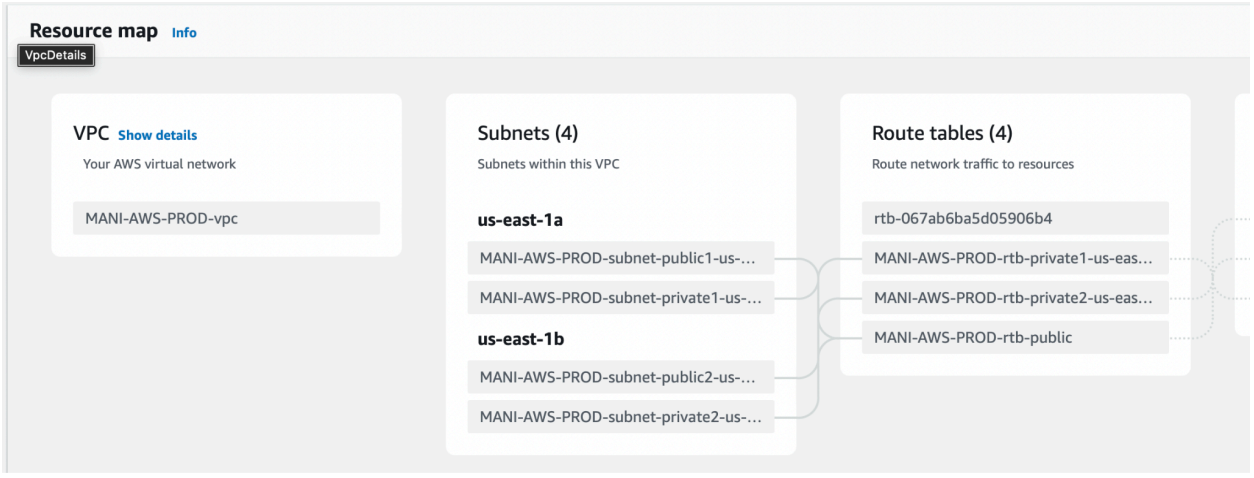
AWS Production-Ready VPC Deployment and High-Performance Application Hosting

