**Task 7**

**Task description:** Implement load balancing and high availability to open source application.

This task is in two parts

**Part 1**: you need to explore the methodologies and possibilities that can be implemented in load balancing and creating an application with high availability and document them.

**Part 2:** To implement one of the explored methodologies in real time and document it.

* I did this task in AWS

**Part 1 –** Load balancing and high availability methodologies.

**Load Balancing:**

Load balancing distributes incoming network traffic across multiple servers to ensure no single server is overwhelmed, thus optimizing resource utilization, maximizing throughput, and minimizing response time. Some methodologies for load balancing include:

* Application load balancer:

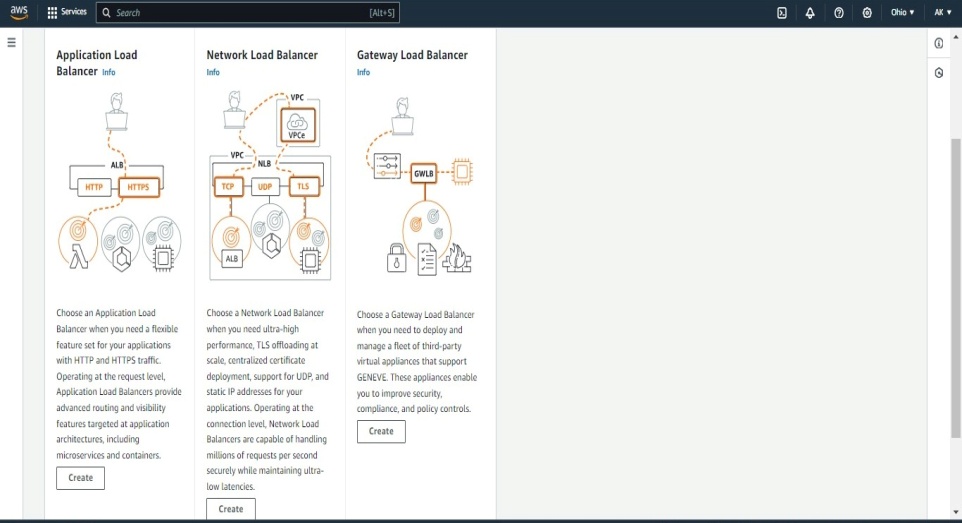
🡪When you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

* Network load balancer:

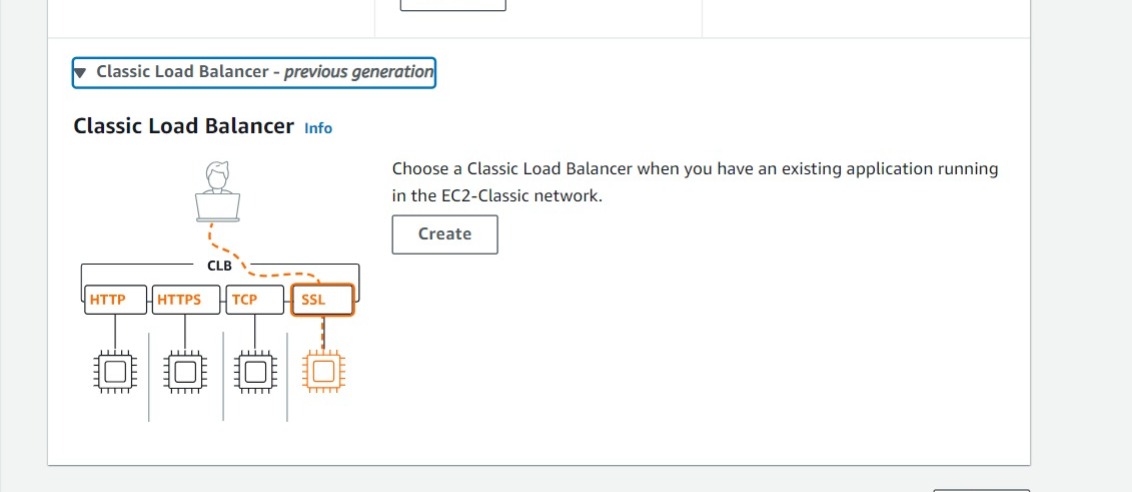
🡪 When you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

* Gateway load balancer:

🡪 When you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.



