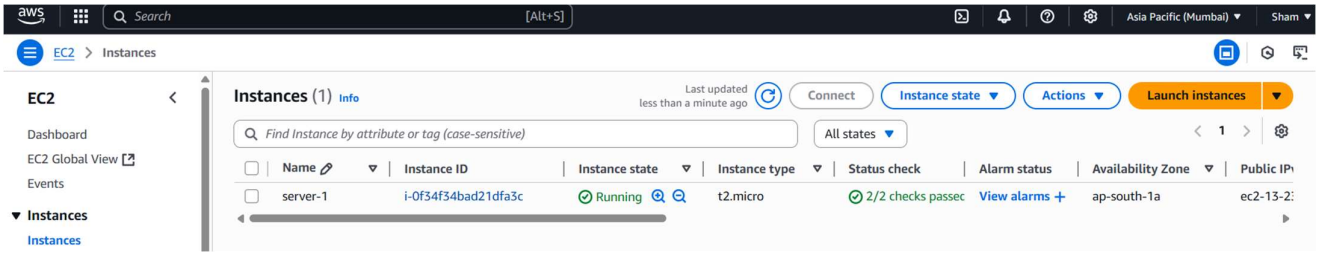
Task: Run a docker container with port mapping and access through load balancer.

Creating two separate EC2 Instances and installing docker.

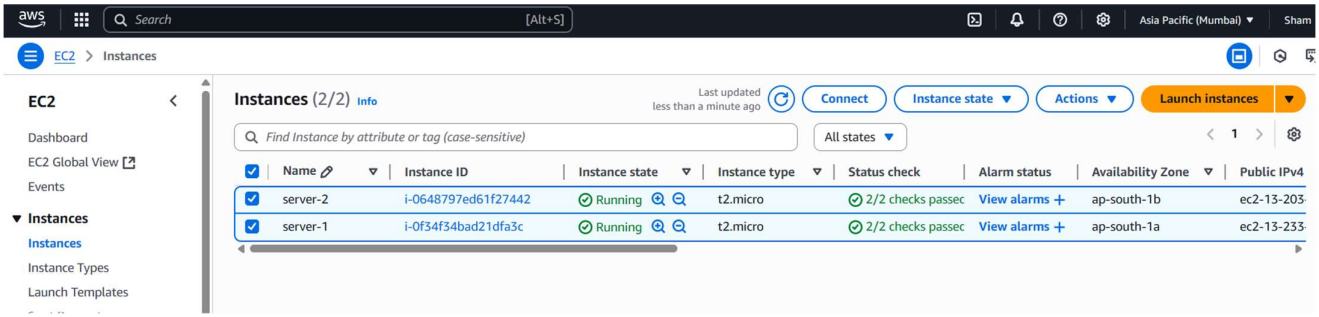
1. **Server-1, OS: Amazon Linux**




```
[root@ip-172-31-42-62 ~]# docker --version
Docker version 25.0.8, build 0bab007
[root@ip-172-31-42-62 ~]# systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled;>
     Active: active (running) since Tue 2025-05-27 08:52:03 UTC; 27s a>
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Process: 27877 ExecStartPre=/bin/mkdir -p /run/docker (code=exited>
    Process: 27878 ExecStartPre=/usr/libexec/docker/docker-setup-runti>
   Main PID: 27879 (dockerd)
      Tasks: 7
     Memory: 37.9M
        CPU: 267ms
     CGroup: /system.slice/docker.service
             └─27879 /usr/bin/dockerd -H fd:// --containerd=/run/conta>

May 27 08:52:02 ip-172-31-42-62.ap-south-1.compute.internal systemd[1]>
May 27 08:52:02 ip-172-31-42-62.ap-south-1.compute.internal dockerd[27>
May 27 08:52:02 ip-172-31-42-62.ap-south-1.compute.internal dockerd[27>
```

