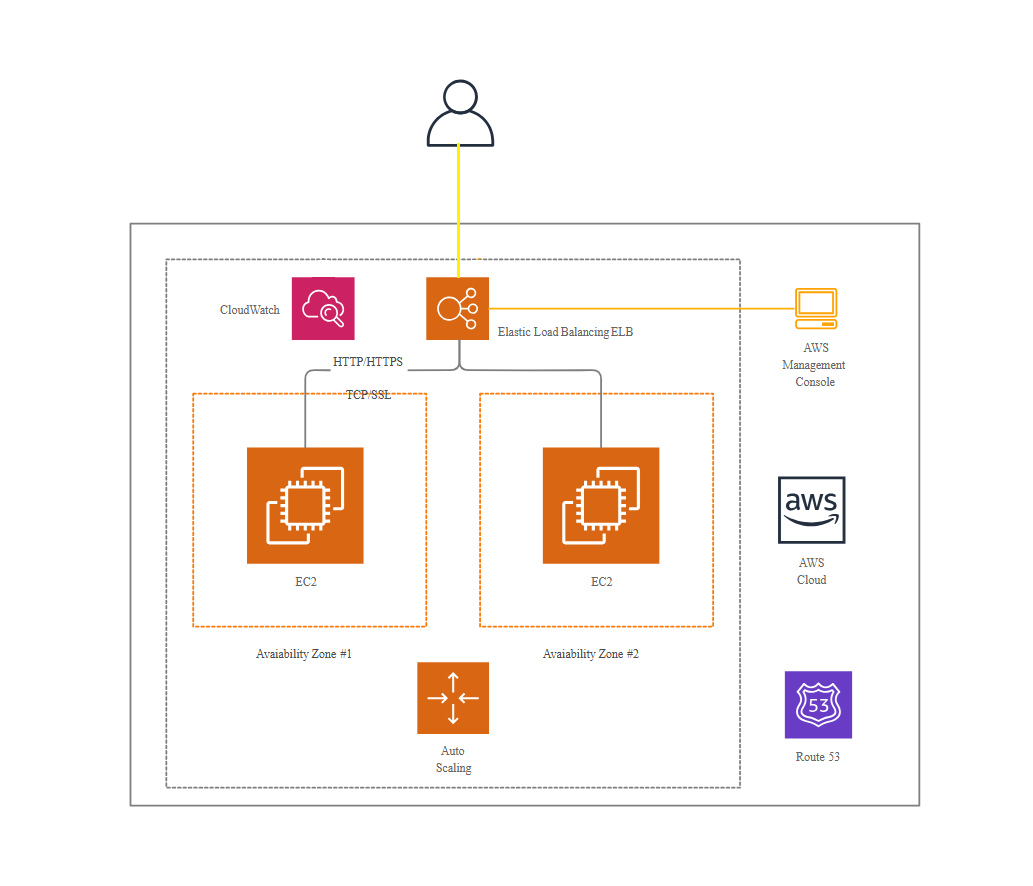
**The company intends to deploy an e-commerce website on the AWS cloud with high availability and auto-scaling to handle unpredictable traffic. Additionally, the company plans to offer a festival discount, which may require scaling resources to accommodate the increased workload. your task is to build and deploy the application while also architecting the cloud infrastructure. You are expected to utilize primary resources, follow AWS best practices, and incorporate cost optimization strategies.**

**Project Requirements**

* **Operating System and Web Server**
* **Use AWS Linux Ubuntu 22.04 or 24.04 as the operating system.**
* **Install and configure a web server Apache**.

**AWS Services**

* **Implement an Auto-Scaling Group to ensure scalability and handle traffic fluctuations. ). DB is not Required**
* **Utilize an Application Load Balancer for efficient traffic distribution across instances.**
* **Instance Type**
* **For demonstration purposes, use free-tier eligible instances, such as t2.micro or t3.micro (availability depends on the AWS region)**

****

**Repository Setup**

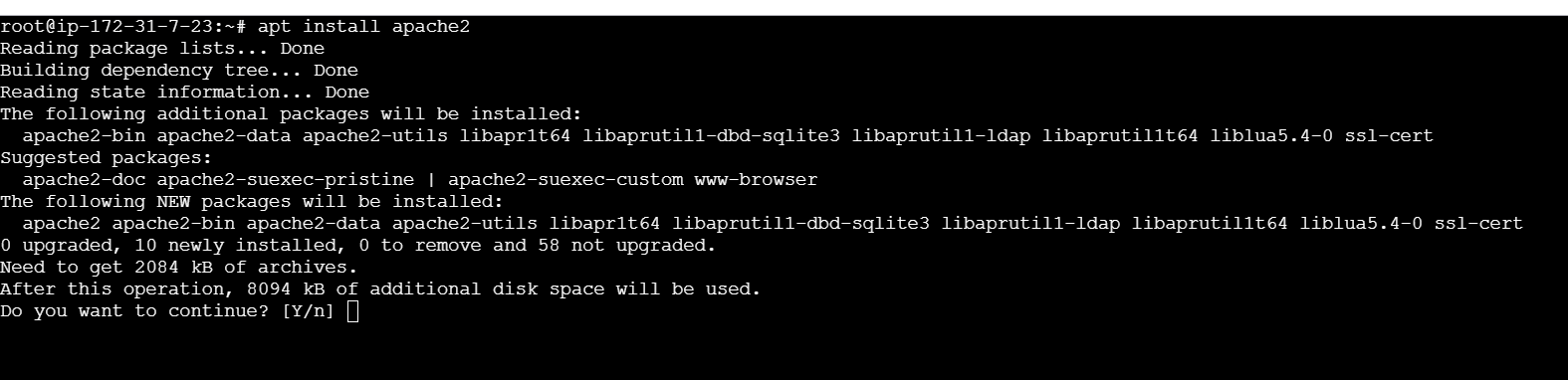
* **Clone the repository (details below) into the web server's root folder.**
* **This repository is public and can be cloned anytime.**
* **The codebase is written in HTML and JavaScript and requires no additional dependencies to install or configure.**
* **Ensure the repository is cloned into the /var/www/ directory of the webserver to function properly on Linux machines.**

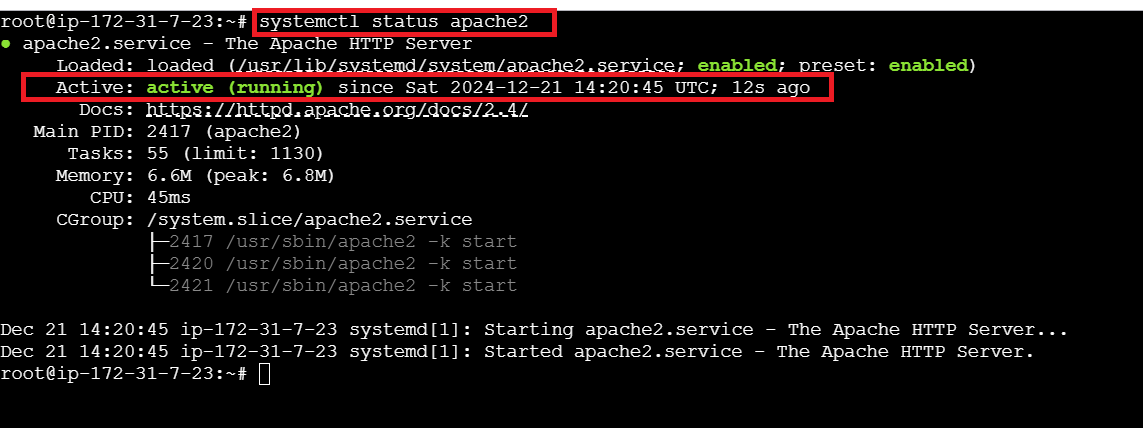
Repository URL:

