# Step 1: Launch an EC2 Instance

## Log in to AWS Management Console:

* + Go to the AWS Management Console at https://aws.amazon.com/console/
  + Sign in with your AWS credentials.

## Navigate to EC2 Dashboard:

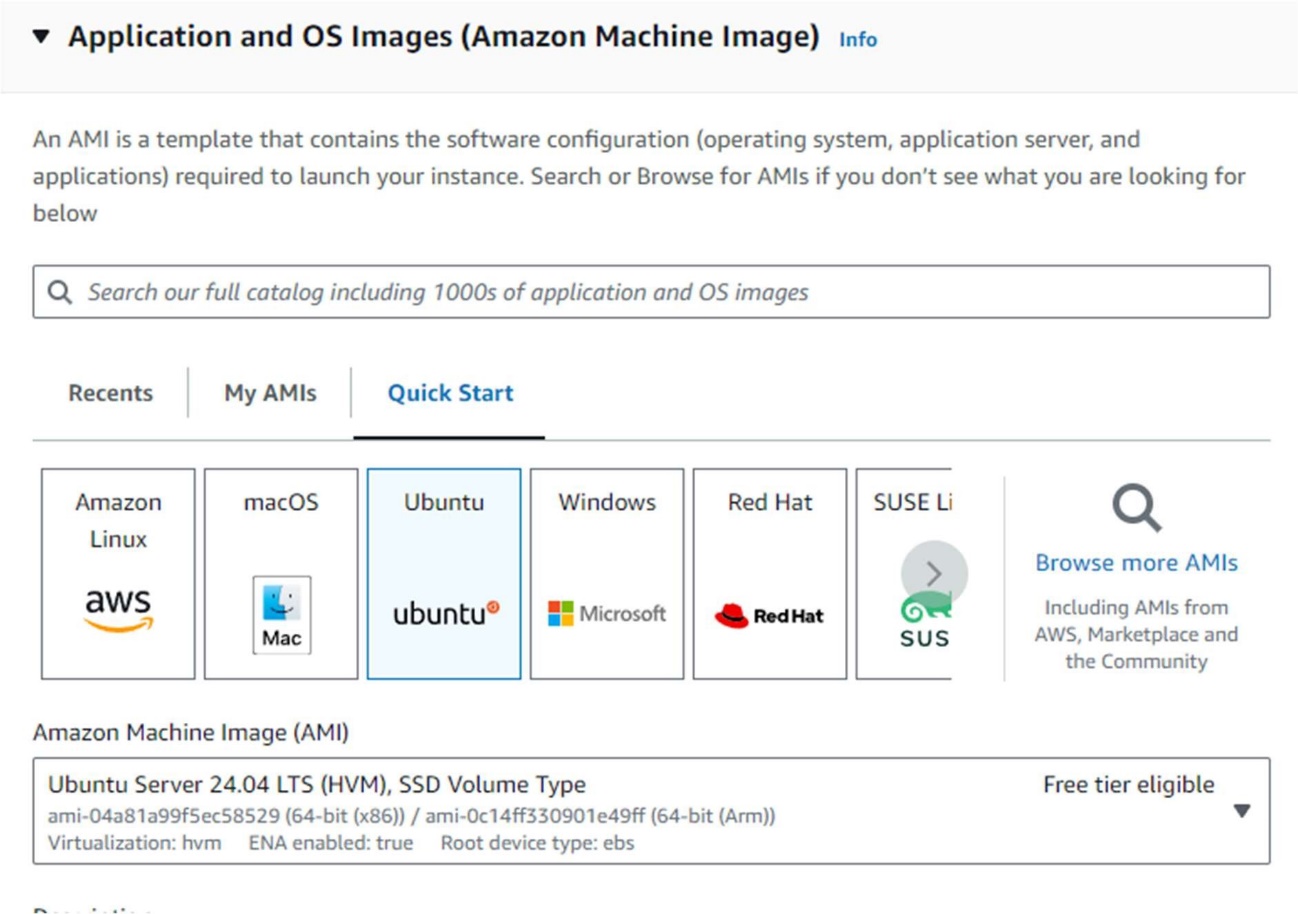
* + In the AWS Management Console, type "EC2" in the search bar and select EC2 to navigate to the EC2 Dashboard.