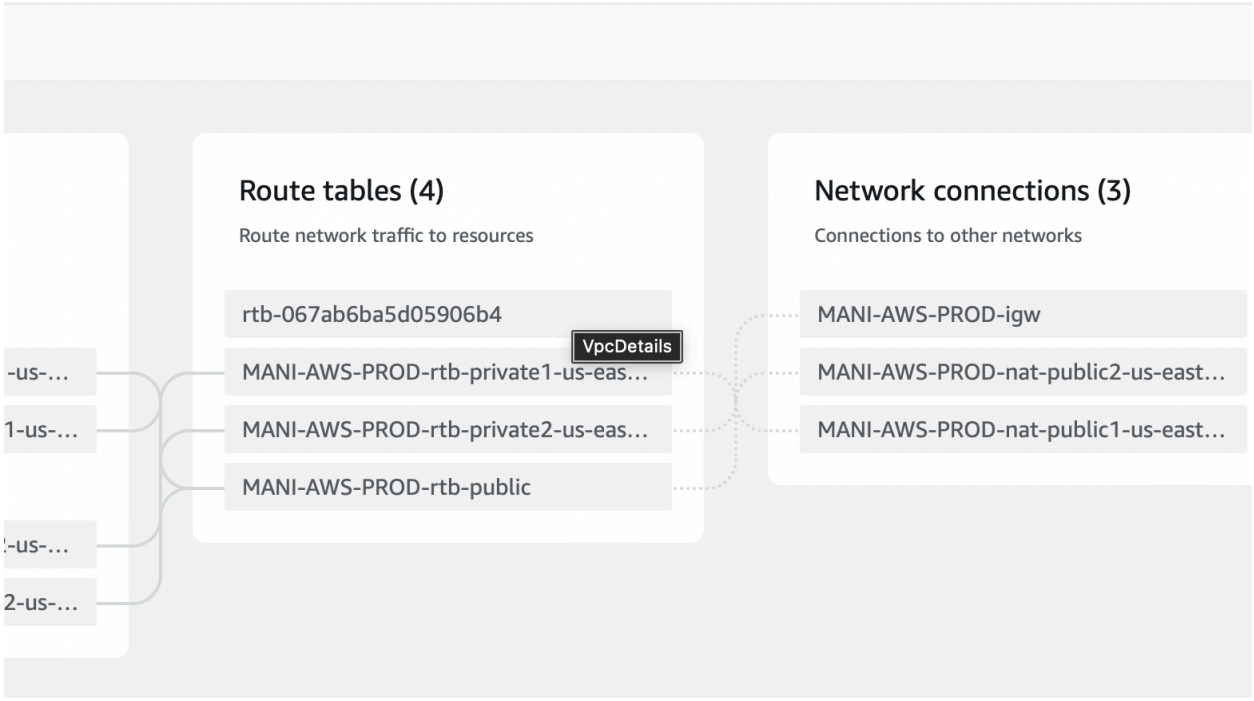
Manikandan Muthukumaran

Step 1: VPC Setup

VPC Creation:

- Created a Virtual Private Cloud (VPC) with two public subnets and two private subnets across two different availability zones.
- Configured route tables, associating one with public subnets and two with private subnets.
- Mapped public subnets with an Internet Gateway and set up two NAT Gateways per availability zone for private subnets.

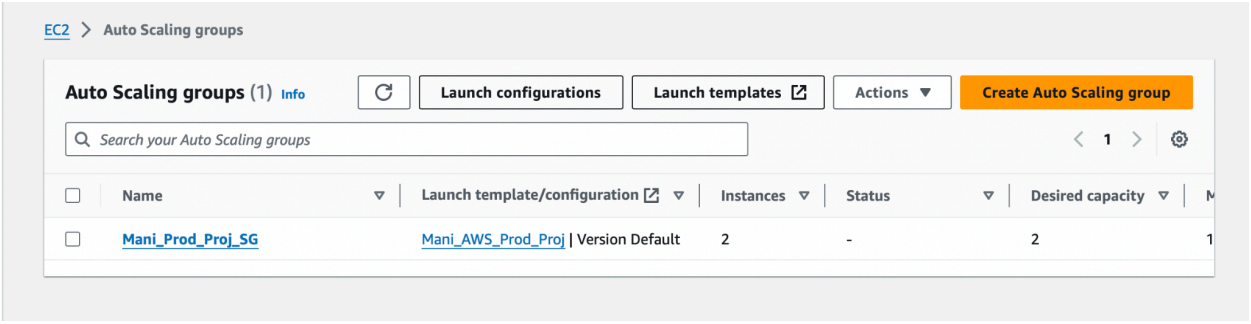