<https://github.com/SHRIDHARMUDASHI/E-Commerswebsite.git>

**Here are the complete details about building and running the project.**

**Launch an Instance of Linux Ubuntu and Install the Webserver**

****

****

**Change the webserver directory to clone the repository**

**cd /var/www**

**A black and white text

Description automatically generated**

**clone the repository**   
  
git clone <https://github.com/SHRIDHARMUDASHI/E-Commerswebsite.git>

**A computer screen with white text

Description automatically generated**

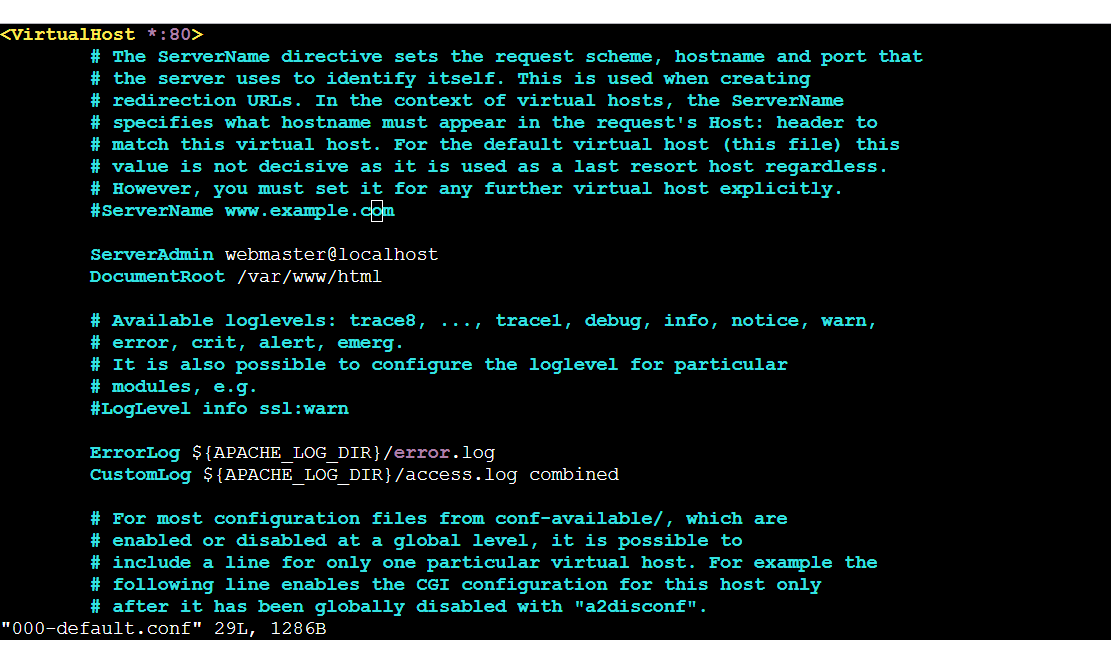
**Check you have cloned the code properly using ll command  
A computer screen with white text

Description automatically generated**

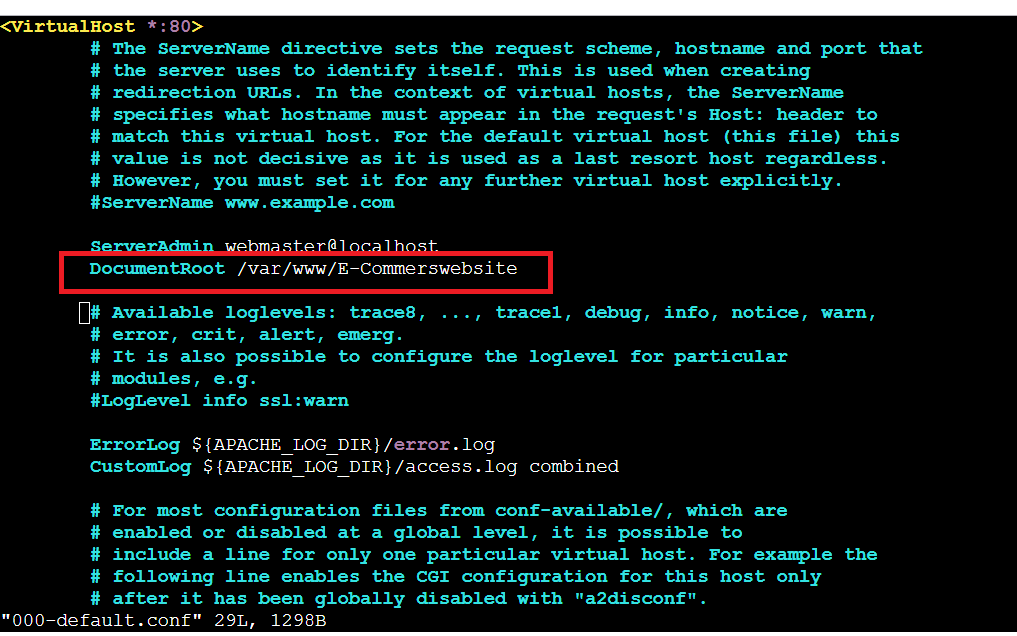
**Change the default directory of the web server to the cloned repository to run the application.  
  
cd /etc/apache2/sites-available/ and press ll command and check  
A screen shot of a computer

Description automatically generated**

**Open the 000-default.conf file using vi or nano I will be using vi**

****

**Switch to insert mode by pressing the I key on the keyboard and change the DocumentRoot from /var/www/html to /var/www/E-Commerswebsite.**