2. **Server-2, OS: Ubuntu**

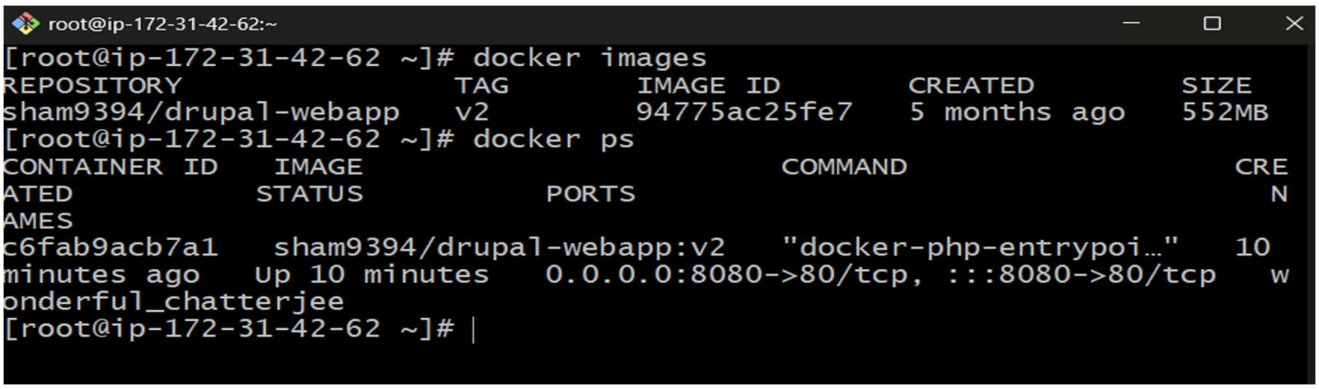


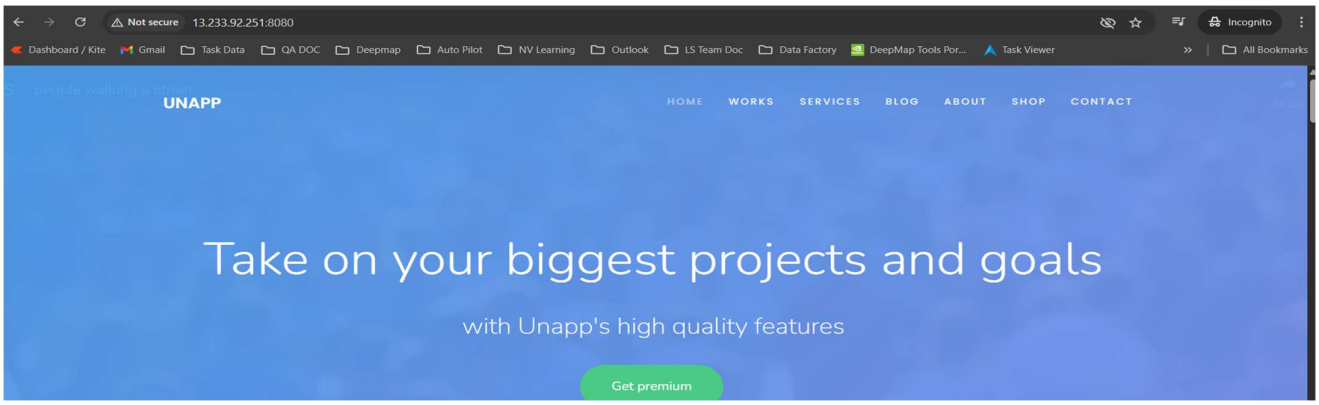

```
root@ip-172-31-9-165: ~
root@ip-172-31-9-165:~# docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
root@ip-172-31-9-165:~# systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enab>
     Active: active (running) since Tue 2025-05-27 08:53:43 UTC;
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 1961 (dockerd)
      Tasks: 8
     Memory: 60.5M (peak: 60.6M)
        CPU: 303ms
     CGroup: /system.slice/docker.service
             └─1961 /usr/bin/dockerd -H fd:// --containerd=/run/c>

May 27 08:53:43 ip-172-31-9-165 systemd[1]: Starting docker.servi>
May 27 08:53:43 ip-172-31-9-165 dockerd[1961]: time="2025-05-27T0>
```