Step 2: Auto Scaling and Instance Launch

Auto Scaling Group:

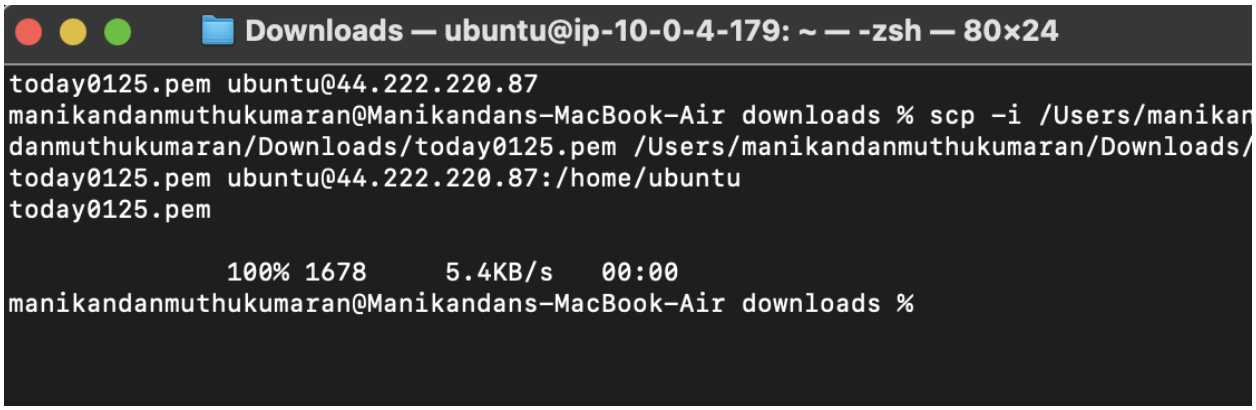
- Set up an Auto Scaling Group template with Ubuntu OS, associating it with the previously created VPC.
- Configured inbound traffic rules to allow access from any source.
- Created an Auto Scaling Group with a minimum of one instance, a desired capacity of two instances, and a maximum of three instances.
- Launched instances in private subnets across two availability zones.