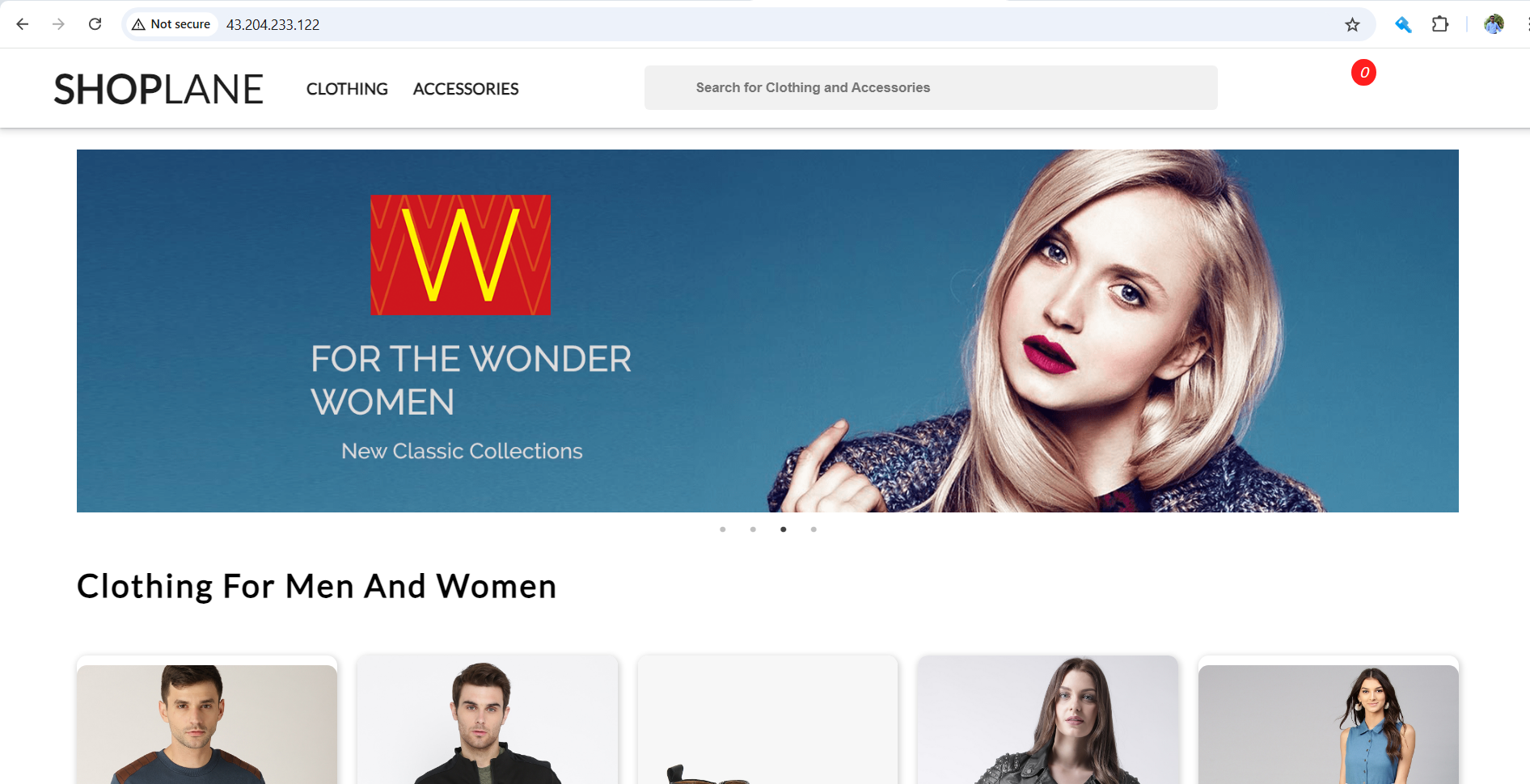
**Save and exit by pressing the escape :wq**

**Restart the webserver systemctl restart apache2**

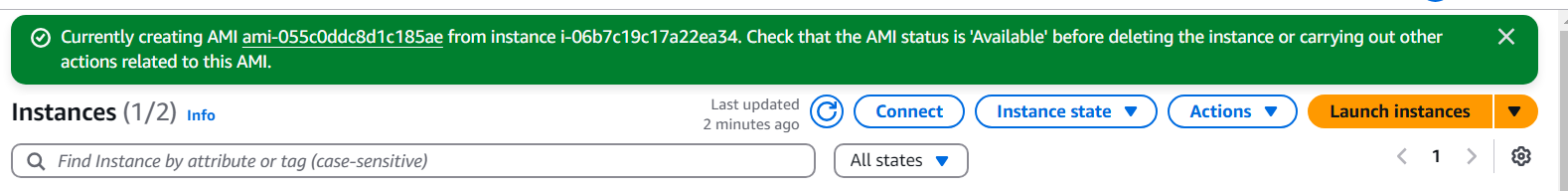
****

**Restart the webserver systemctl restart apache2**

**Check the application by accessing the public IP or Public DNS in your browser.**

****

**Once you have verified that the application is running properly, create an AMI to use as the template for the Auto-Scaling Group.**