## Launch an Instance:

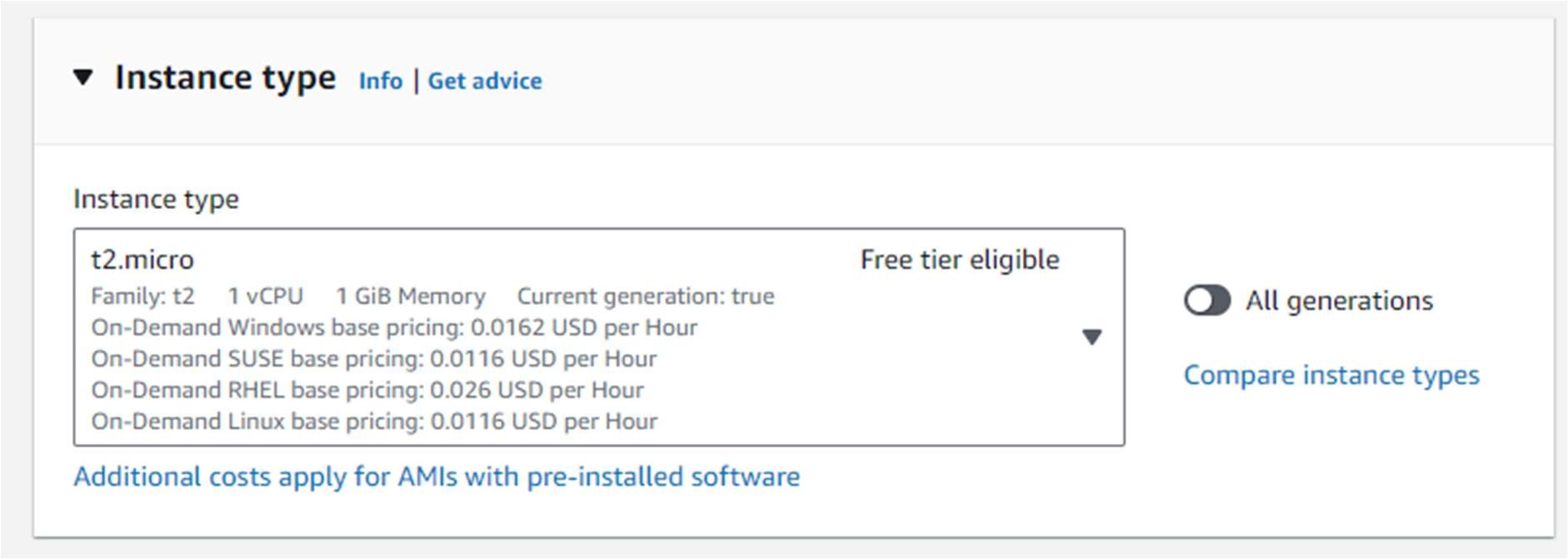
* + Click on the "Launch Instance" button.



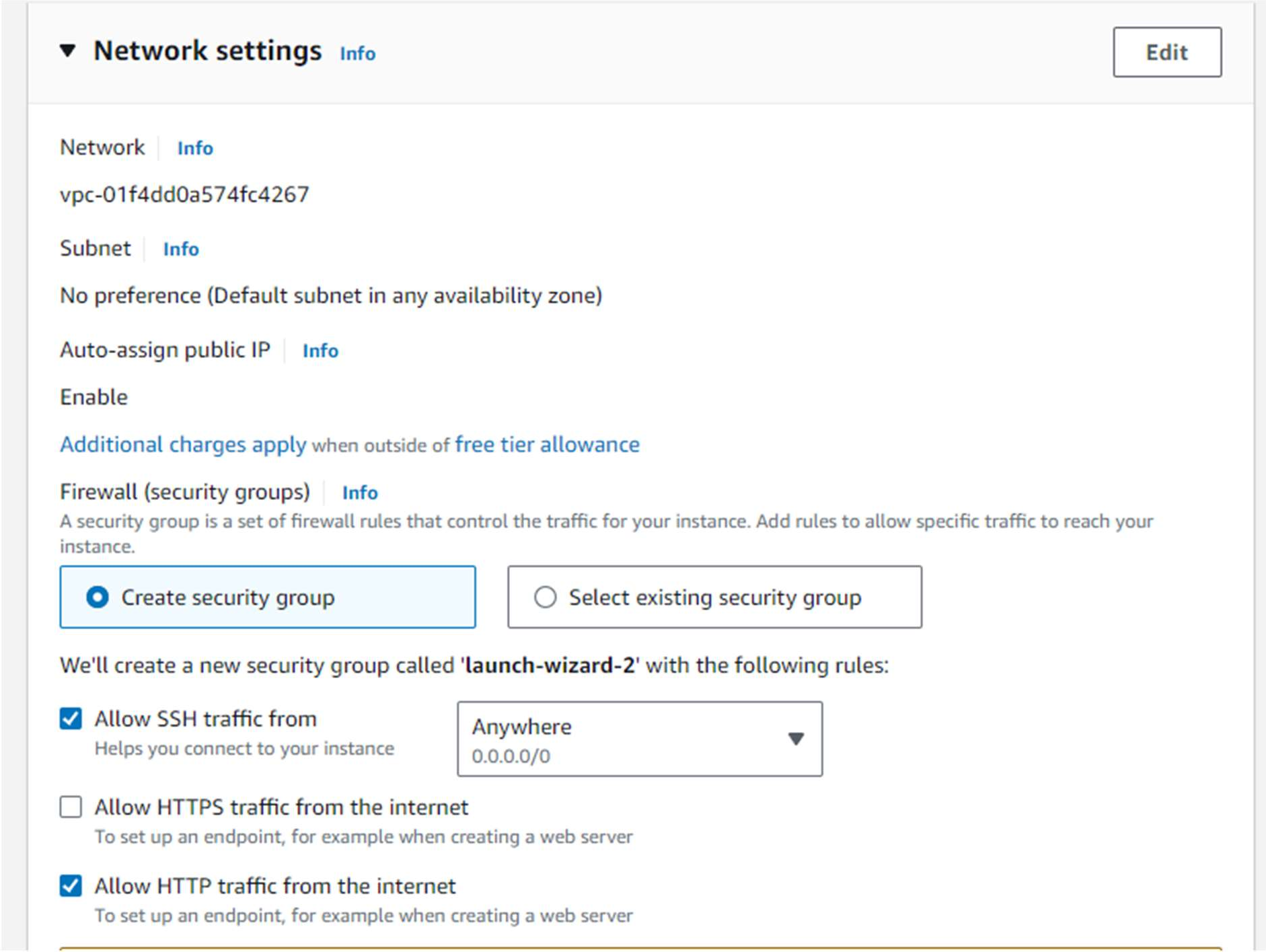
* + Choose an Amazon Machine Image (AMI): Select "Ubuntu Server 20.04 LTS (HVM), SSD Volume Type".



