**Placement Empowerment Program**

**Cloud Computing and DevOps Centre**

**Set Up a Load Balancer in the Cloud**

**“*Configure a load balancer to distribute traffic across multiple VMs hosting your web application..*”**

Name: **DHARSHINI P** DEPARTMENT: **IT**



**Introduction**

In cloud computing, handling high traffic efficiently is crucial for ensuring application performance and availability. A **load balancer** helps distribute incoming traffic across multiple virtual machines (VMs), preventing overload on a single instance. This ensures high availability, fault tolerance, and improved responsiveness for web applications

**Overview**

A **load balancer** acts as an intermediary between users and backend servers, distributing requests to available instances based on predefined rules. Popular cloud providers like **AWS, Azure, and Google Cloud** offer managed load-balancing services, making it easy to set up and scale applications.

**Objective**

**The primary objectives of this POC are:**

* Deploy and configure a **cloud-based load balancer** to distribute traffic across multiple VMs.
* Improve the **scalability, reliability, and availability** of a web application.
* Ensure seamless traffic distribution, reducing server overload and improving performance.
* Implement **health checks** to monitor VM availability

**Important**

* **Ensures High Availability** – If one server fails, traffic is redirected to healthy instances.
* **Prevents Overload** – Balances traffic to avoid a single VM becoming a bottleneck.
* **Enhances Performance** – Optimizes response times by directing requests to less busy servers.
* **Supports Auto-Scaling** – Works with auto-scaling to dynamically adjust resources.
* **Improves Security** – Shields backend servers from direct exposure to users