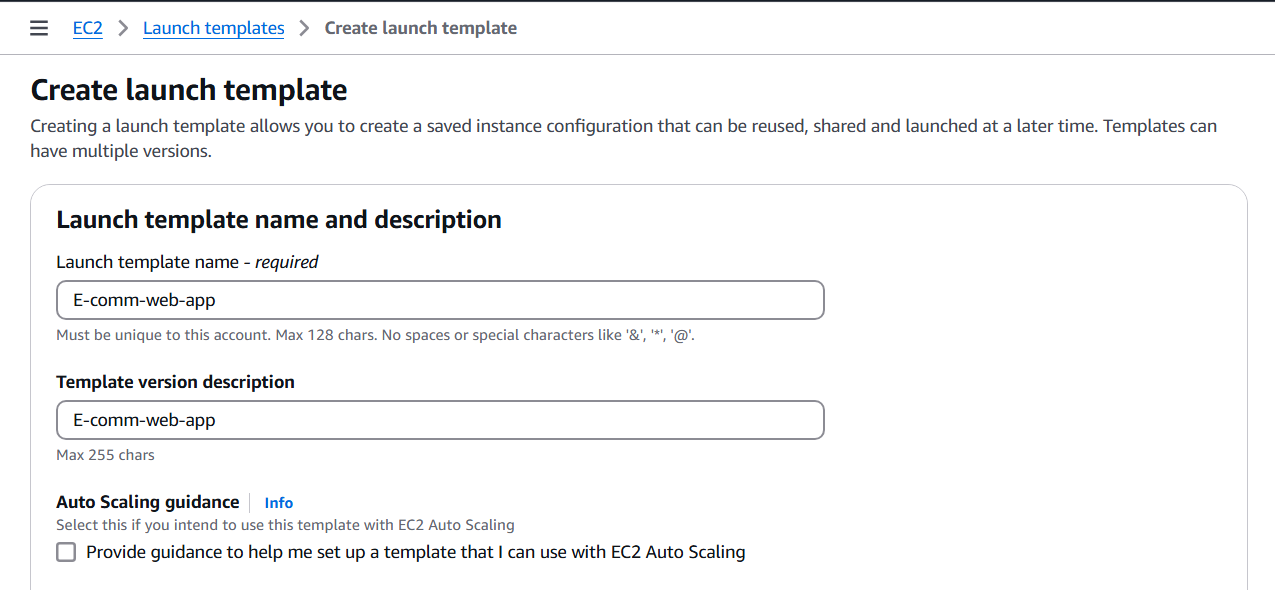
****

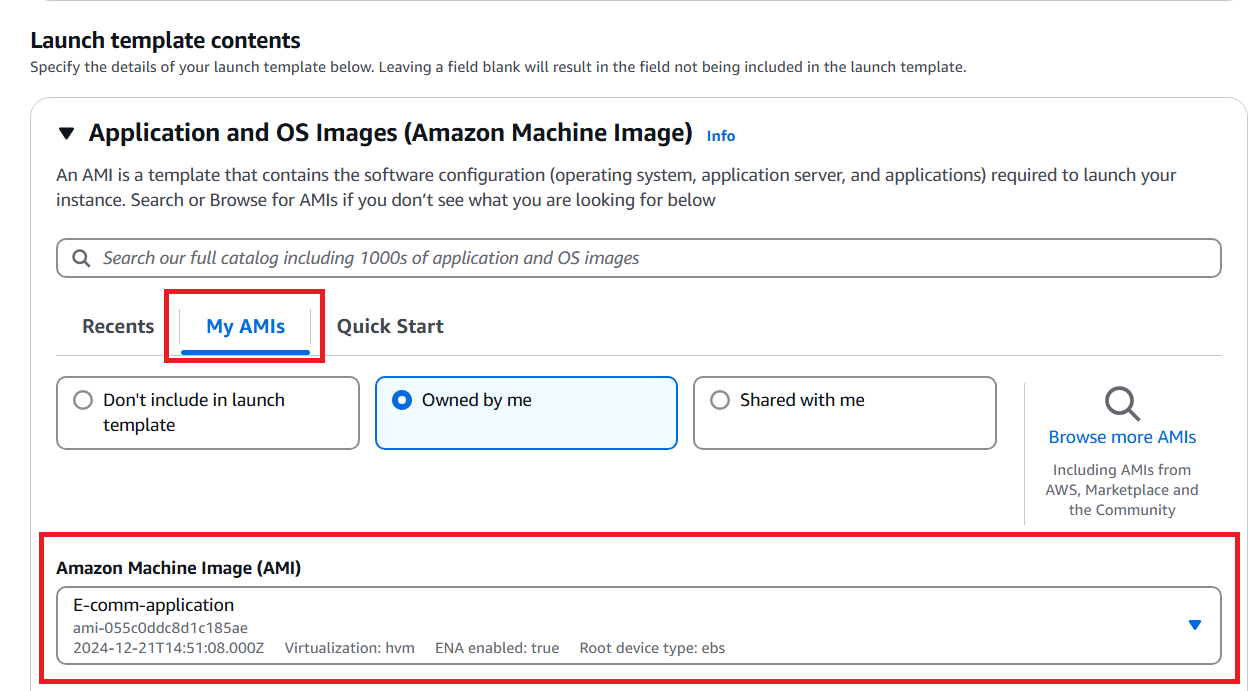
**Once the AMI has been created and is available, proceed to create the launch template for the Auto-Scaling Group.**