* Classic load balancer – previous generation



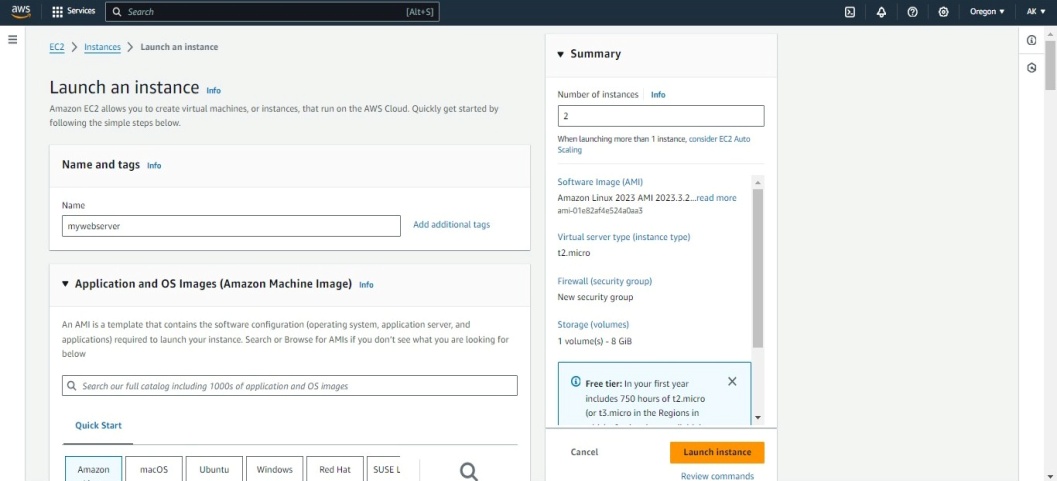
**High Availability:**

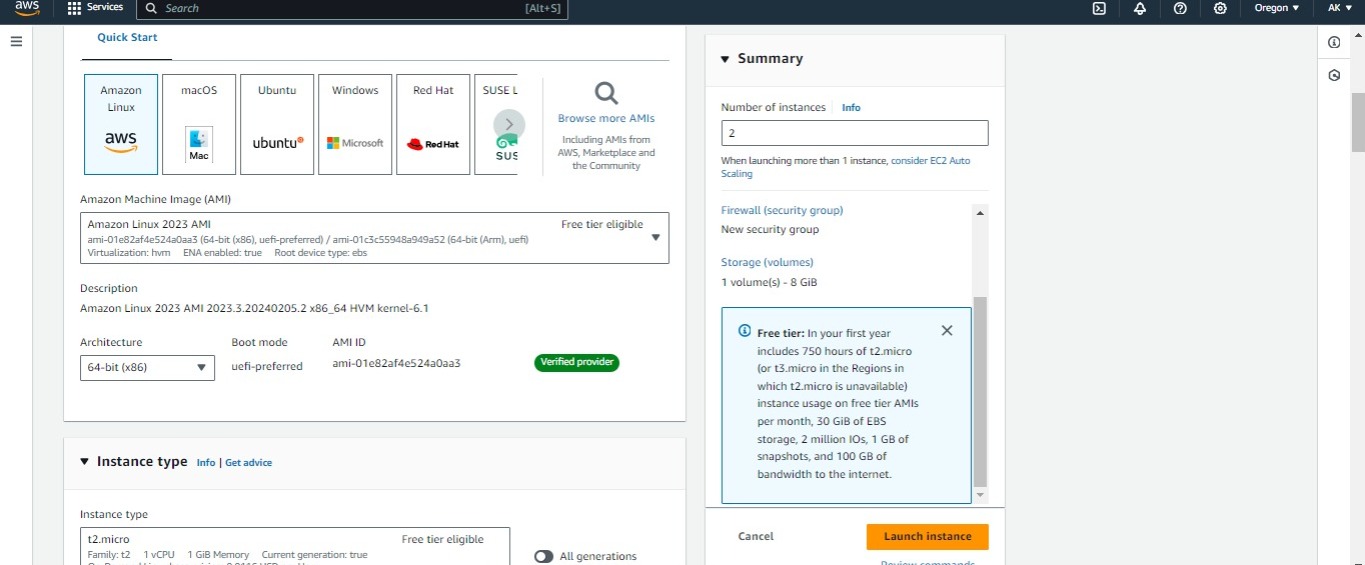
High availability ensures that a system remains operational and accessible for users for the vast majority of the time. Key methodologies include:

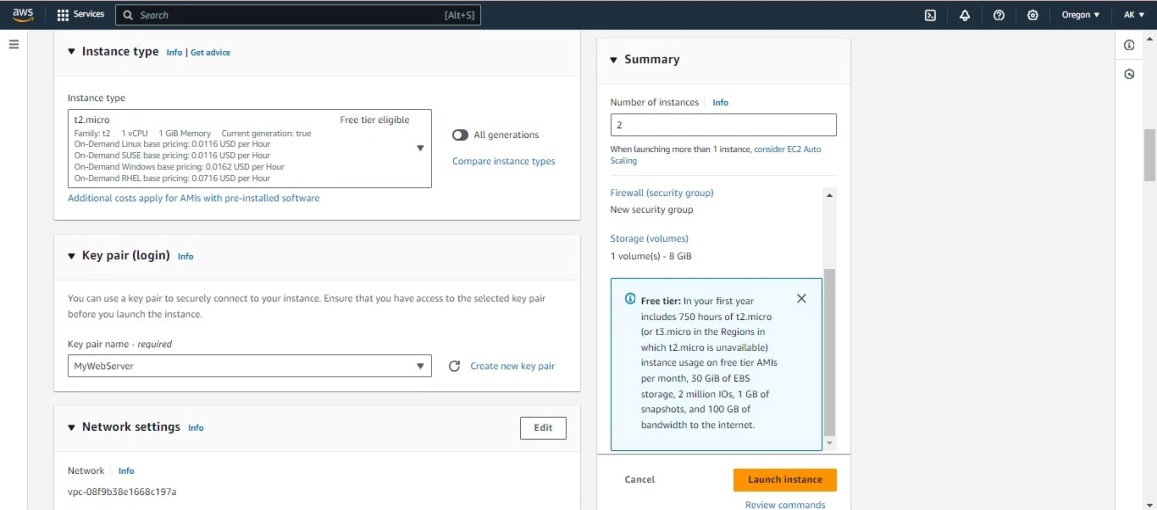
* Redundancy: Deploying multiple instances of critical components to ensure failover capability.
* Failover Mechanisms: Automated processes that detect failures and switch to redundant components seamlessly.
* Load Balancing: Distributing incoming traffic across multiple servers helps to avoid single points of failure and ensures continuous service availability.
* Geographic Redundancy: Deploying redundant systems across different geographical locations to mitigate risks associated with localized failures or disasters.
* Automated Monitoring and Recovery: Implementing systems that constantly monitor the health of components and automatically recover from failures.