**Steps to Set Up a Load Balancer**

**Step 1:**

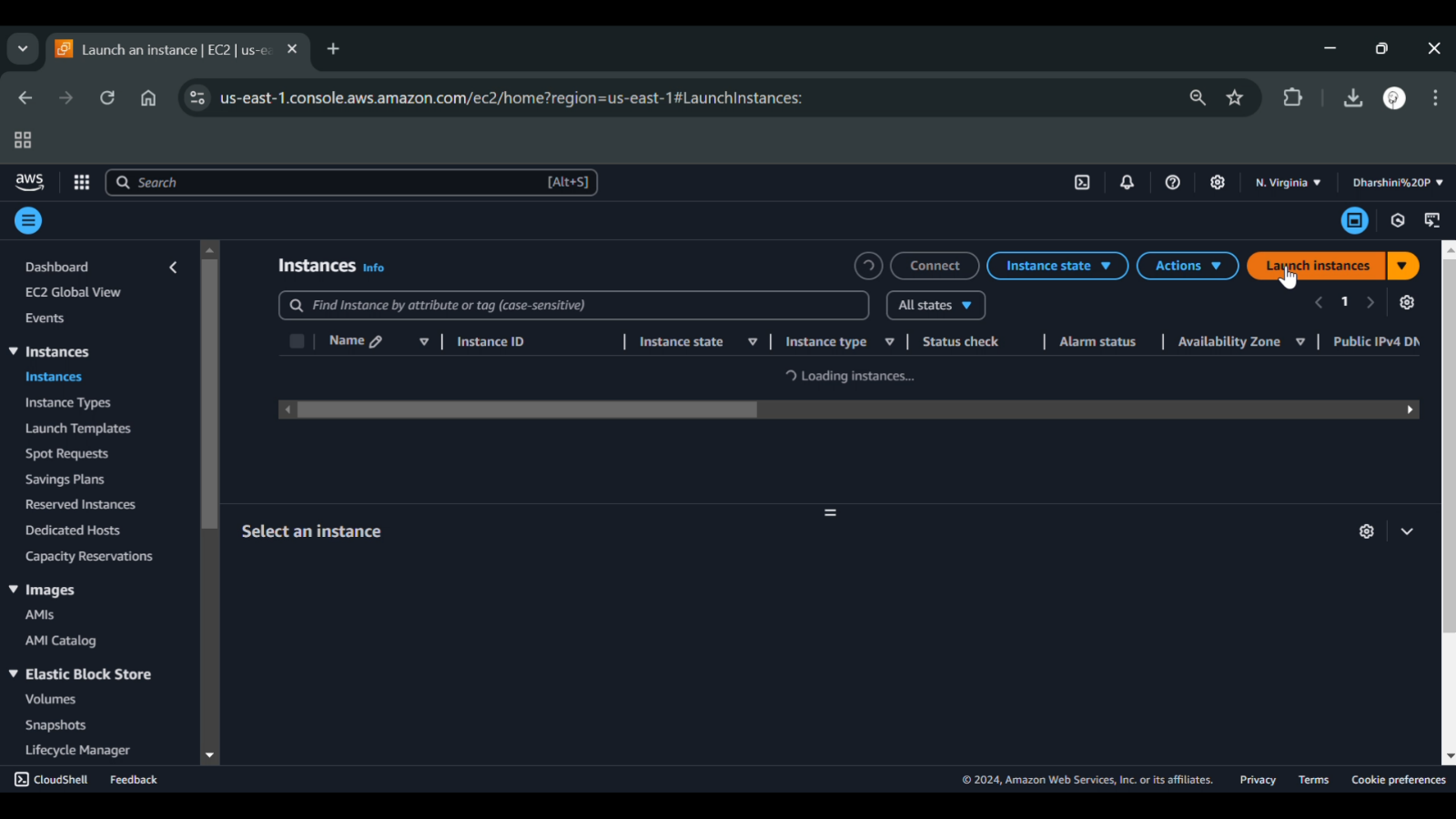
**Choose a Load Balancer Type**

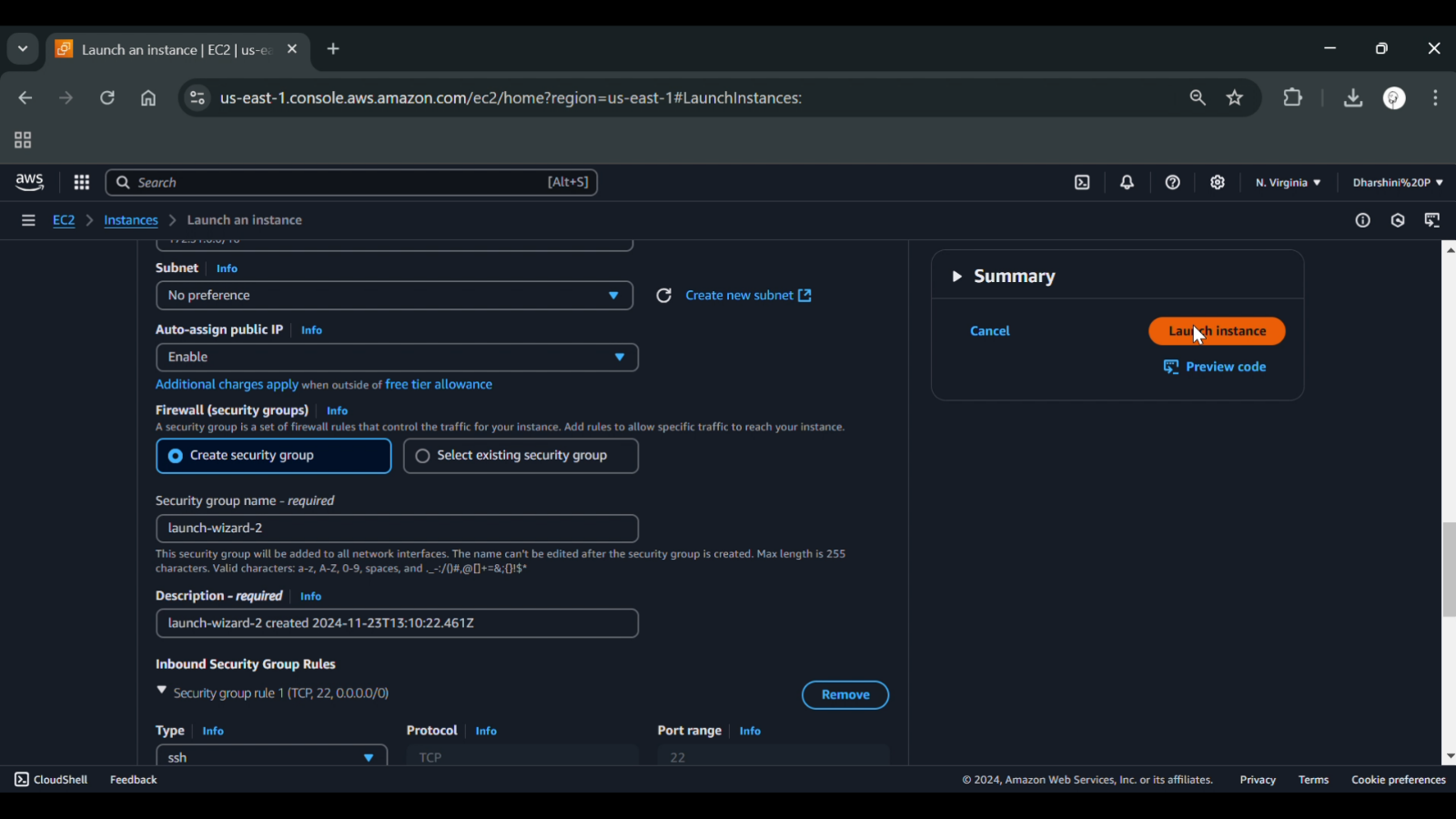
* AWS: Elastic Load Balancer (ELB)
* Azure: Azure Load Balancer or Application Gateway
* Google Cloud: Cloud Load Balancing