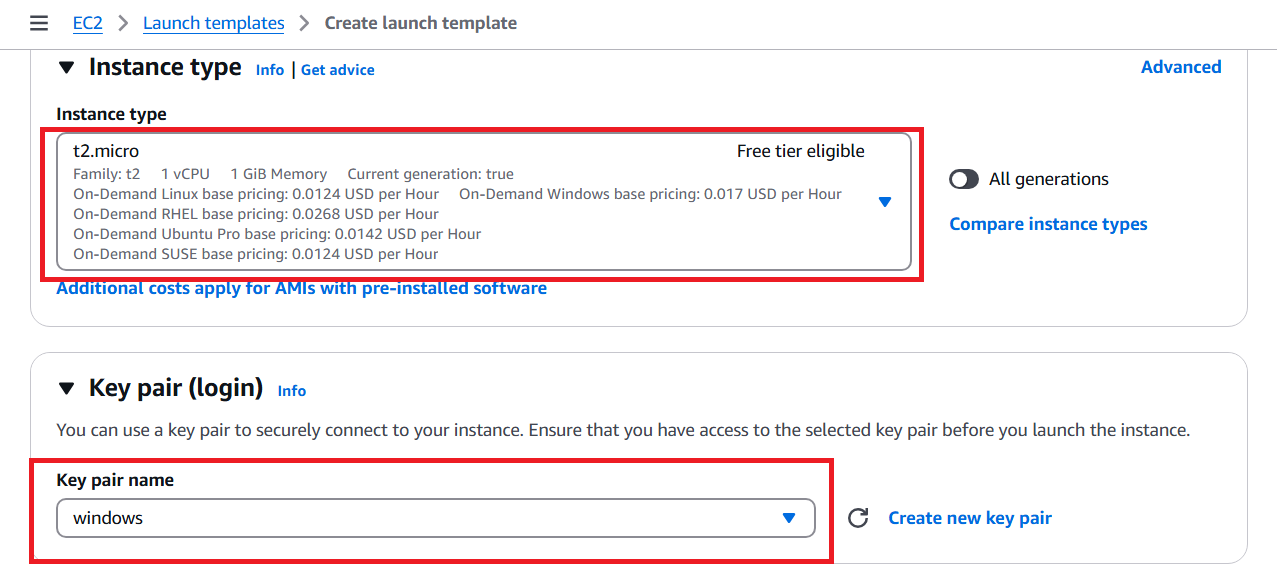
**A screenshot of a computer

Description automatically generated**

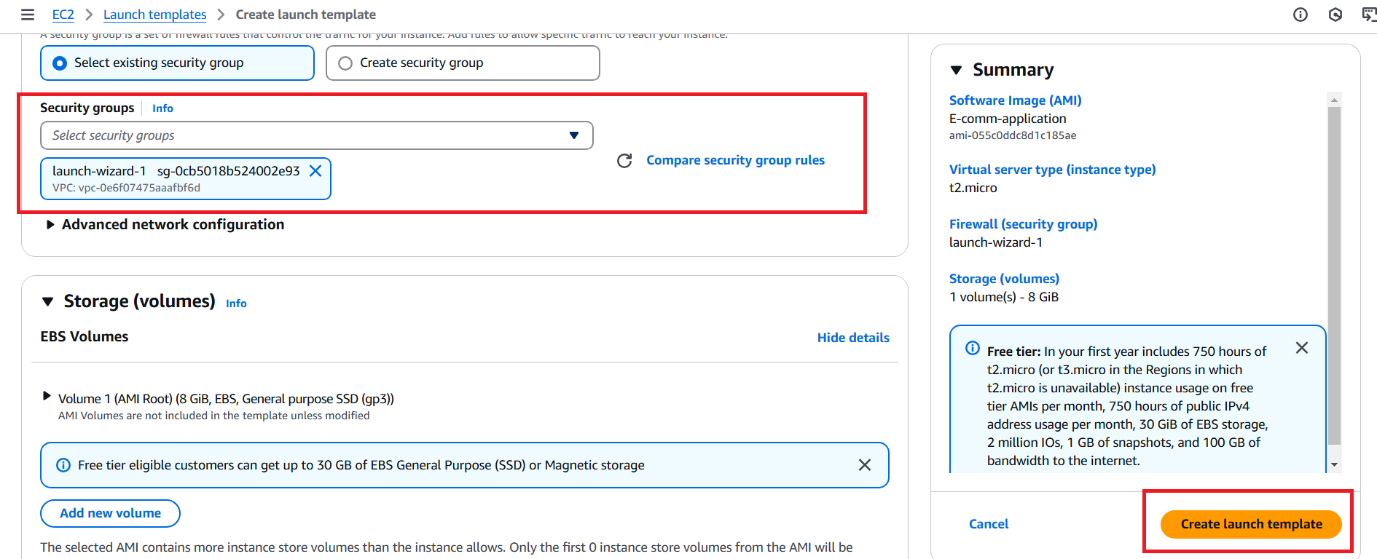
**Creating the launch template**

****

****

****

**Select the correct security group which is allowed HTTP and HTTPS port**

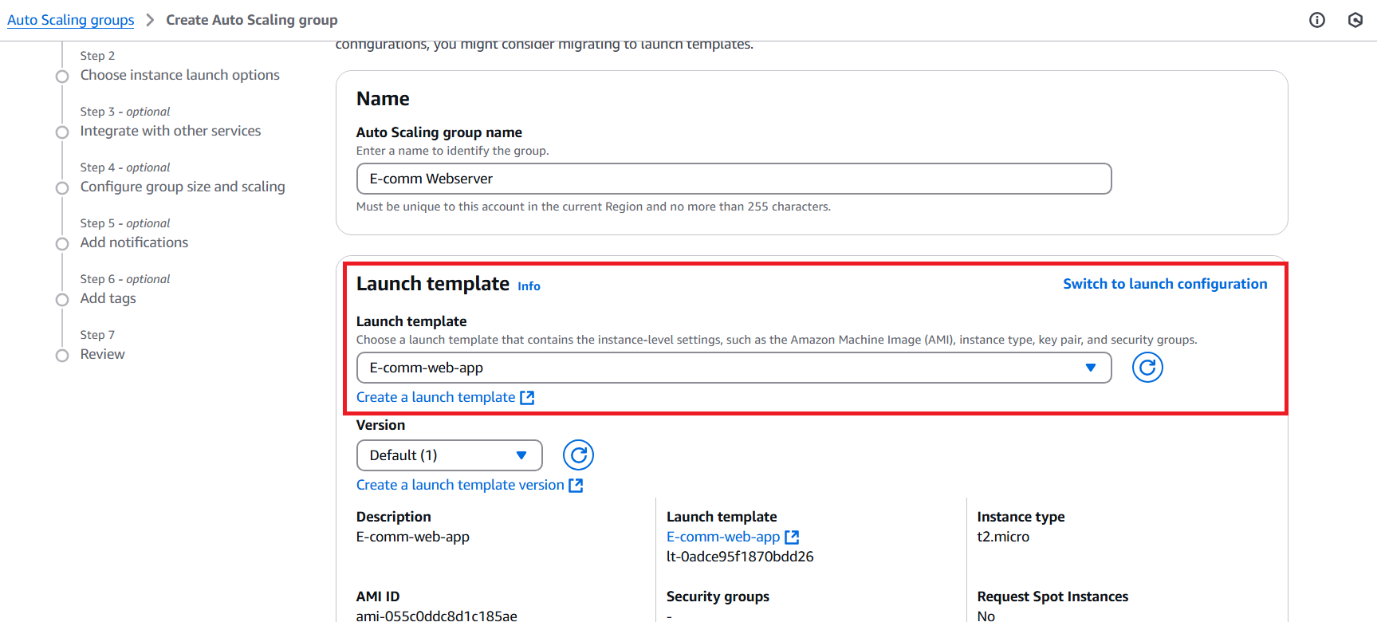
****

**A green and white stripe

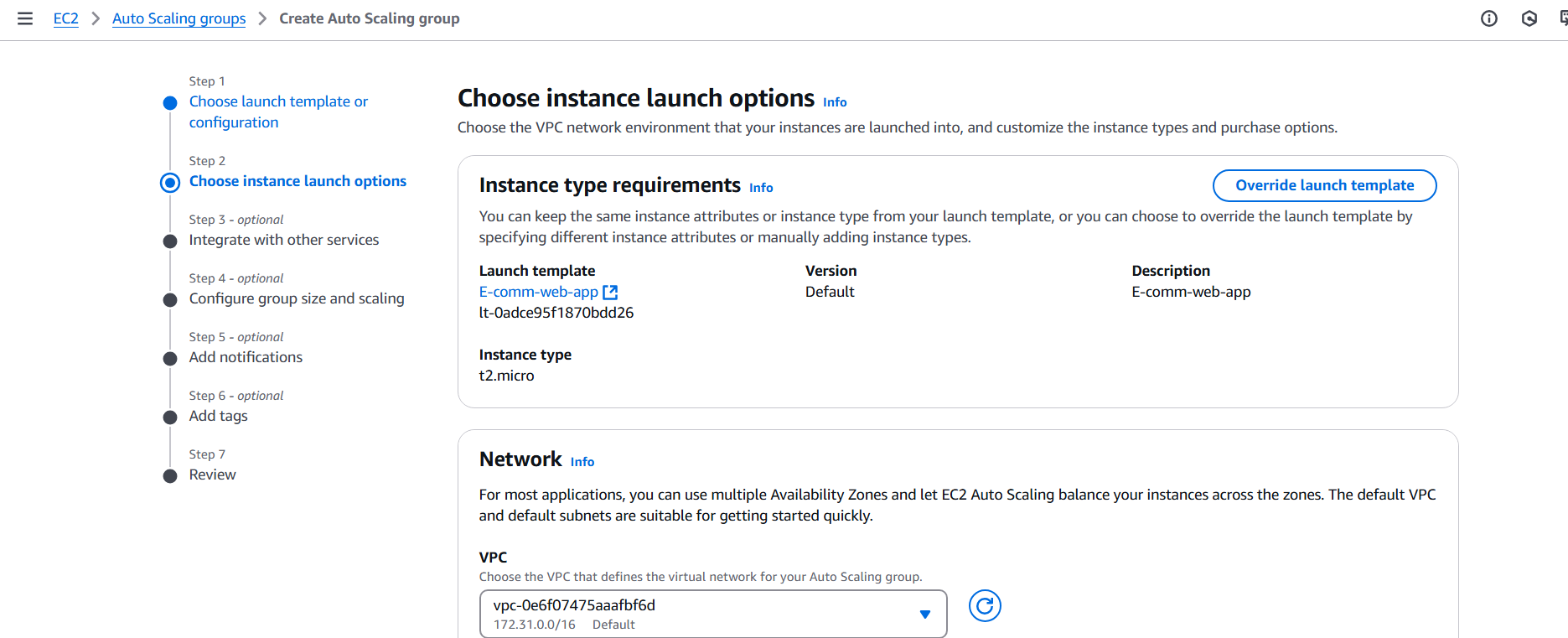
Description automatically generated**

**You may terminate the instance that was launched, configured, and used to install the web server and clone the repository, as you now have the AMI copy. It is no longer required since you will configure the Auto-Scaling Group.**