* + Choose an Instance Type: Select t2.micro (eligible for the free tier).



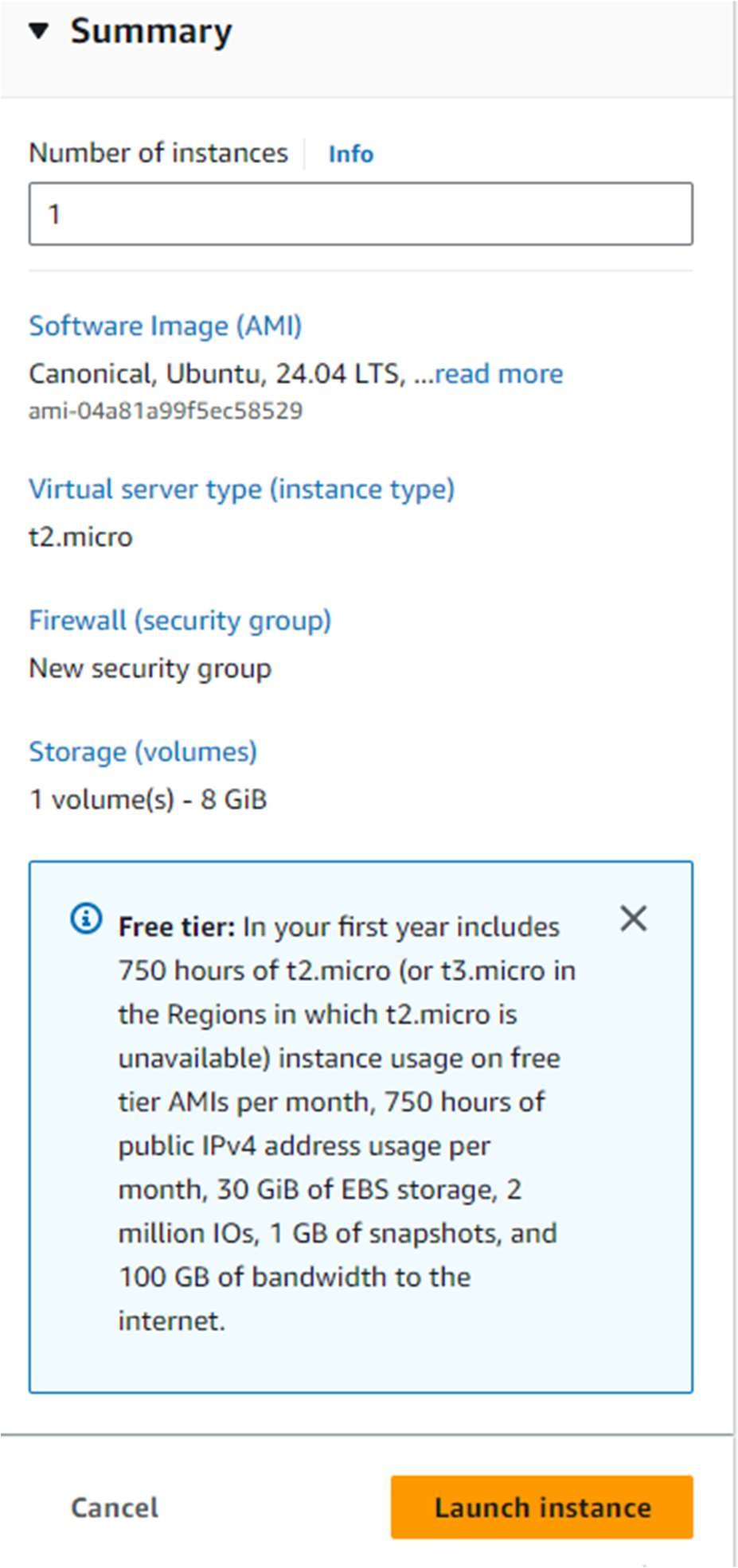
* + Configure Instance:
    - Select an existing key pair or create a new one.
    - Network: Choose the default VPC.
    - Subnet: Choose a subnet in the US-East-1 (N. Virginia) region.
    - Enable Auto-assign Public IP.



