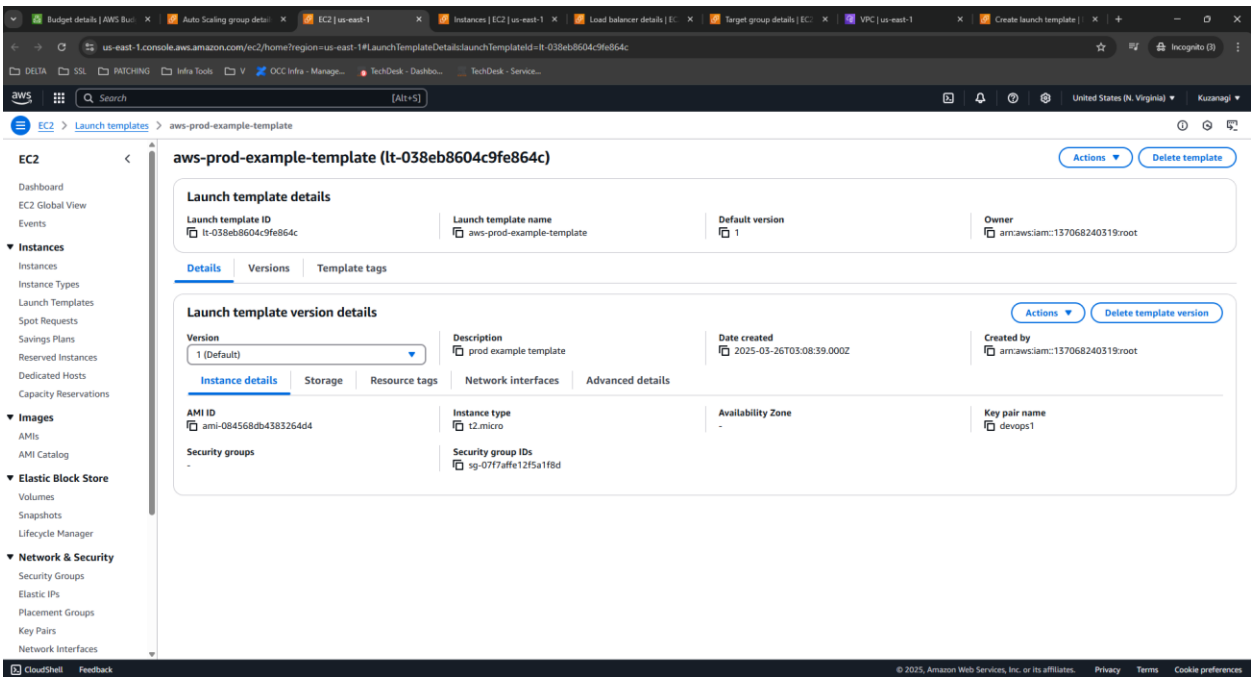
AWS Console setup:
VPC



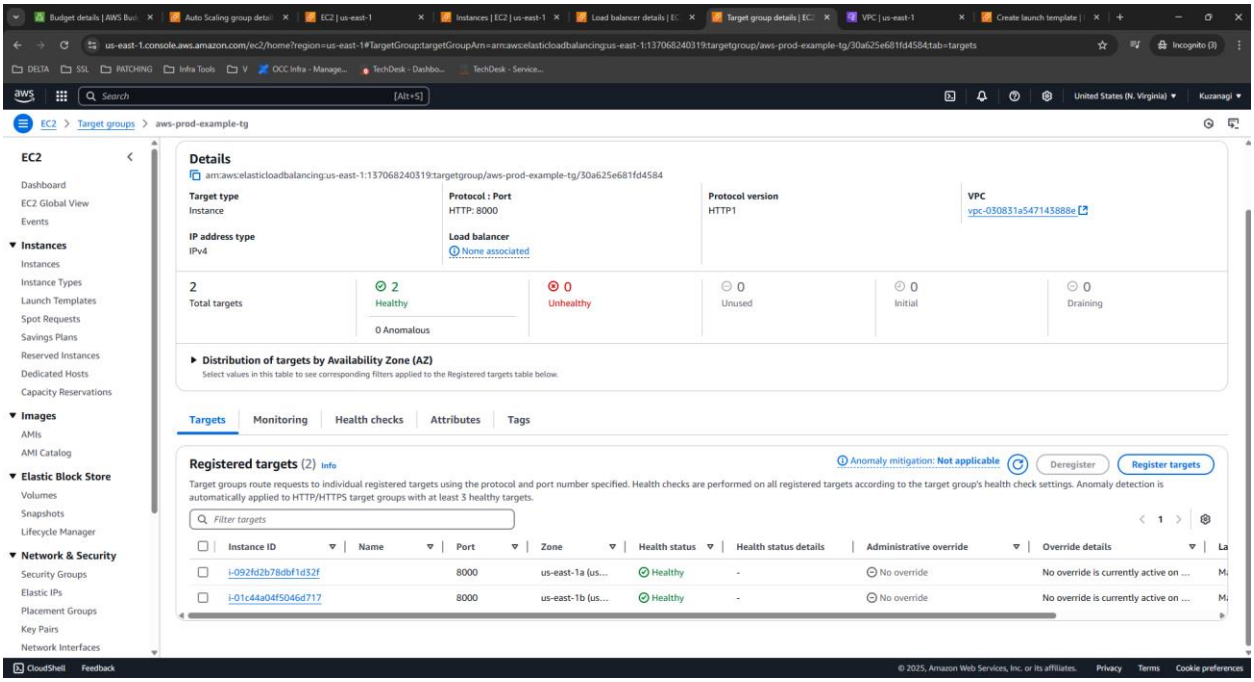
Auto Scaling Group



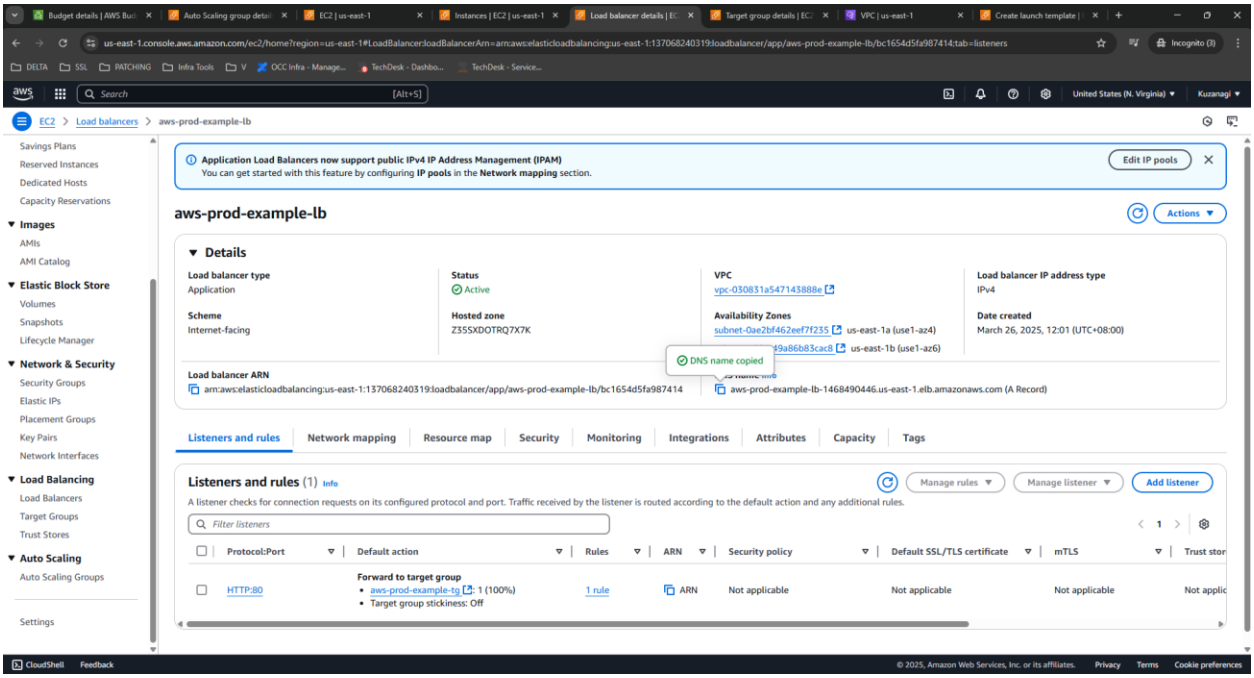
Launch Template



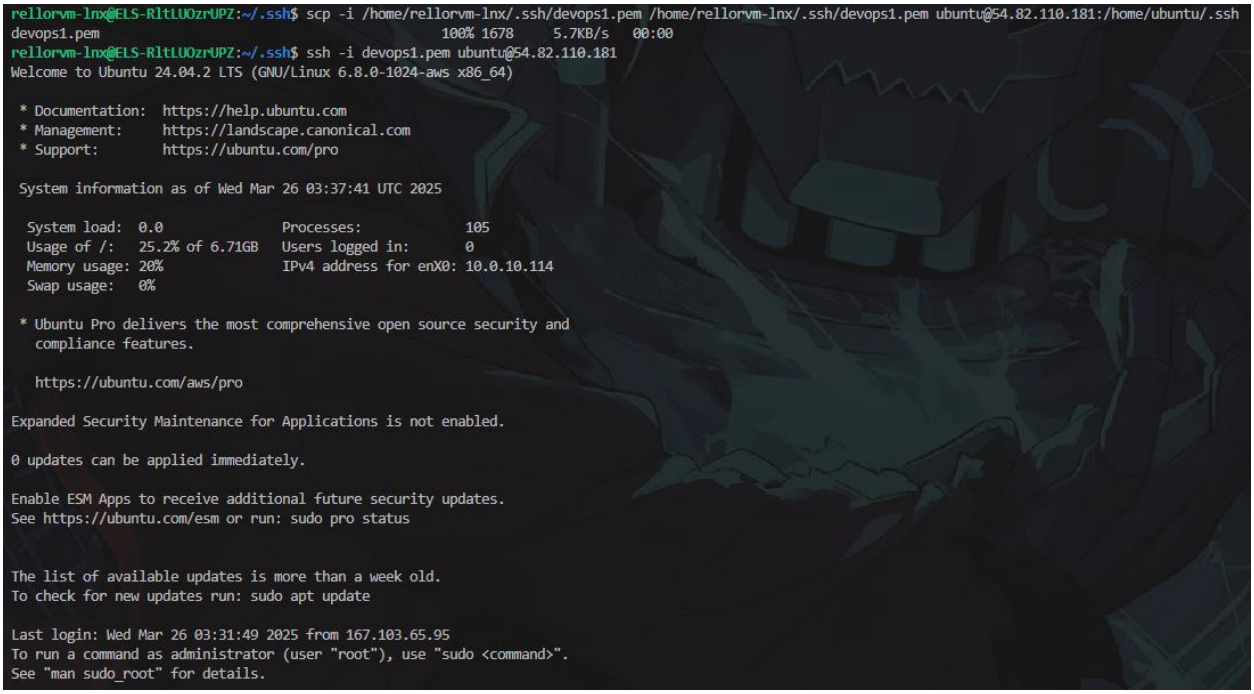