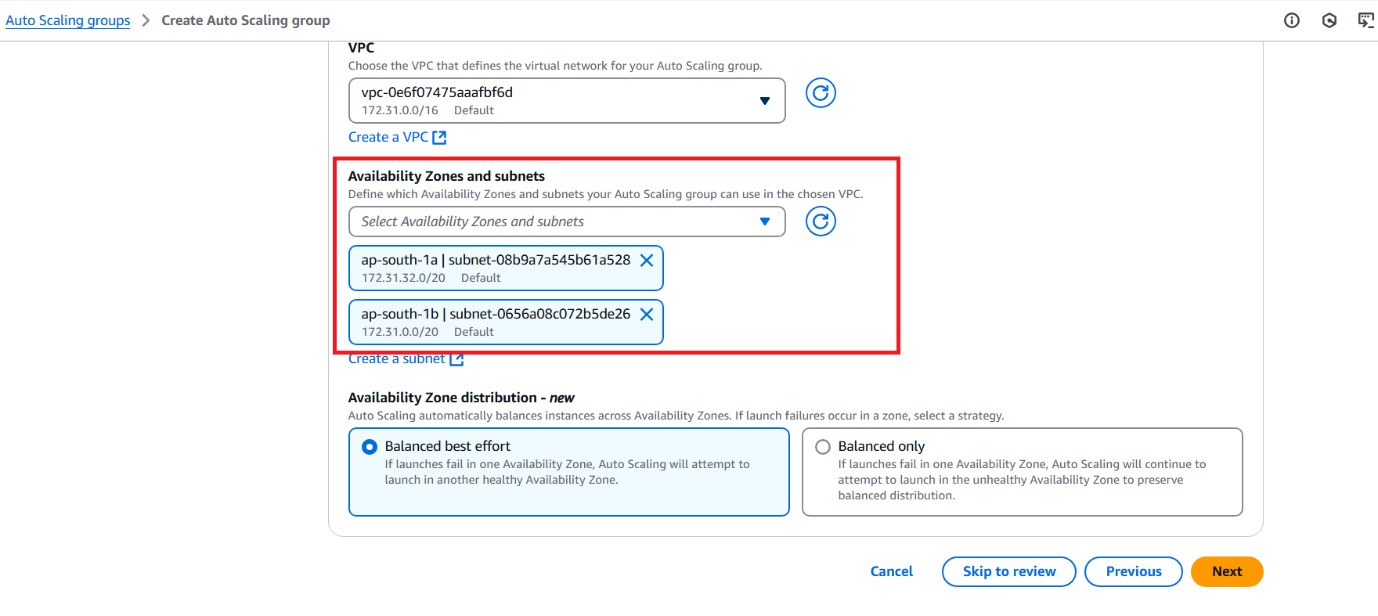
**Once the template has been created proceed with configuring the Auto-Scaling Group.**