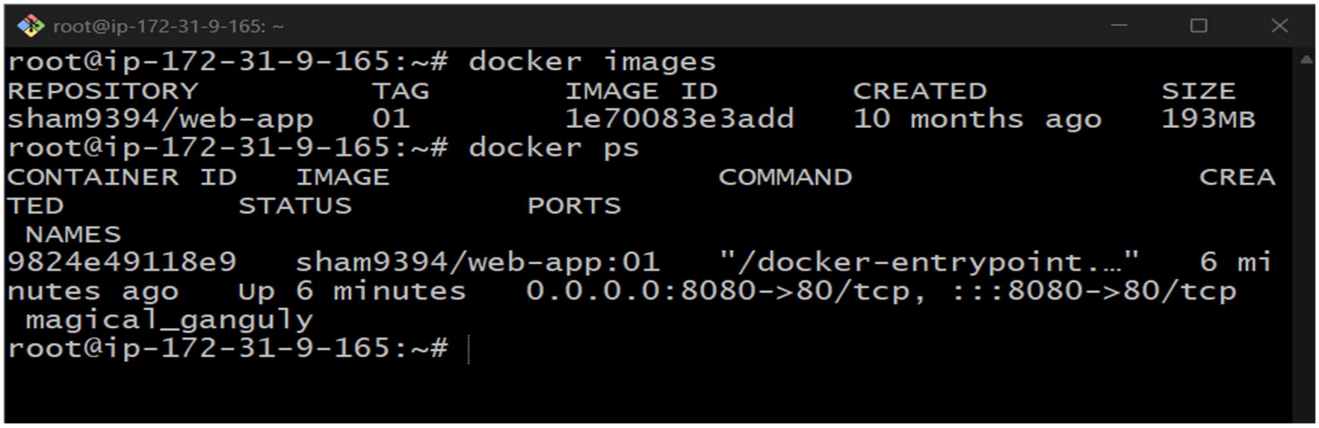
Running container on each EC2 server with port mapping.