* + Add Storage: Keep the default settings.
  + Add Tags: Add a tag to identify your instance (e.g., Key: Name, Value: Nginx).

## Review and Launch:

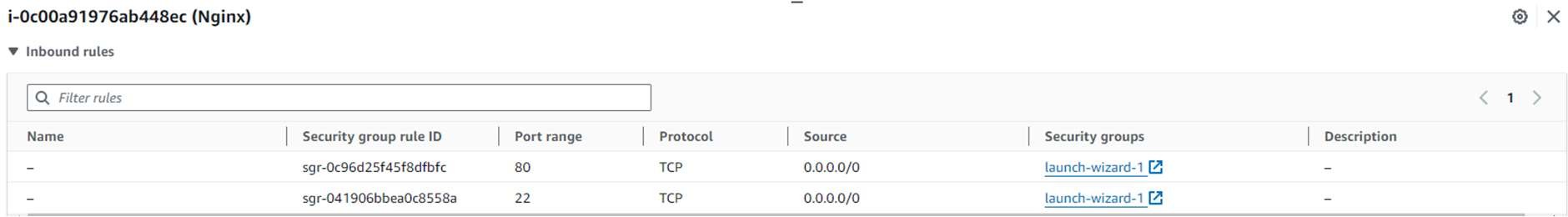
* + Review your instance settings and click "Launch".





## Configure Security Group:

* + Add a new security group with the following rules:
    - Type: HTTP, Protocol: TCP, Port Range: 80, Source: 0.0.0.0/0
    - Type: SSH, Protocol: TCP, Port Range: 22, Source: 0.0.0.0/0



# Step 2: Connect to Your Instance

## Connect to the EC2 Instance:

* + In the EC2 Dashboard, select your instance.
  + Click on "Connect" and follow the instructions to connect to your instance using SSH.

# Step 3: Install Nginx

## Update Package List:

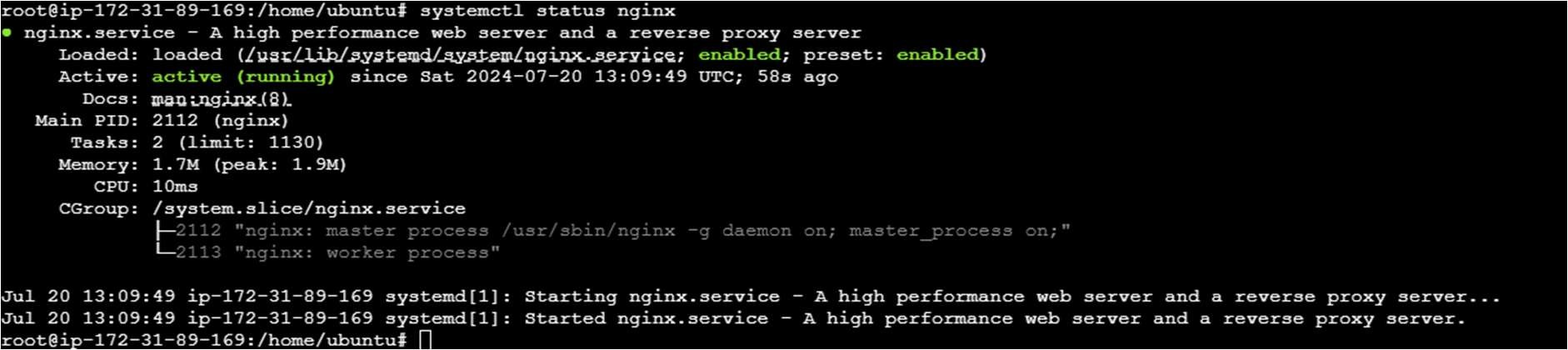
sudo apt update

## Install Nginx:

sudo apt install nginx -y

## Start and Enable Nginx:

sudo systemctl start nginx sudo systemctl enable nginx sudo systemctl status nginx



# Step 4: Configure Nginx to Display "Hello World"

## Modify the Default Nginx Webpage:

* + Open the default Nginx configuration file:

sudo nano /var/www/html/index.nginx-debian.html

* + Replace the content with the following HTML:

<!DOCTYPE html>

<html>

<head>

<title>Welcome to Nginx!</title>

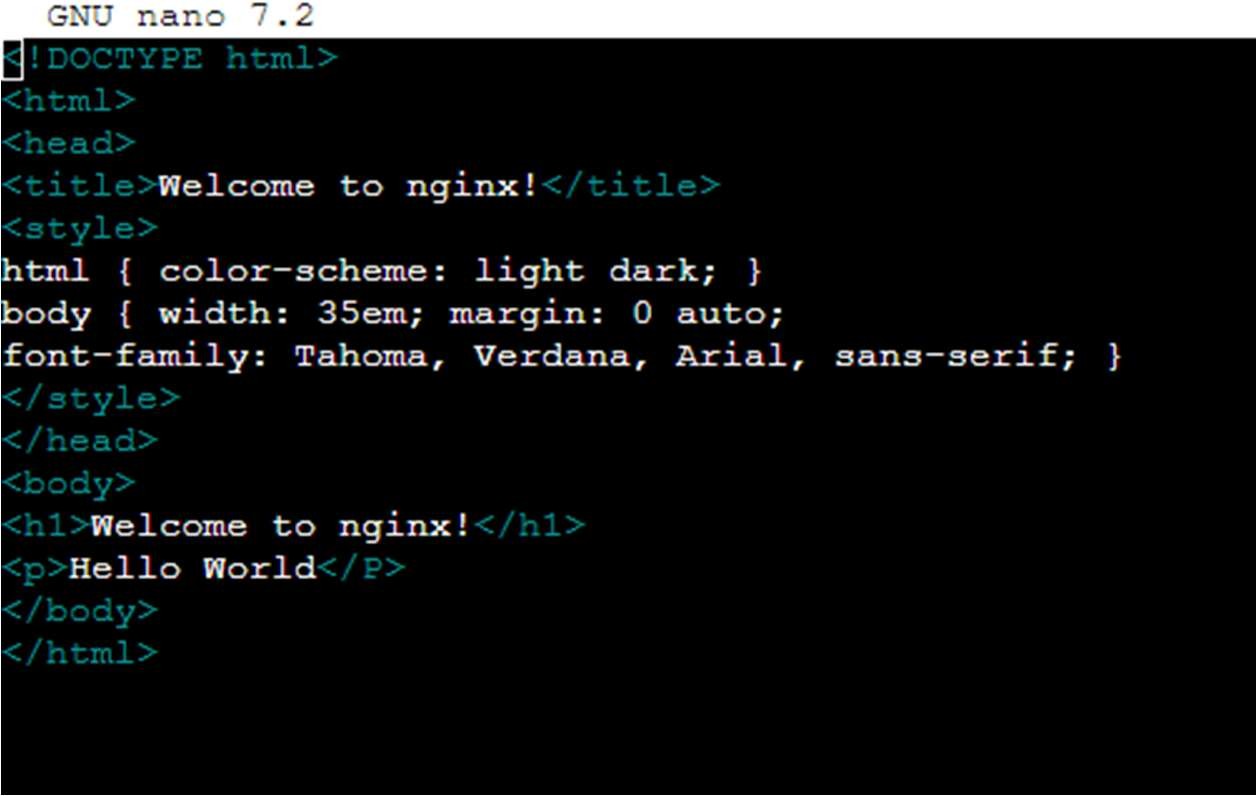
</head>

<body>

<h1>Hello World</h1>

</body>

</html>



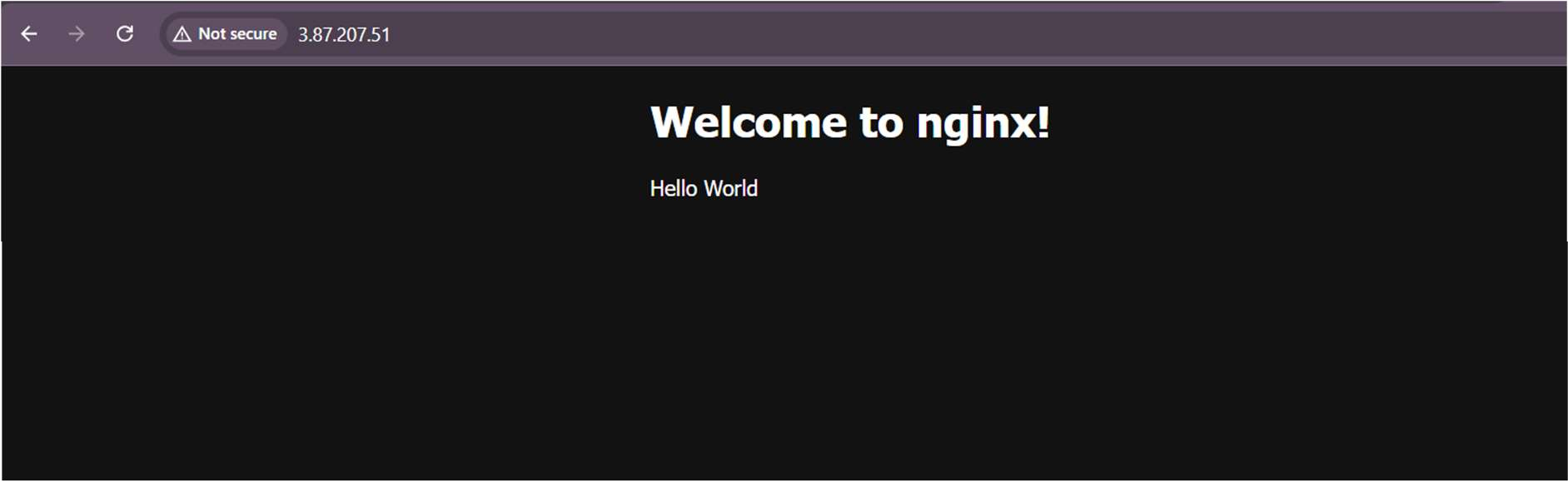
## Save and Close the File:

* + Press Ctrl + X to close the file.
  + Press Y to confirm changes, then press Enter.