****

**Click on next**

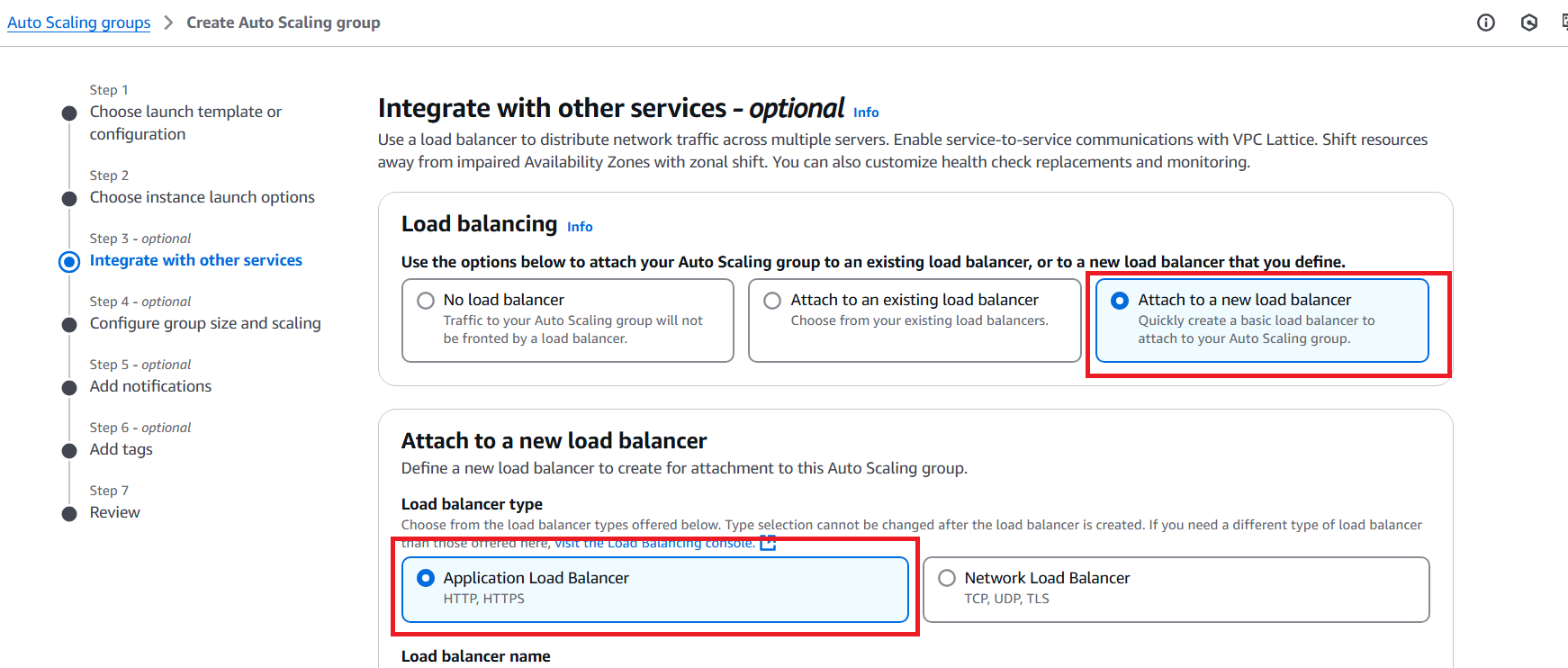
****

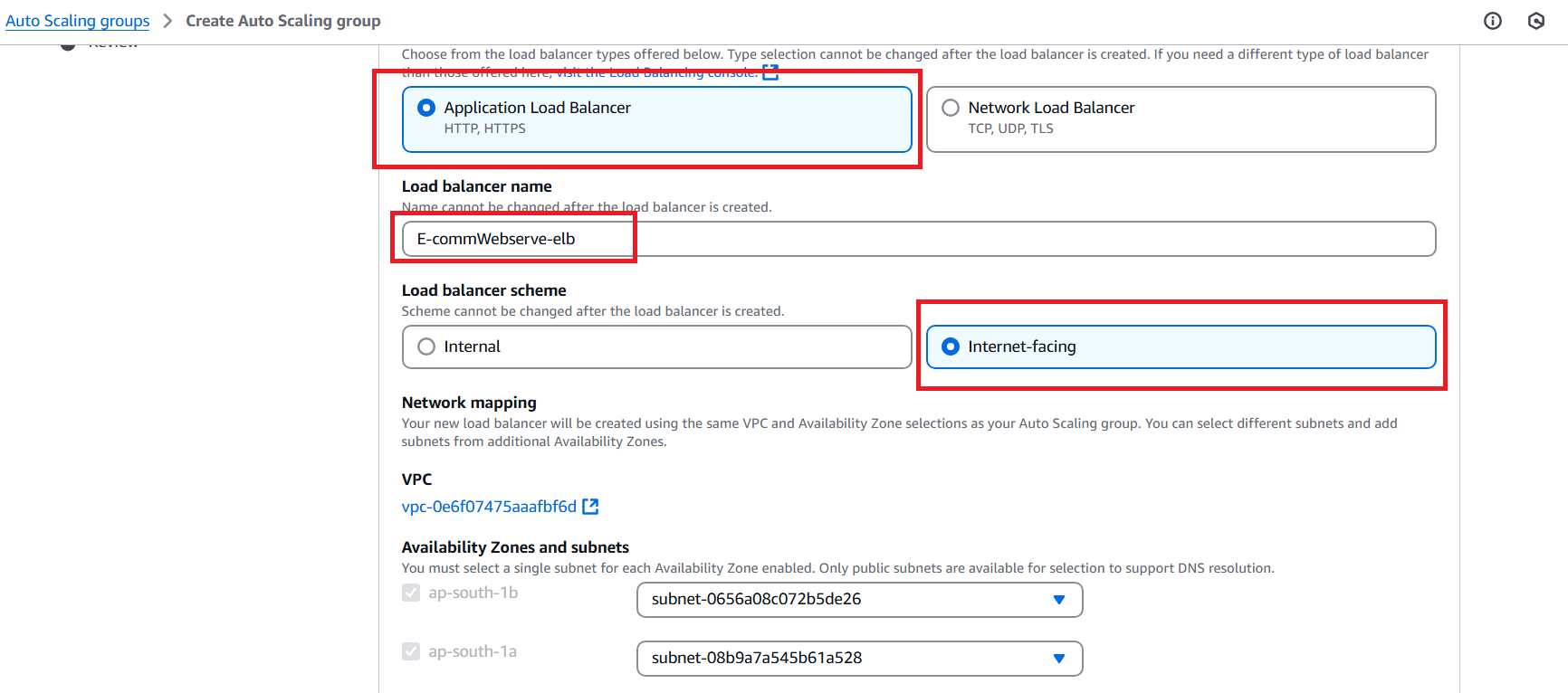
**Select any of the two availability zones for high availability. And click next**