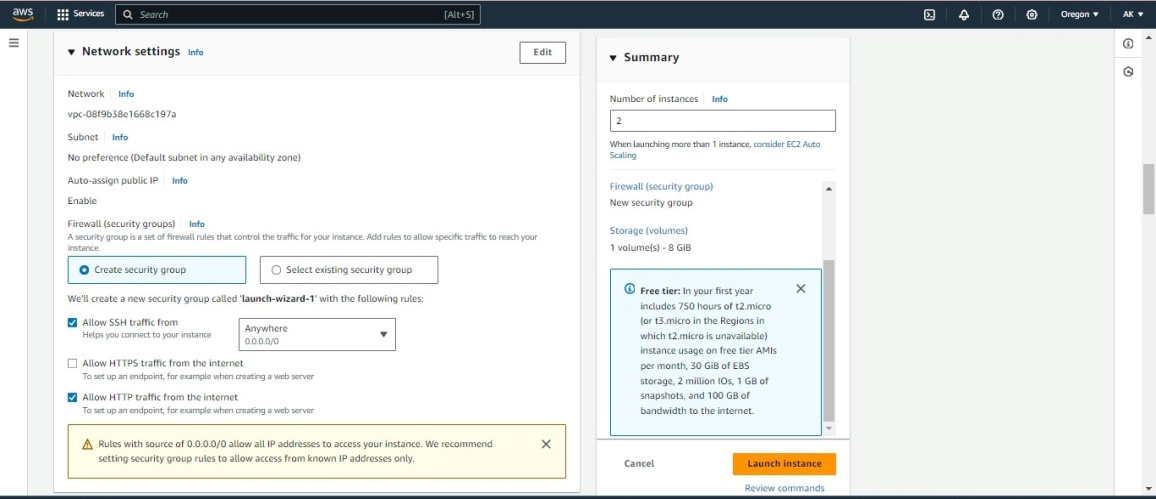
**Part 2** – Implementation

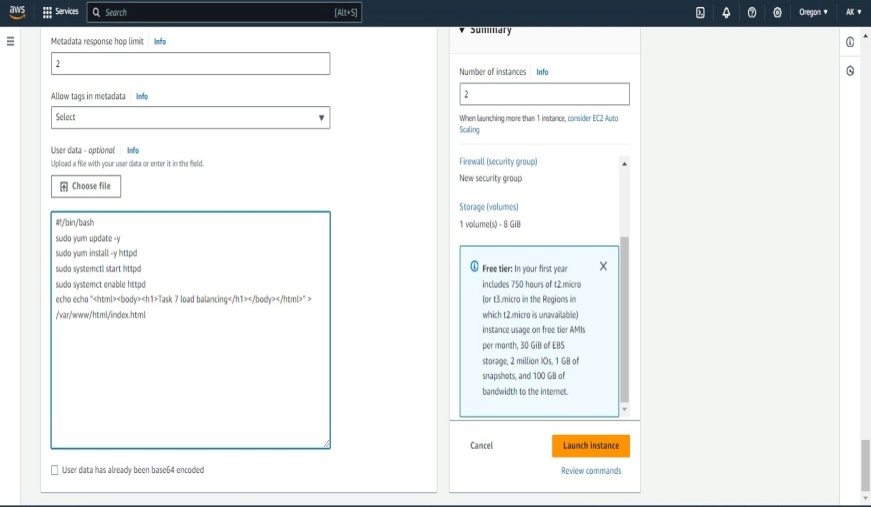
* Create two EC2 instance.