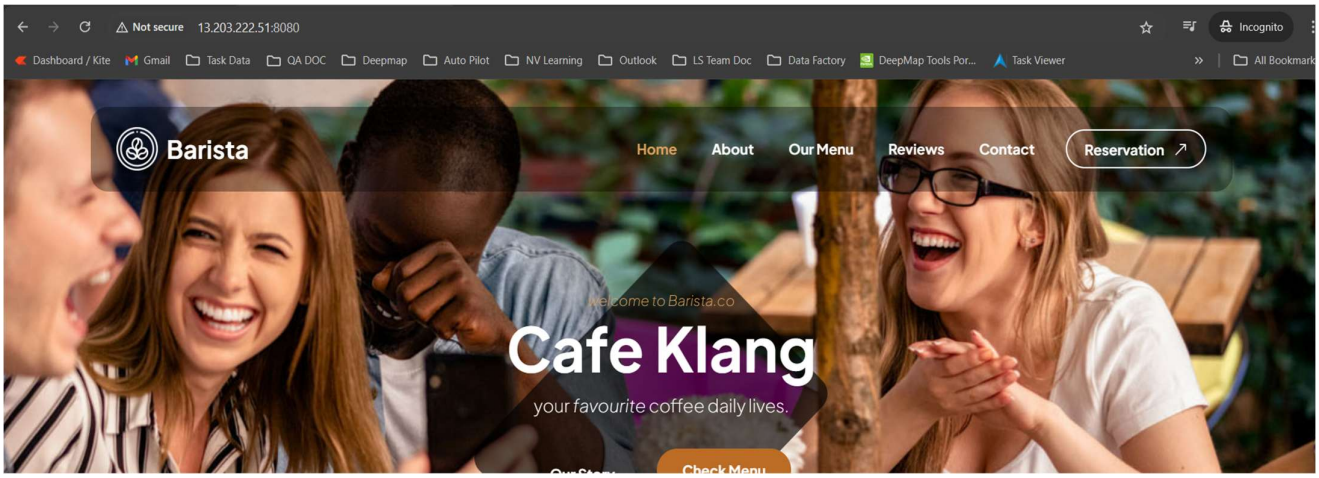
1. **Server-1**





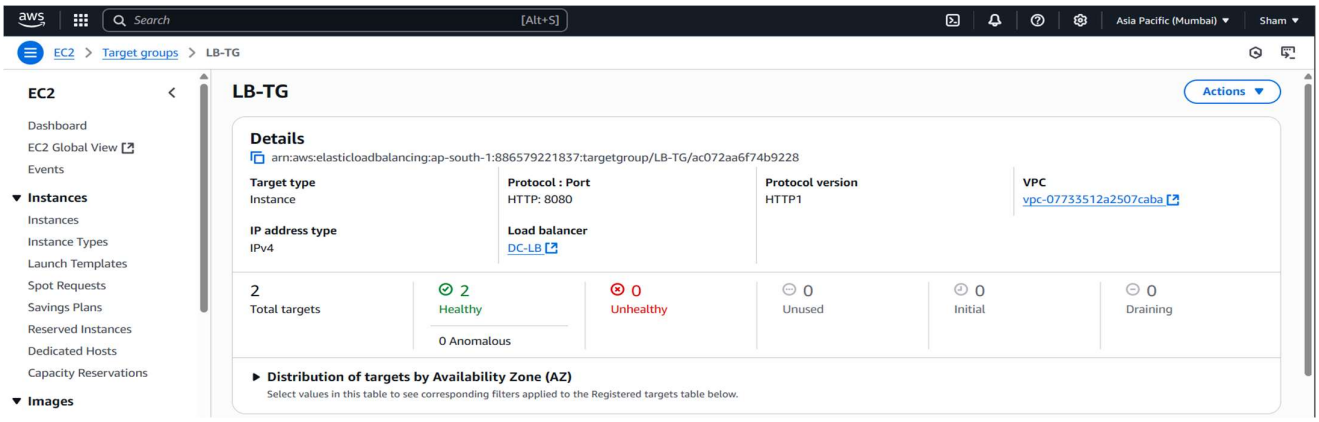
2. **Server-2**

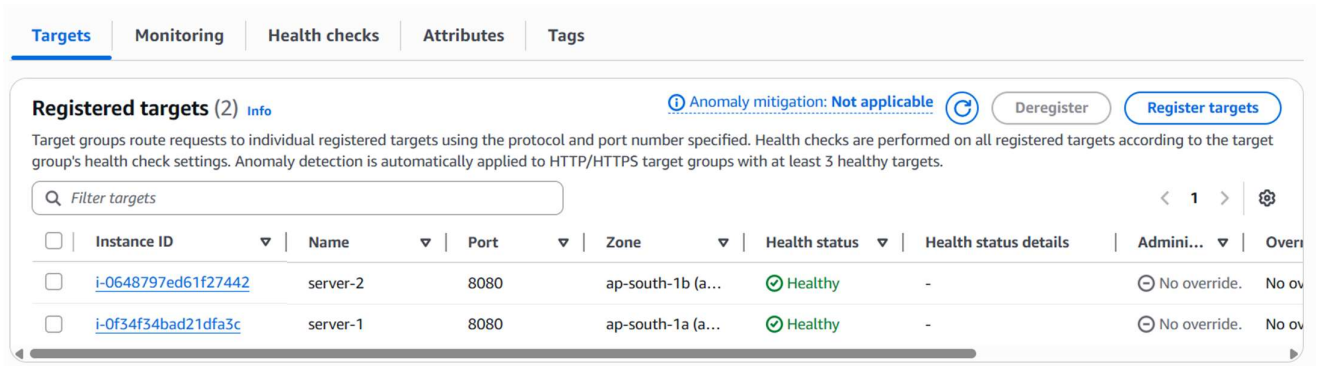




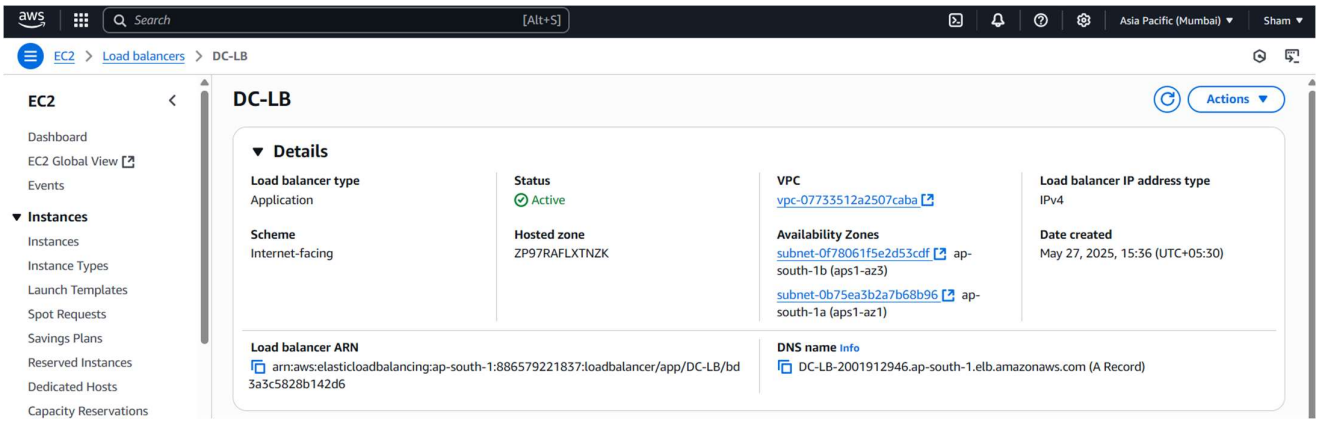
Access application through Load balancer