Target Group



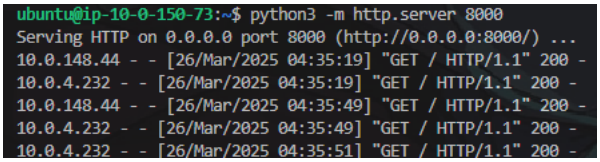
Load Balancer



Bastion setup

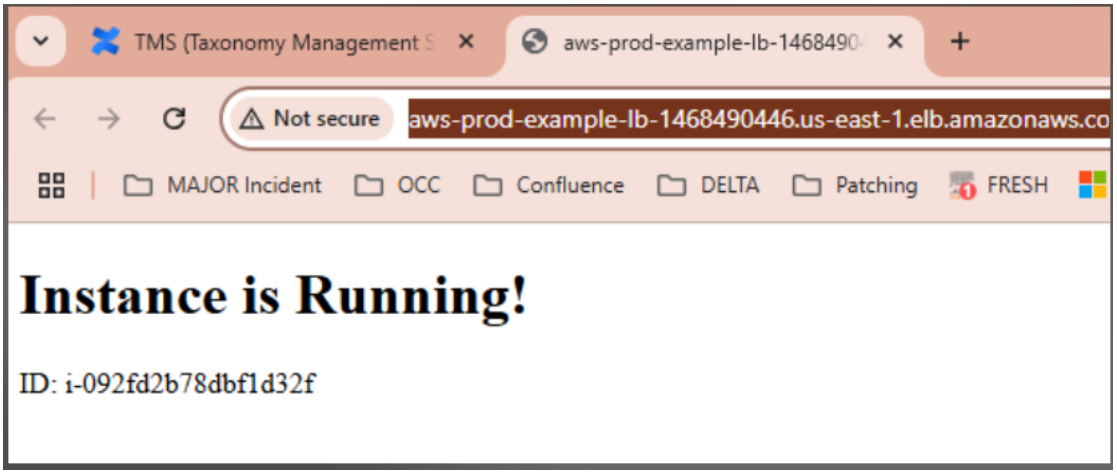


Running http server on both EC2 instances hosted in the Private subnet

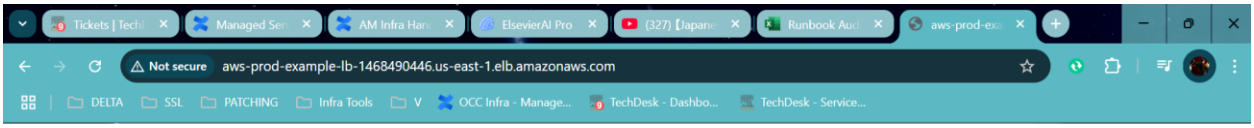


Output:

Accessing the DNS of the load balancer (shows html landing page set on Instance #1)



Accessing the DNS of the load balancer (shows html landing page set on Instance #2)



2nd Instance Running

ID: i-01c44a04f5046d717