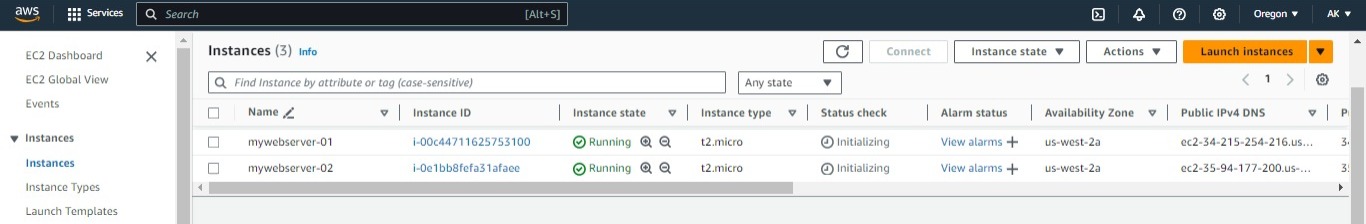


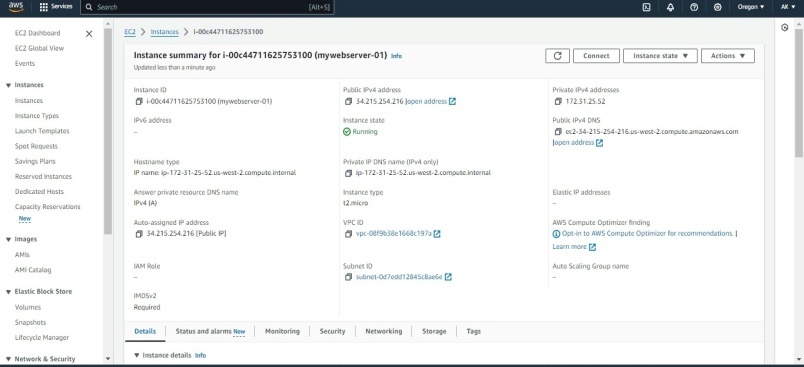


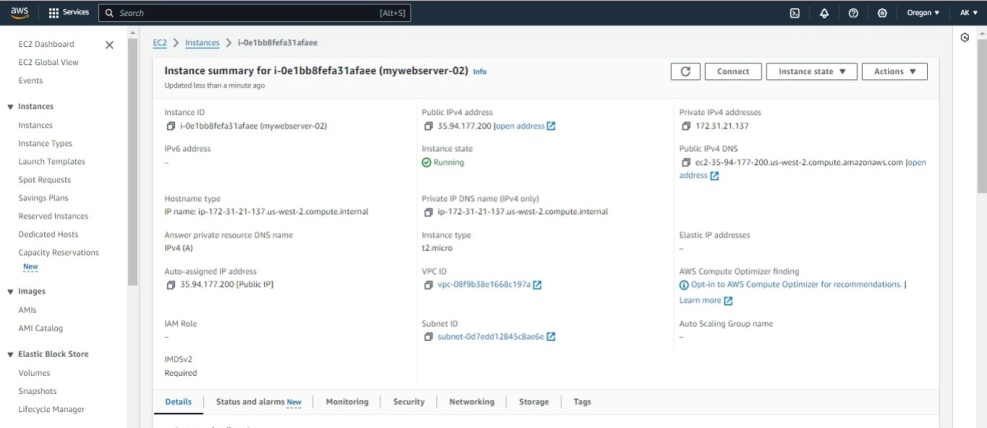




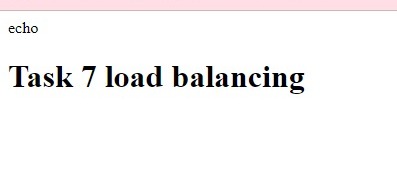
* These are my instances.



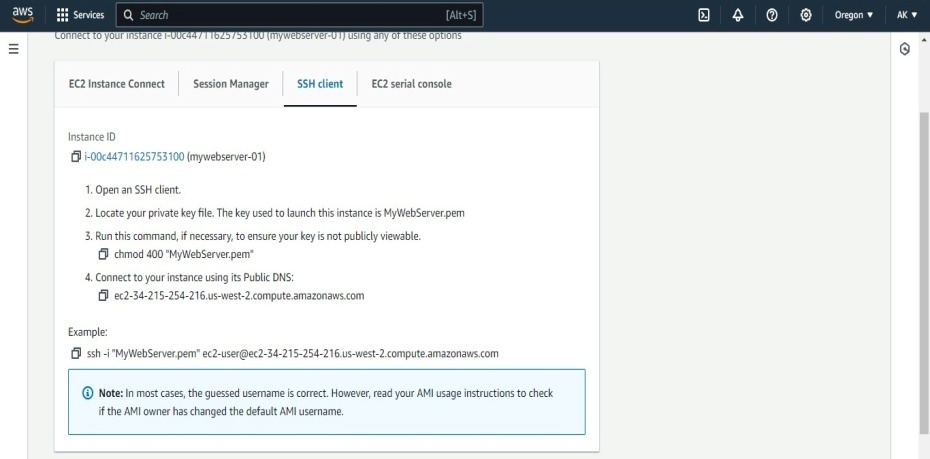


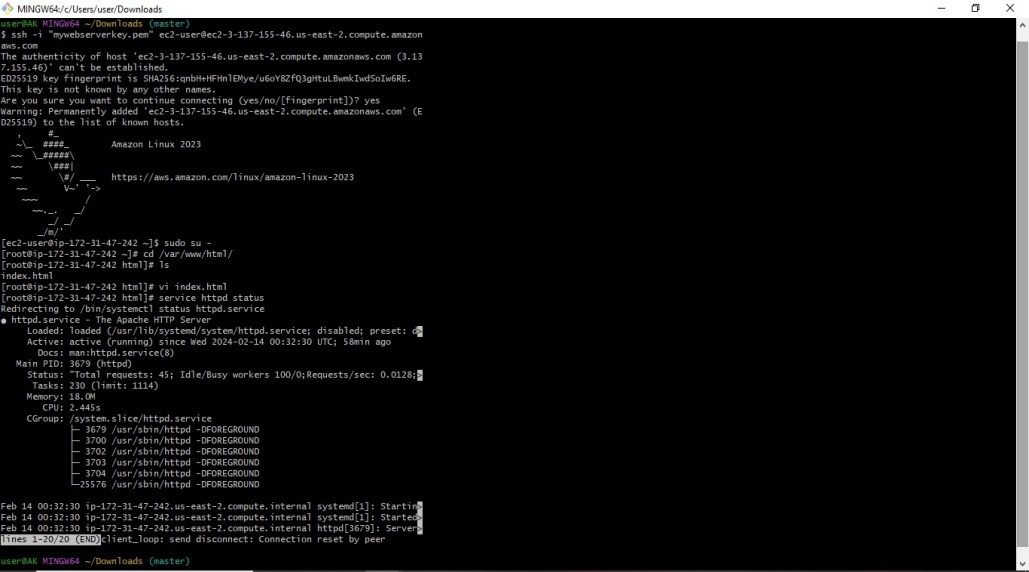


* After creating two instances, try to access the website.
* Copy the publicIPv4 address and paste it in new tab. Then I accessed this from my servers.