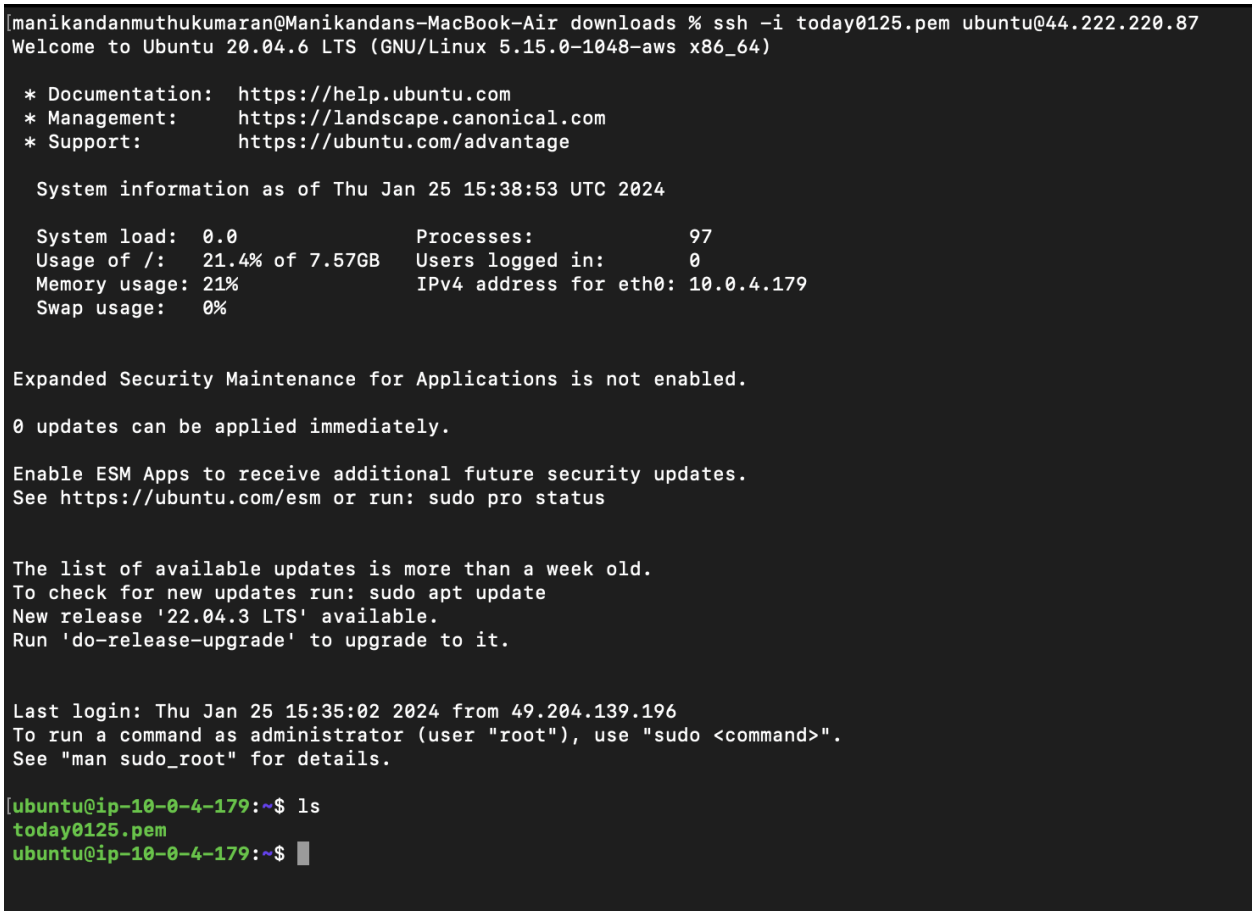


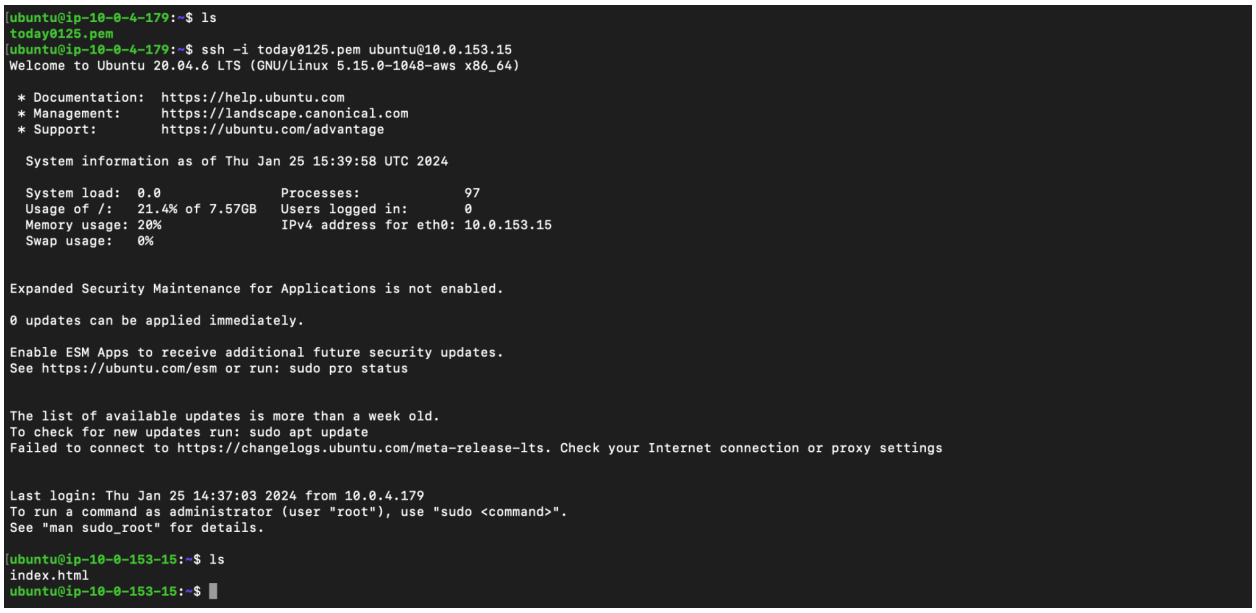
Step 3: Bastion Host Configuration

Bastion Host:

- Created a bastion host with auto-assigned IP addresses for secure access.
- Transferred the key pair file from the local machine to the bastion host.
- Used the bastion host to log into one of the private instances and create a new `index.html` file.
- Ran a Python HTTP application on port 8000