# Step 5: Verify the Configuration

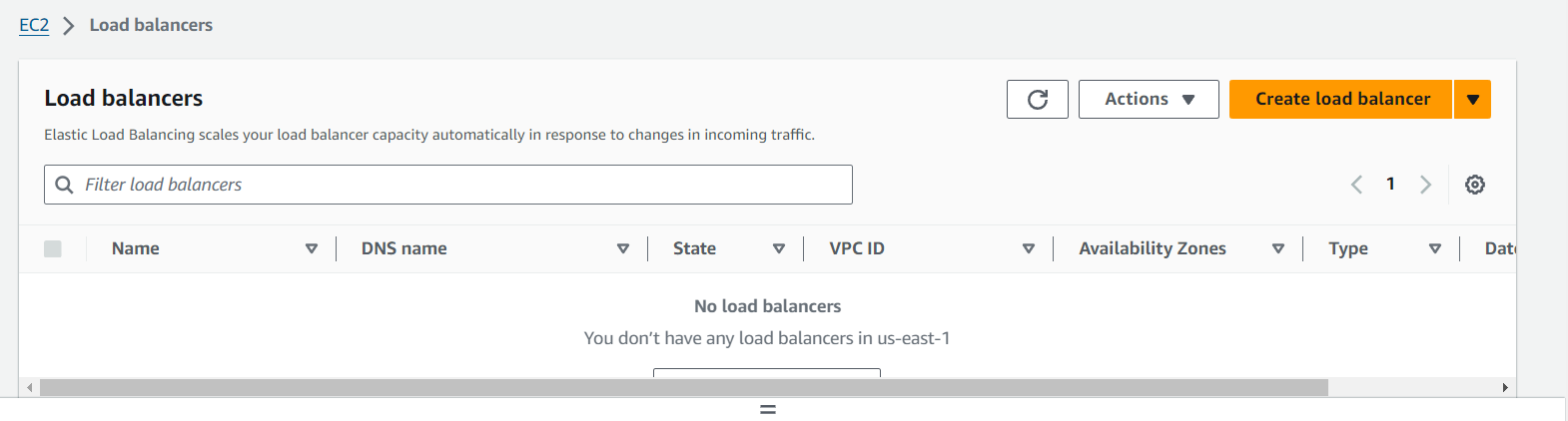
## Open a Web Browser:

* + Enter the public IP address of your EC2 instance in the address bar.
  + You should see a webpage displaying the message: "Hello World".

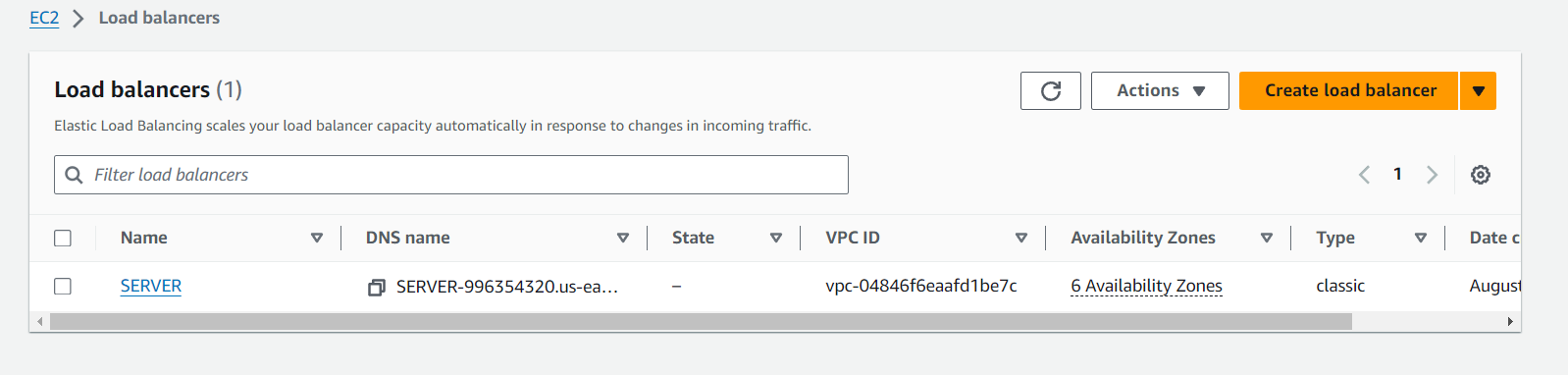


# Step 6: Create a Classic Load Balancer

1. **Go to the EC2 Dashboard.**
2. In the left navigation pane, under **Load Balancing**, select **Load Balancers**.