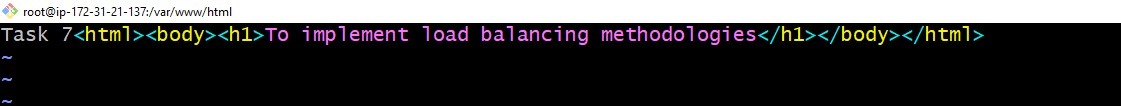


* If we want to change the configuration, connect the instance with our key value in git bash.





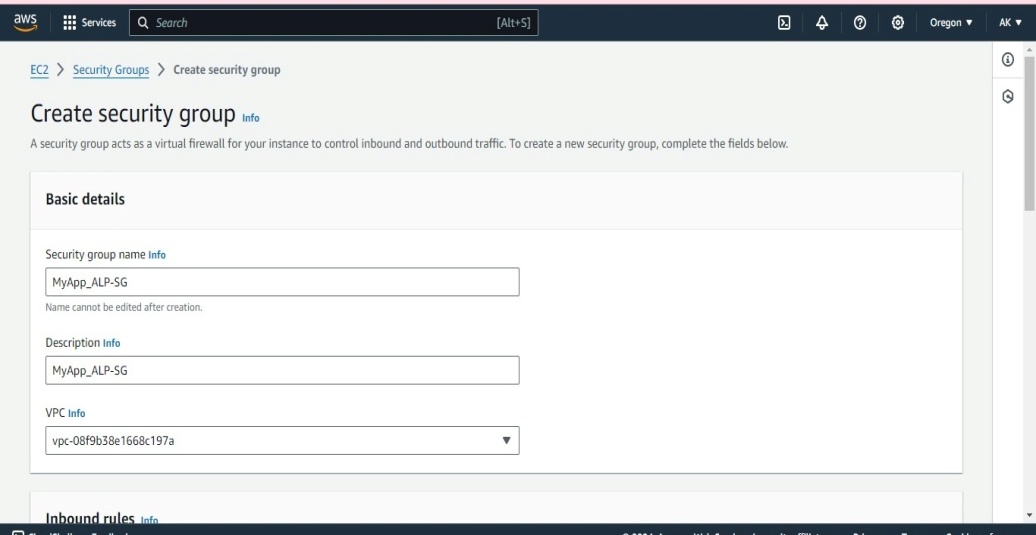
* Here I changed the text from git bash.

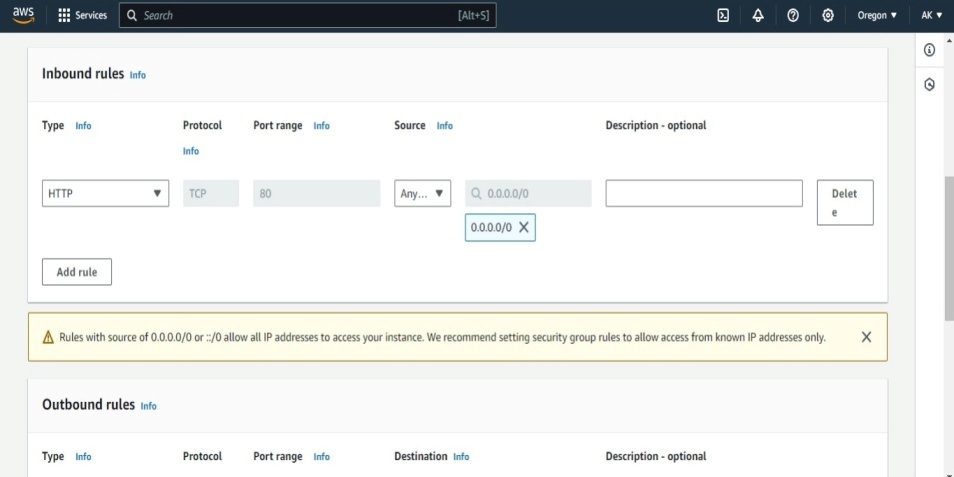


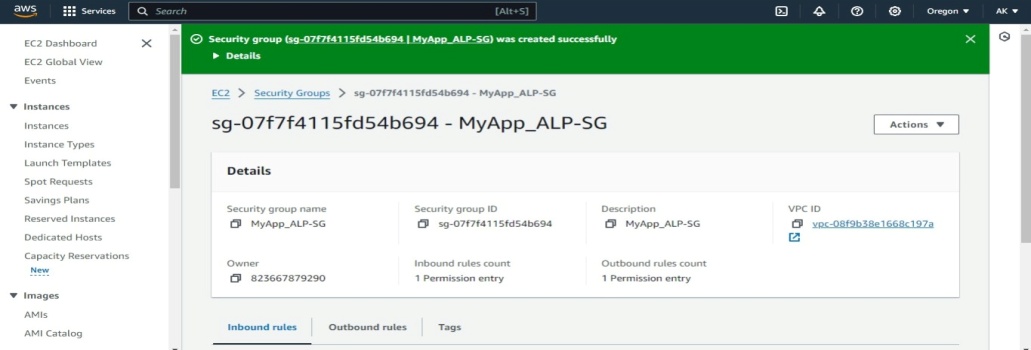
* After refreshing the page, my text is changed.