**Step 2:**

**Deploy Virtual Machines (VMs)**

* Launch multiple VMs with a web application installed.
* Ensure all instances are in the same region and network.

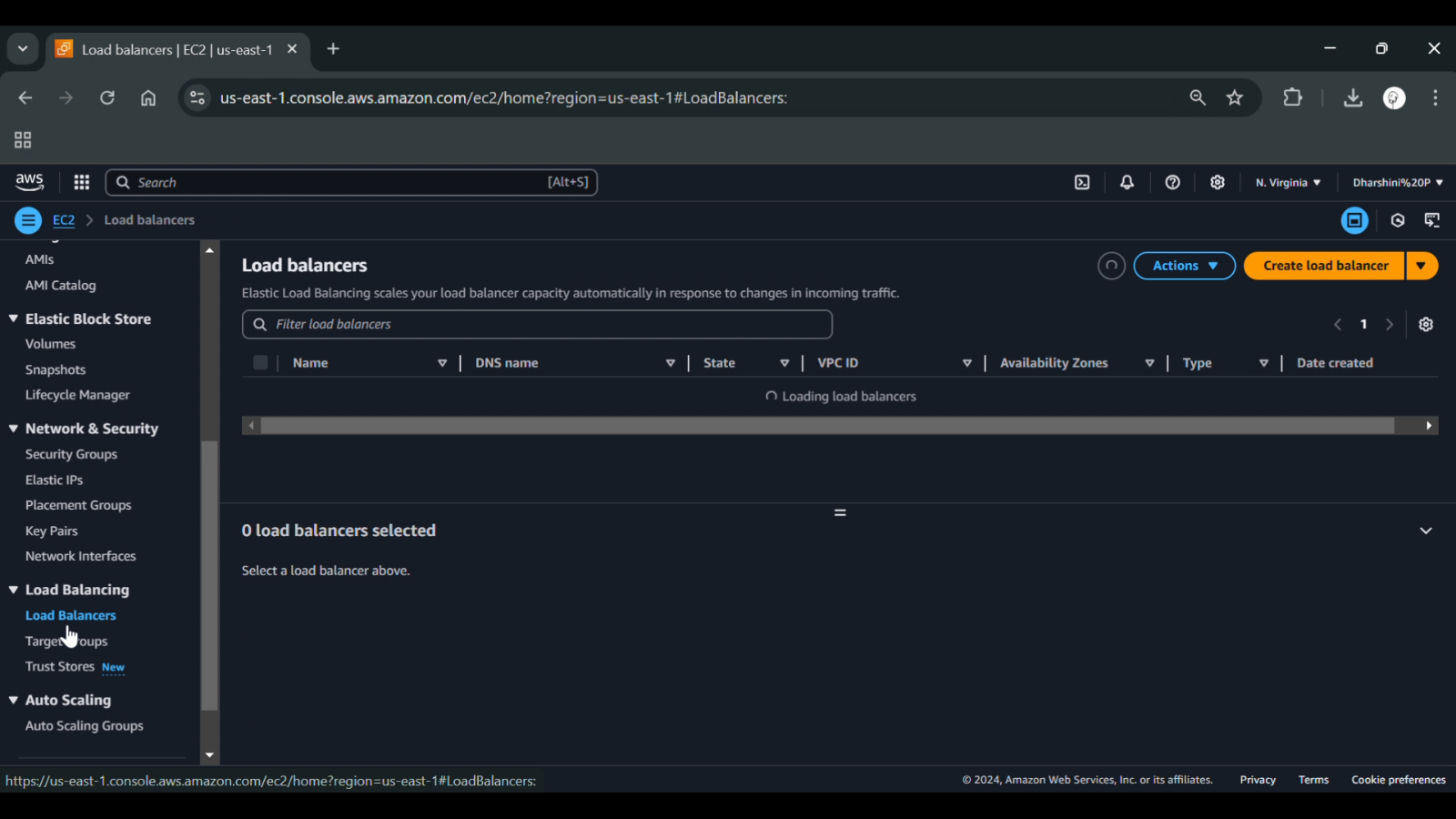