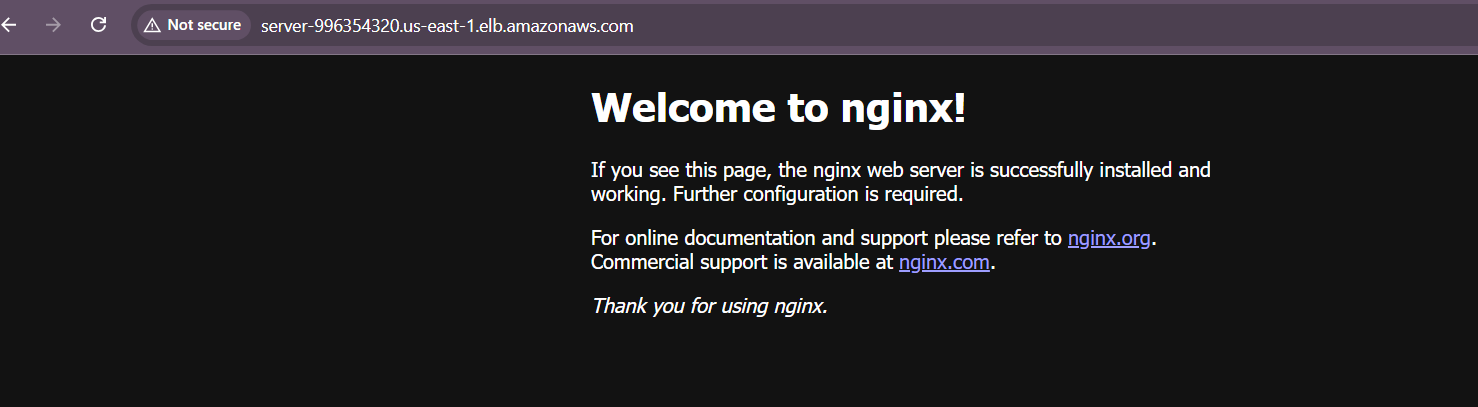


1. Click on **Create Load Balancer** and select **Classic Load Balancer**.
2. **Configure the Load Balancer**:
   * Name your load balancer.
   * Select the VPC and availability zones.
   * Configure listeners (e.g., HTTP on port 80).
   * Configure health checks (e.g., HTTP on /).
   * Select the instances you launched earlier to register with the load balancer.
3. **Review and create the load balancer.**



**Step 4: Test the Classic Load Balancer**

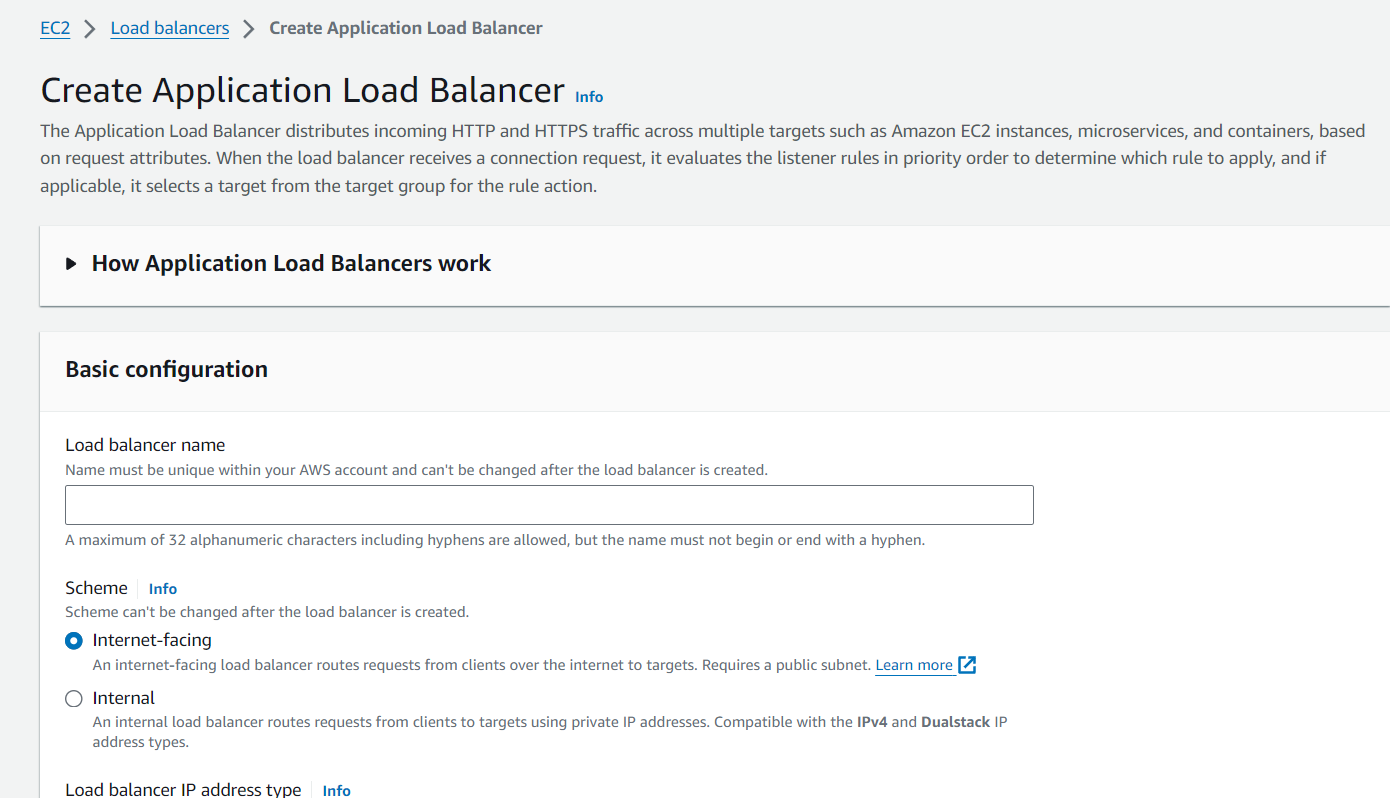
1. **Get the DNS name of the CLB** from the Load Balancers dashboard.
2. Access the DNS name in your web browser to see the load balancing in action.