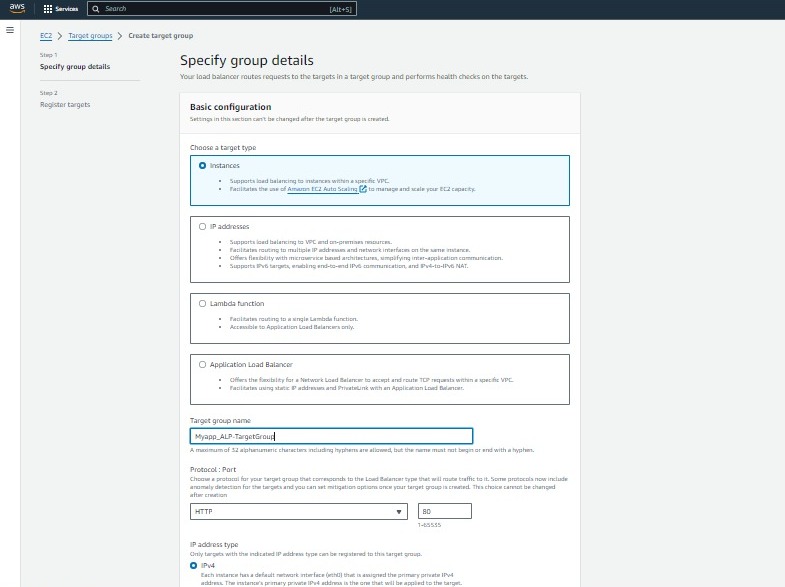
* Now I am able to access my application with individual IP servers
* Now we have to access through load balancer.
* Currently I am creating application load balancer.
* Before that create a security group and target group. My security group name is “MyApp\_ALP-SG”.
* While creating load balancer, select this security group.