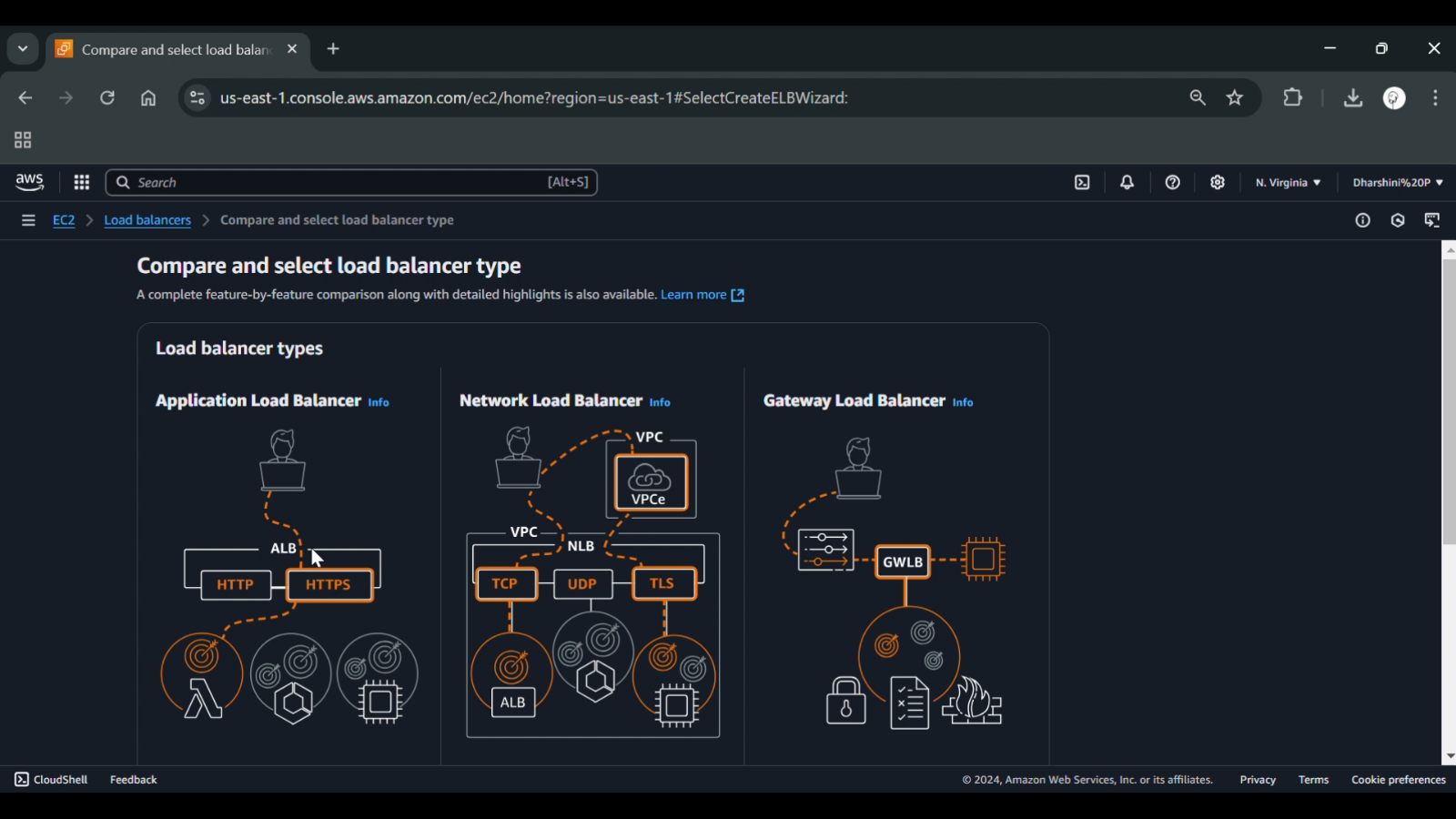


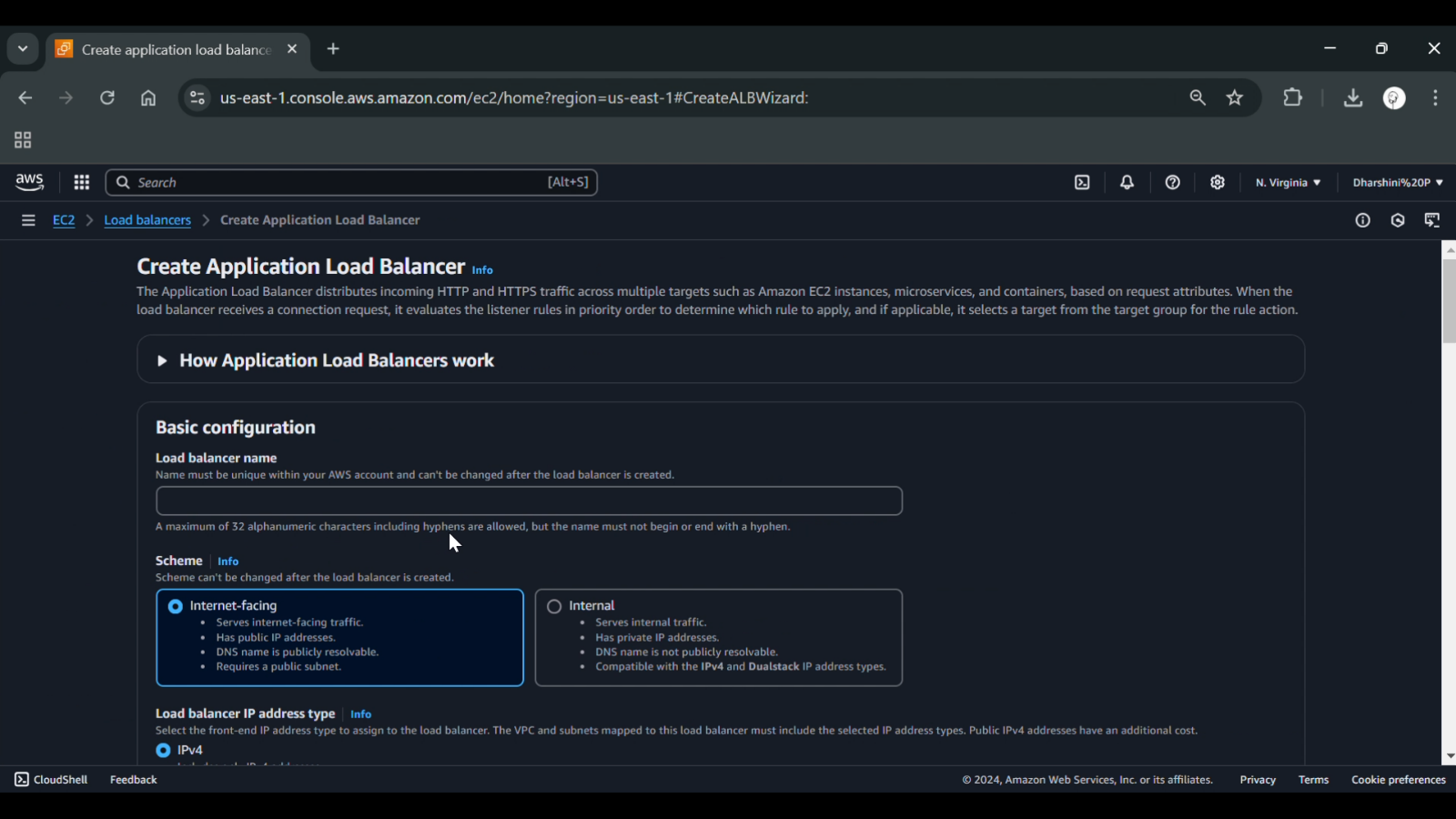
**Step 3:**

**Create a Load Balancer**

* AWS: Use Application Load Balancer (ALB) or Network Load Balancer (NLB).