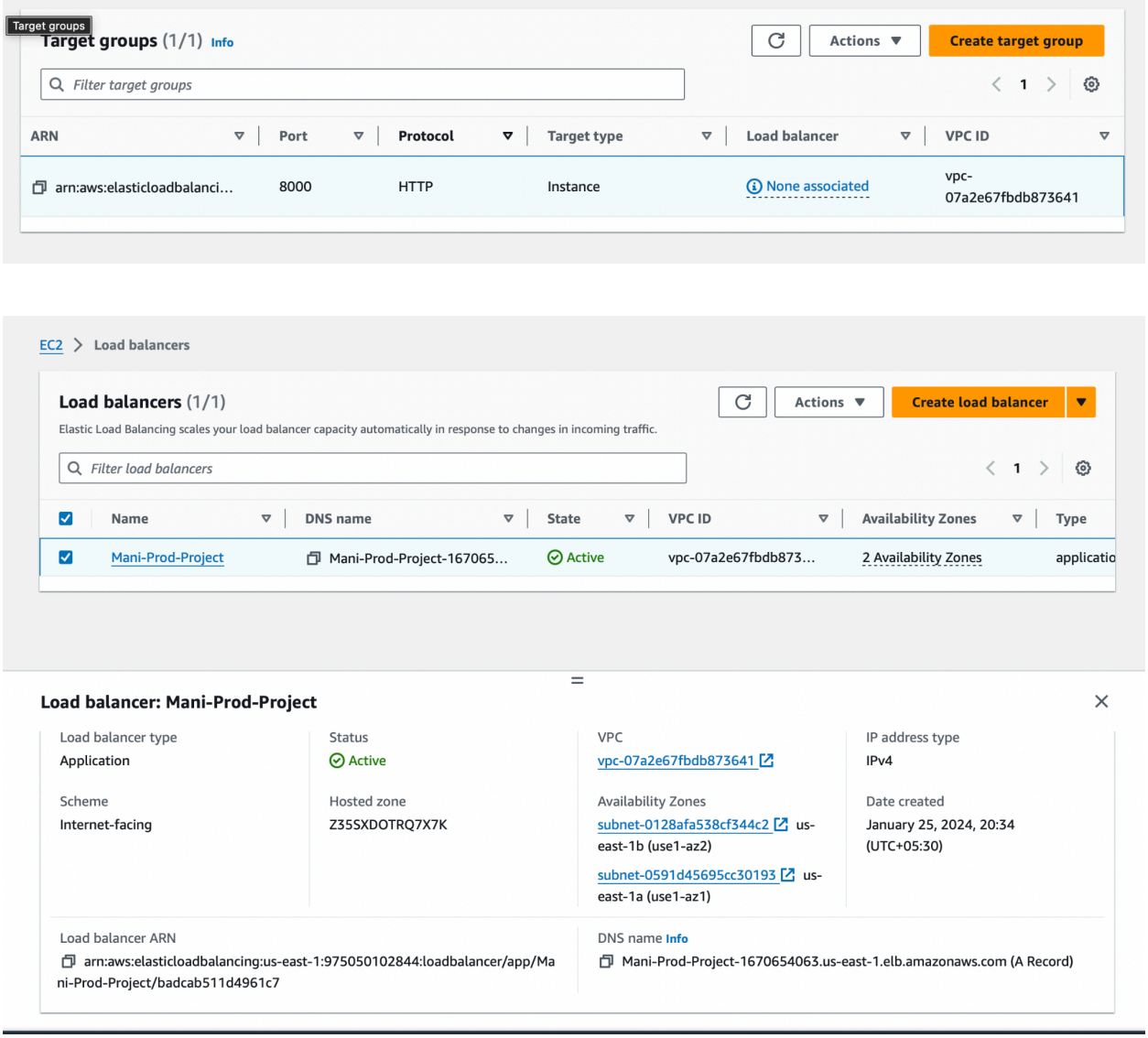




Step 4: Load Balancer Configuration

Load Balancer and Target Groups:

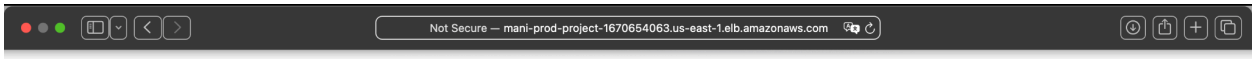
- Created target groups and associated them with instances launched by the Auto Scaling Group on port 8000.
- Configured an Application Load Balancer, associating it with the previously created target groups.



Step 5: Application Deployment

Application Deployment:

- Accessed the application using the DNS provided by the Load Balancer.
- Verified the successful deployment of the HTML application.



Mani's first prod project

Hi there, thanks for viewing my work :).

Thank you!