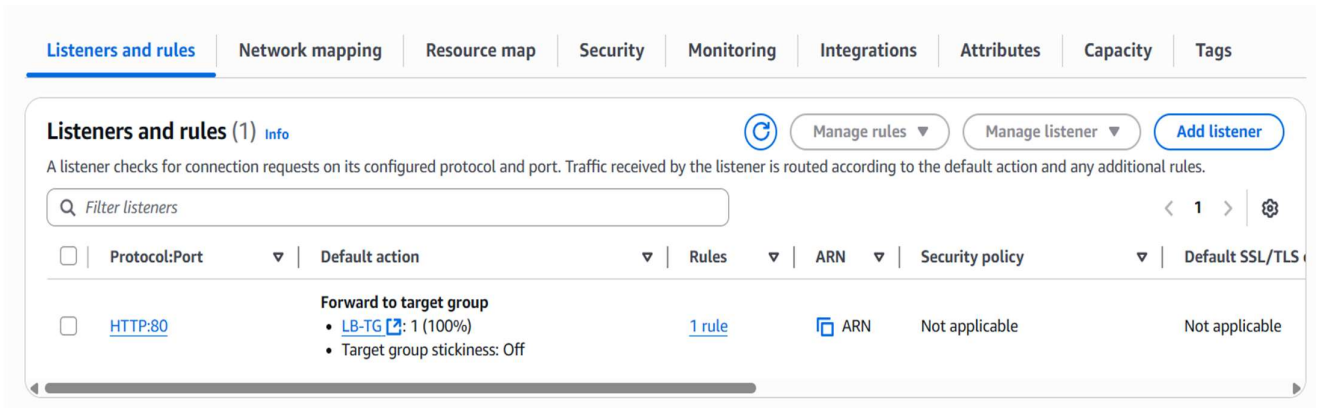
1. Creating Target Group.





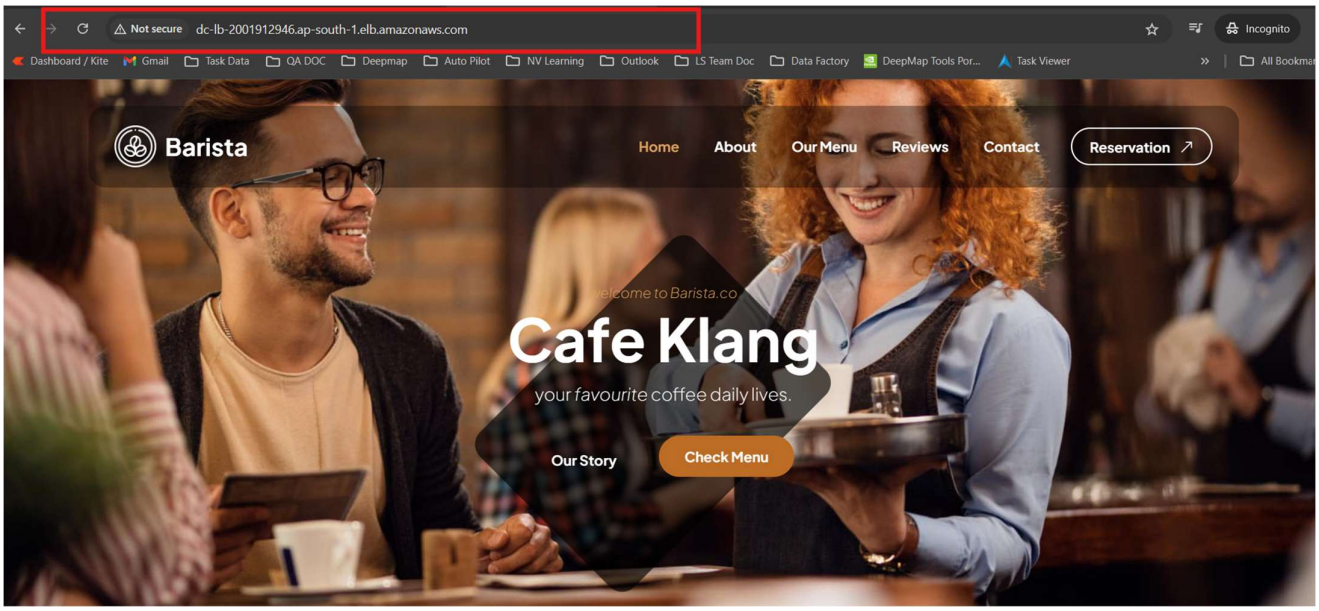
2. Creating Load Balancer.





Accessing web application via Load Balancer:

1. **Initial Output from Server-2**



2. **After refresh output from Server-1.**