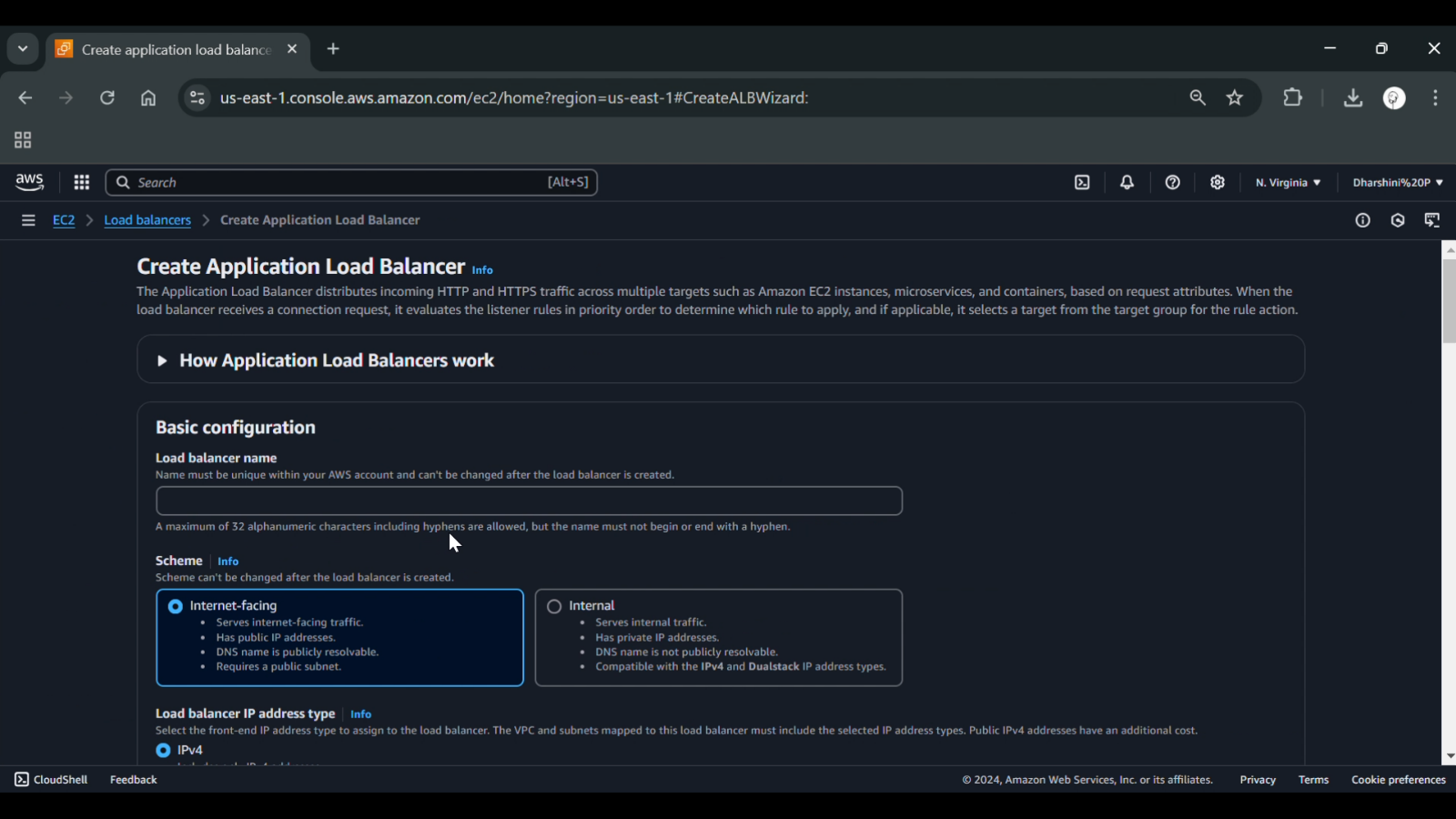


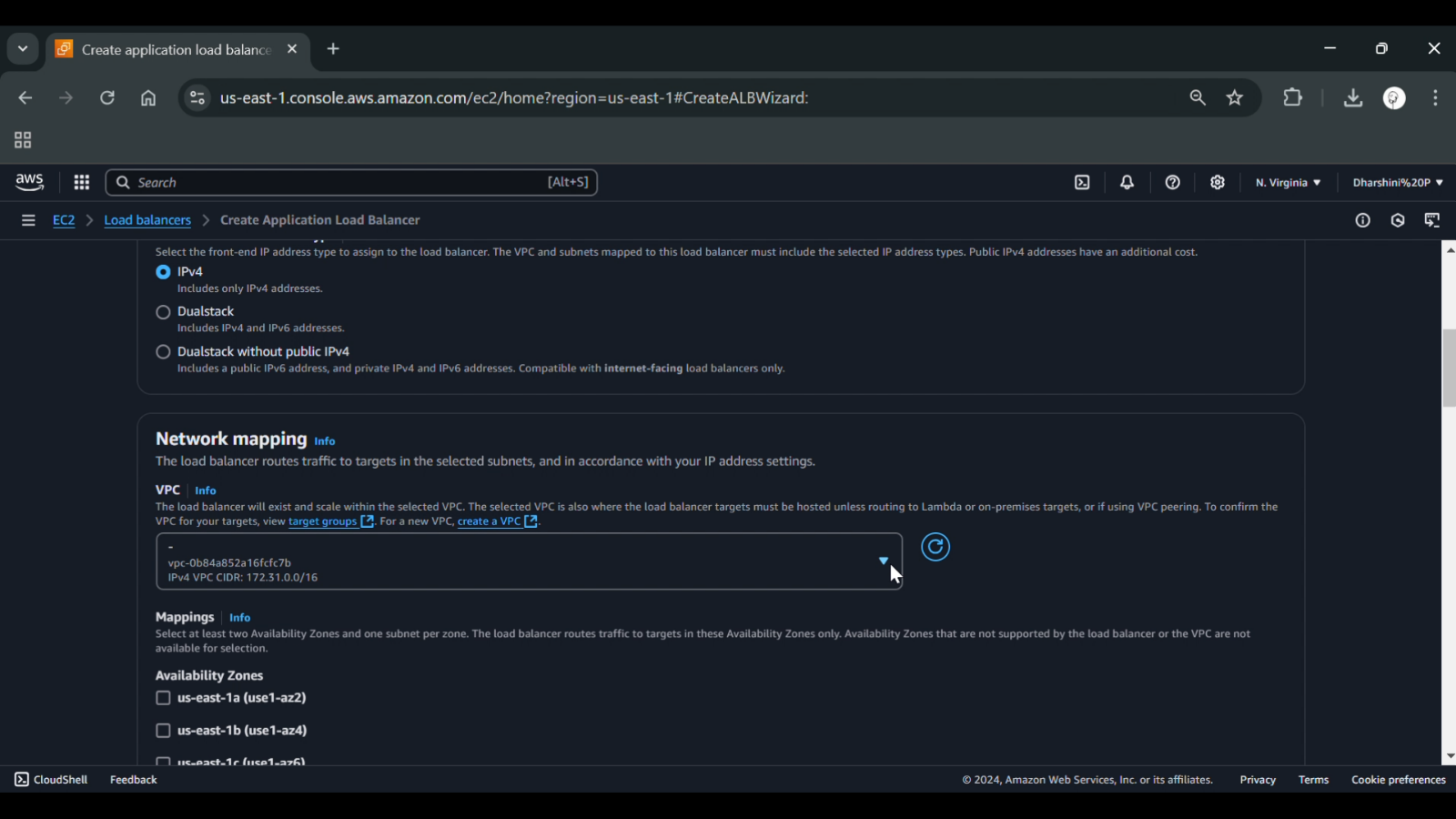


**Step 4:**

**Configure Backend Targets**

* Attach the VMs or compute instances to the load balancer.
* Define routing rules to distribute requests evenly.