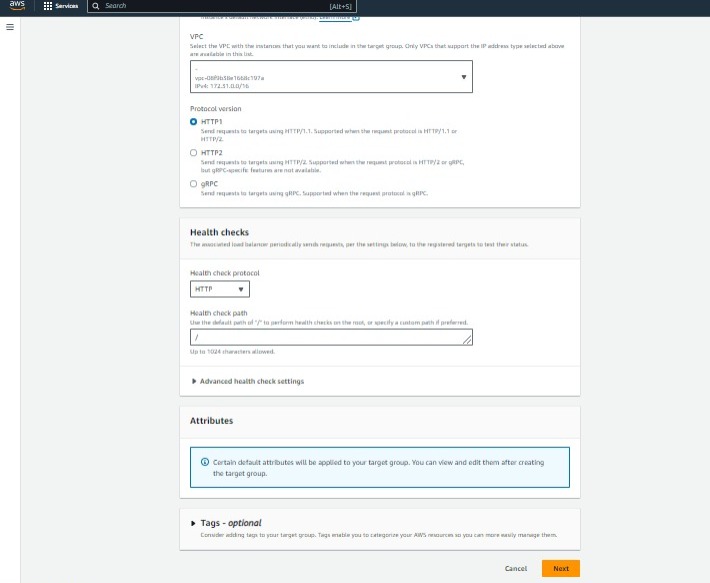




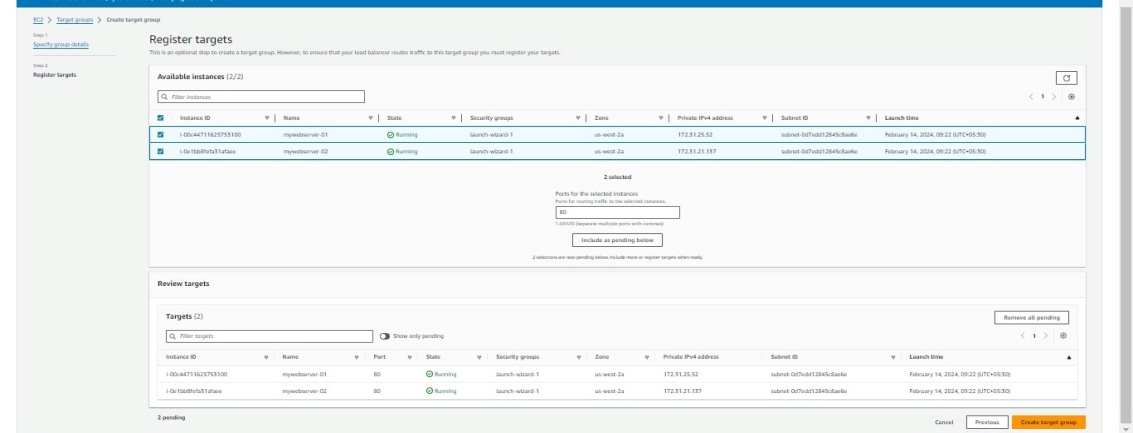


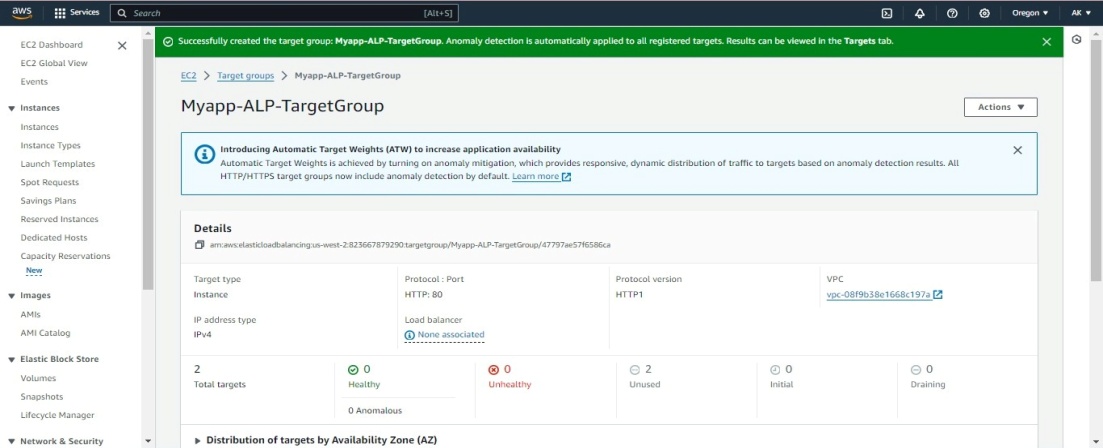
* Create target group.

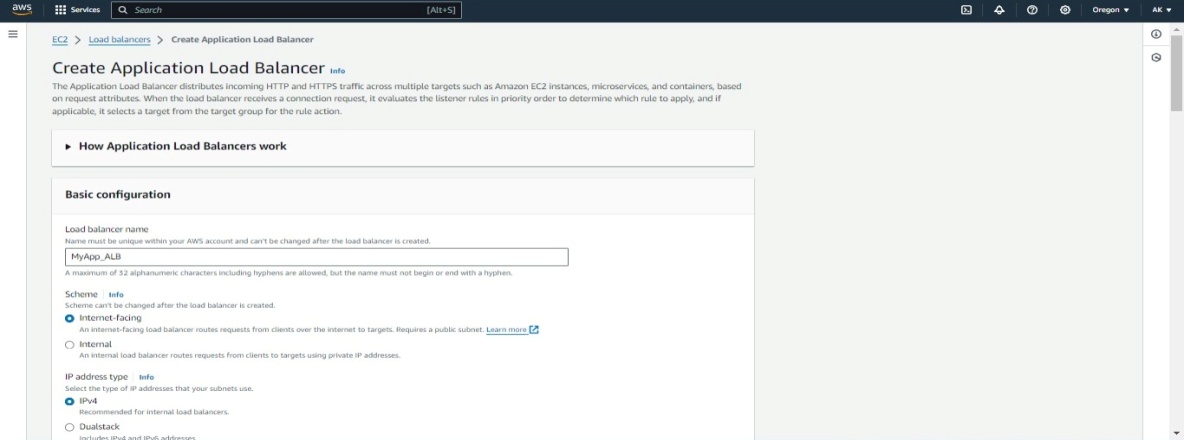


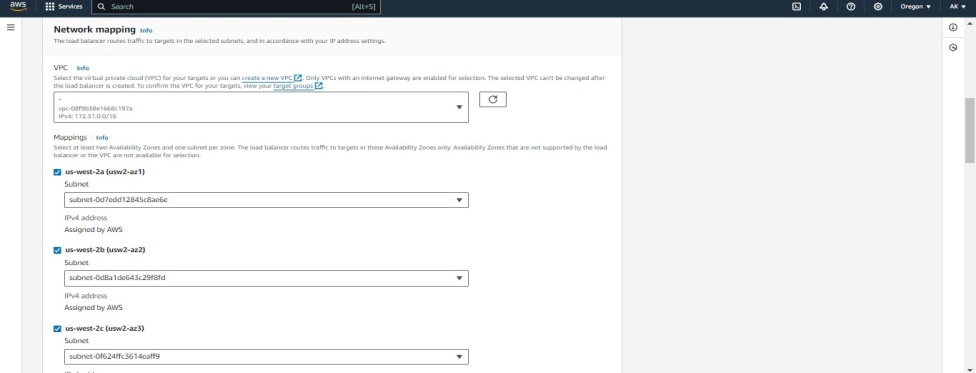


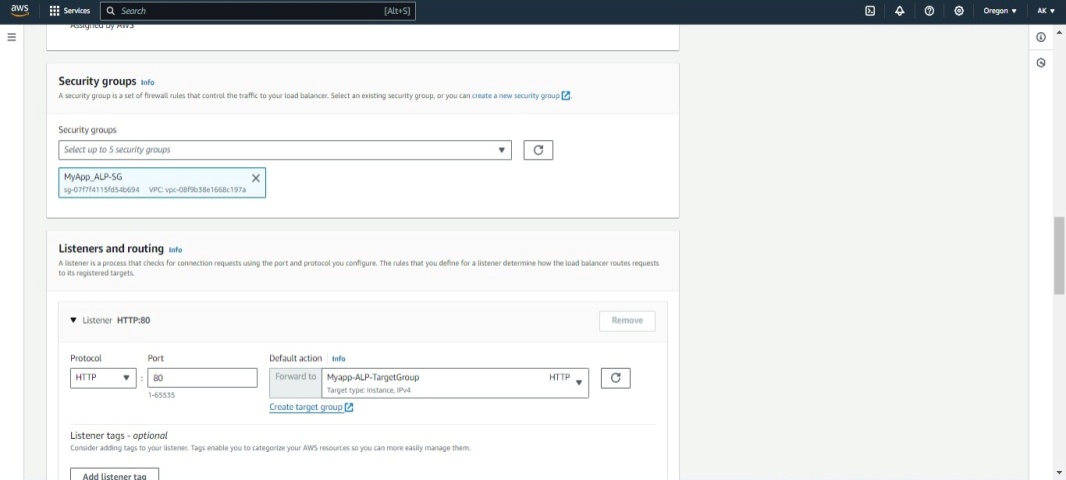
* Here select both the instances.