**Part 2: Migrate to an Application Load Balancer**

**Step 1: Create an Application Load Balancer**

1. **Go to the EC2 Dashboard.**
2. In the left navigation pane, under **Load Balancing**, select **Load Balancers**.
3. Click on **Create Load Balancer** and select **Application Load Balancer**.

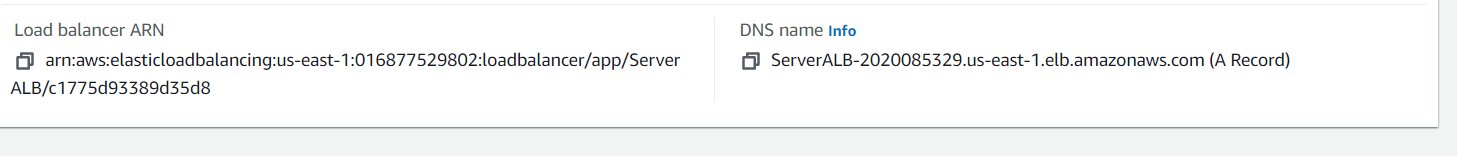


**Step 2: Create a Load Balancer**

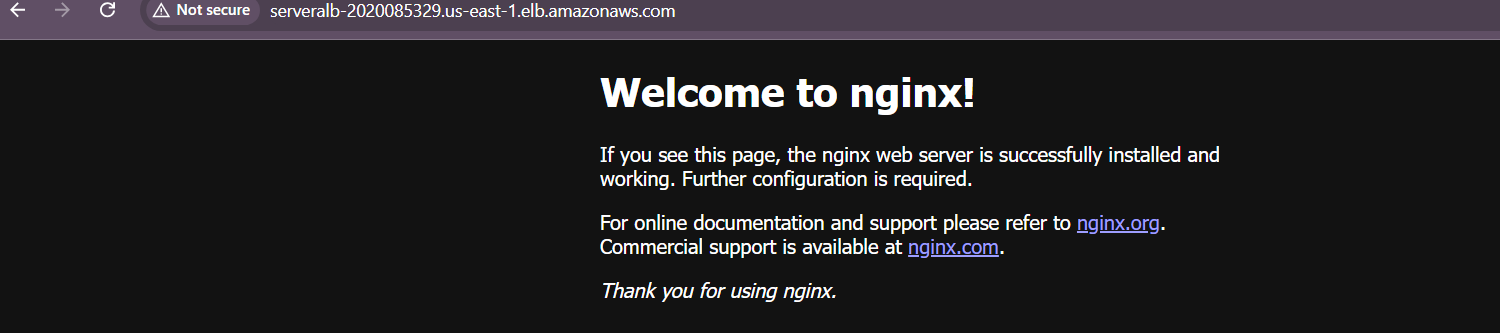
1. **Navigate to the EC2 Dashboard**:
   * Click on **Load Balancers** under the Load Balancing section.
   * Click on **Create Load Balancer**.
   * Choose **Application Load Balancer**.
   * Configure the load balancer:
     + Name: my-load-balancer.
     + Scheme: Internet-facing.
     + Listeners: HTTP (port 80).
     + Availability Zones: Select the VPC and subnets.
2. **Configure Security Groups** for the load balancer:
   * Ensure it allows HTTP traffic.
3. **Configure Routing**:
   * Create a target group:
     + Name: my-target-group.
     + Target type: Instances.
     + Protocol: HTTP.
     + Port: 80.
     + Health checks: HTTP.
   * Register your instances in the target group.
4. **Configure Routing**:
   * Create a new target group.
   * Select Instance as the target type.
   * Register the same EC2 instances you used with the CLB.
5. **Review and Create** the load balancer.

**Step 4: Test the Application Load Balancer**

1. **Get the DNS name of the ALB** from the Load Balancers dashboard.



1. Access the DNS name in your web browser to ensure the ALB is routing traffic correctly to your instances.



**Step 5: (Optional) Delete the Classic Load Balancer**

* Once you have confirmed the ALB is working correctly, you can delete the Classic Load Balancer if it is no longer needed.

**Note**

* Ensure that your security groups allow inbound traffic on port 80 for the ALB and instances.

This process will set up a Classic Load Balancer with registered EC2 instances and then migrate to an Application Load Balancer while ensuring the web pages remain accessible.