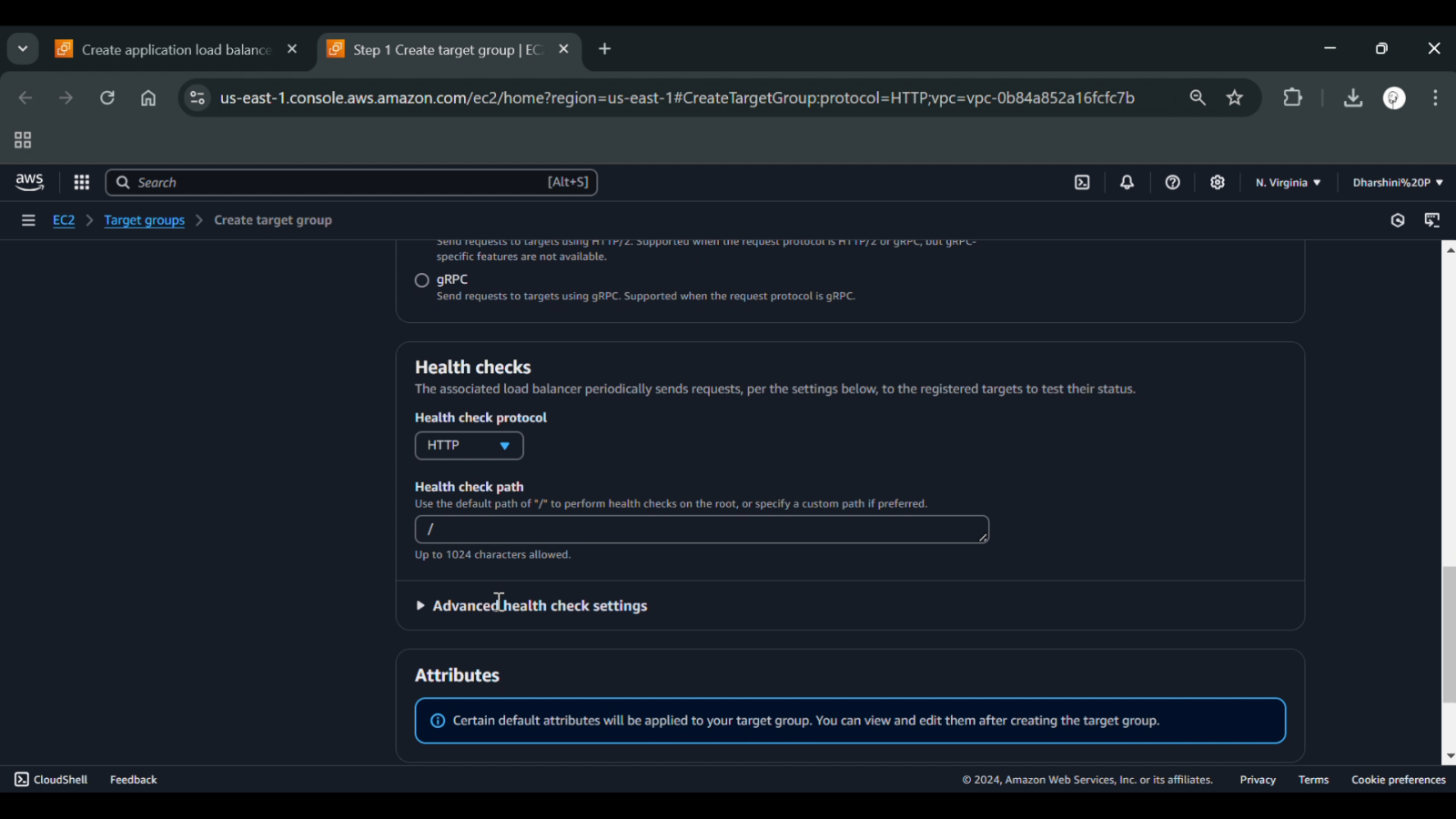




**Step 5:**

**Set Up Health Checks**

* Configure health checks to ensure traffic is only sent to healthy instances.
* Define health criteria such as response time and status codes.



**Step 6:**

**Update DNS and Test the Load Balancer**

* Map the domain name to the load balancer’s IP.
* Test by accessing the application and verifying traffic distribution.

**Expected Outcome:**

**By completing this POC, you will:**

* Evenly distributed traffic across multiple VMs, preventing overload.
* High availability with automatic failover if a server goes down.
* Faster response times due to optimized request routing.
* Seamless scalability by integrating with auto-scaling policies