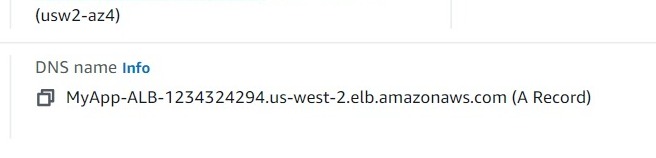




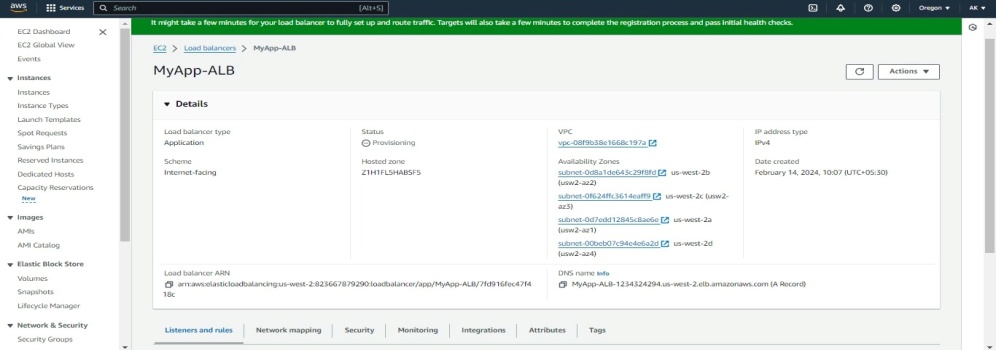




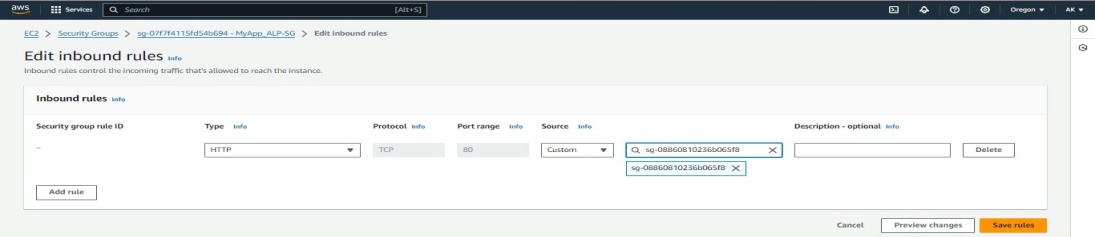


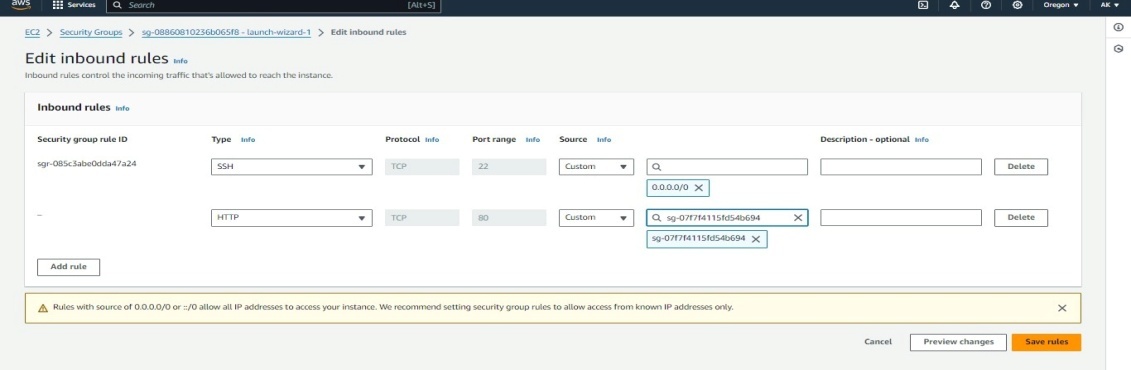


* Instead of accessing servers individually, now we can start accessing our servers through load balancers. So users can access these servers through load balancers. If any node goes down we won’t get issues, still we can access the application.
* By using this DNS Name, we can access our application.

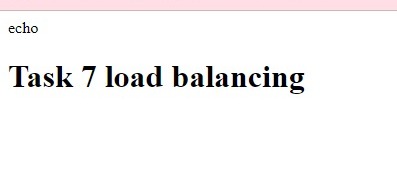


* Still we can access the servers from internet but we should not do this.
* In real time we need to access our application through load balancer only. We should not allow these applications from external IP’s.
* For that we can make some changes in the security group. Edit the inbound rules.





* Copy the DNS name and paste it in new tab. Finally I accessed my application through load balancer.



Now we can only access our application from load balancer, specific servers and specific port numbers.

In this way we can control our security as well. We should not accept any request from outside.