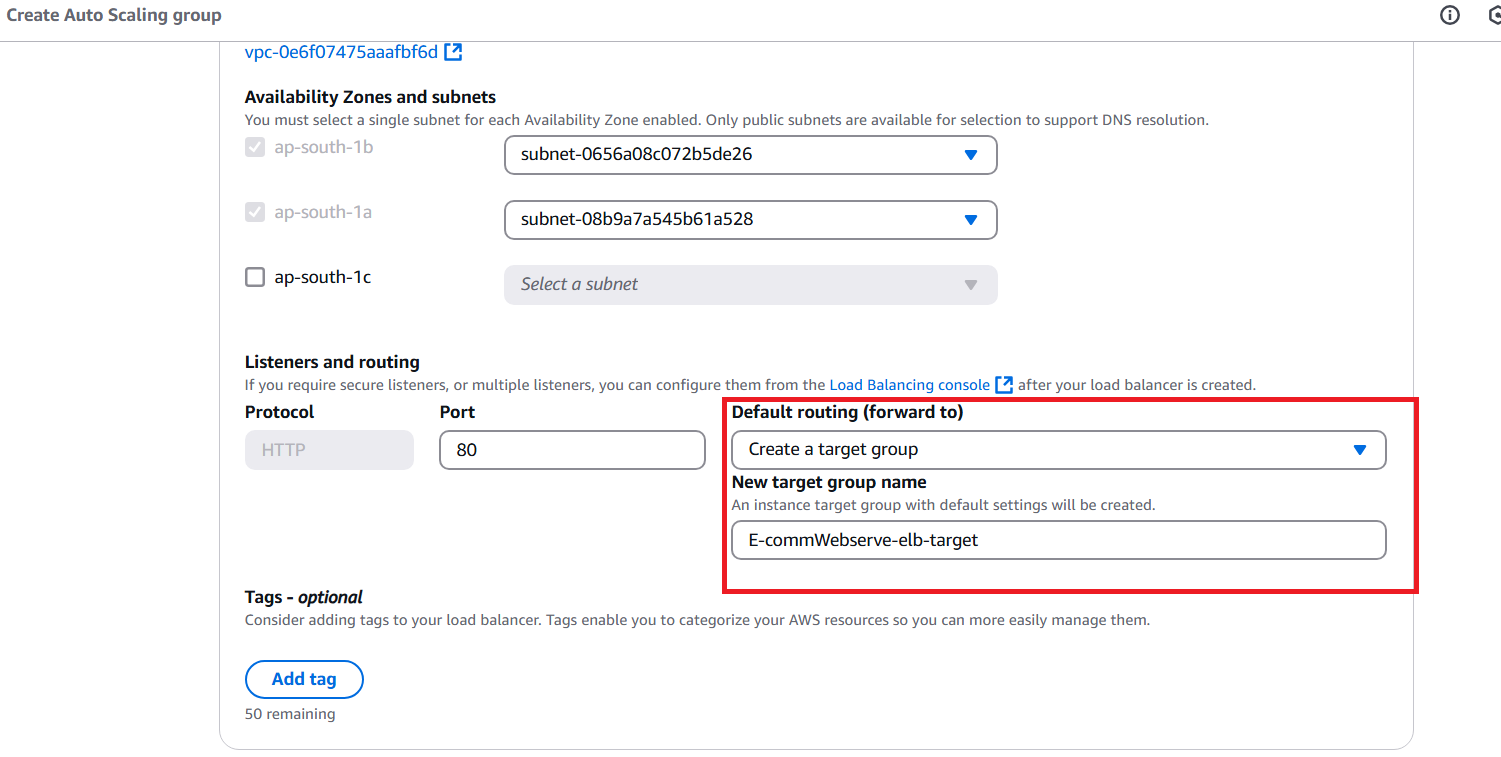
****

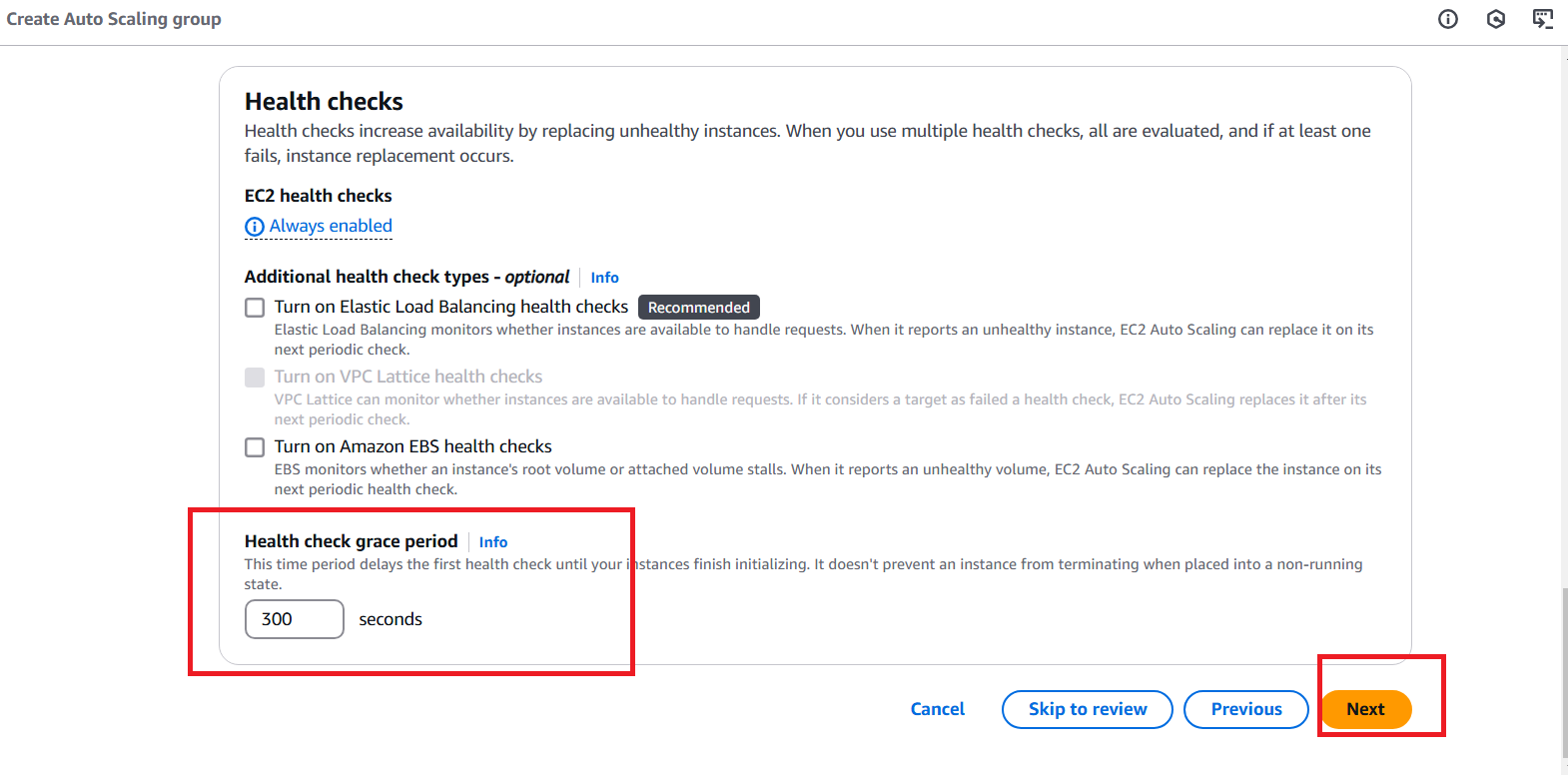
**The architecture requires an Application Load Balancer. While creating the Auto-Scaling Group, you can directly create an Application Load Balancer.**

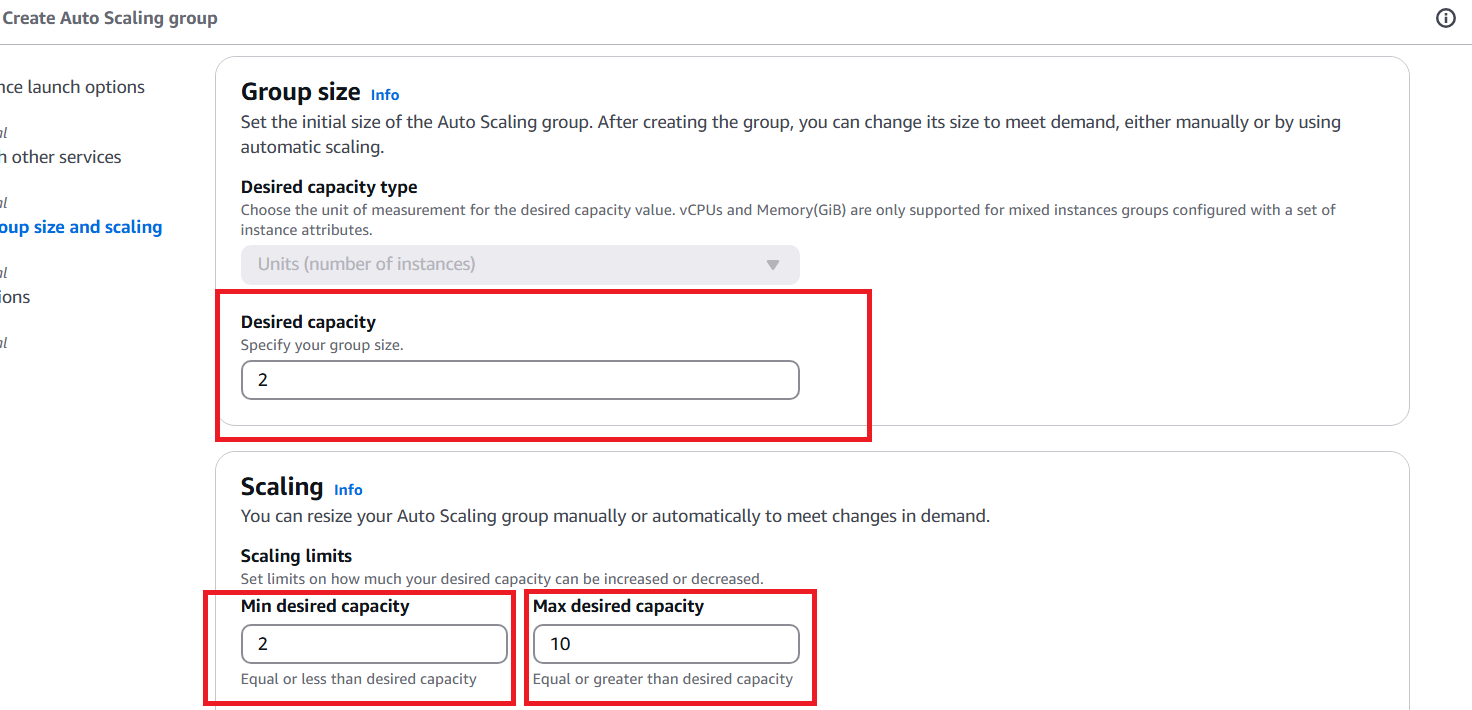
**Correct the ELB name and target group name.**

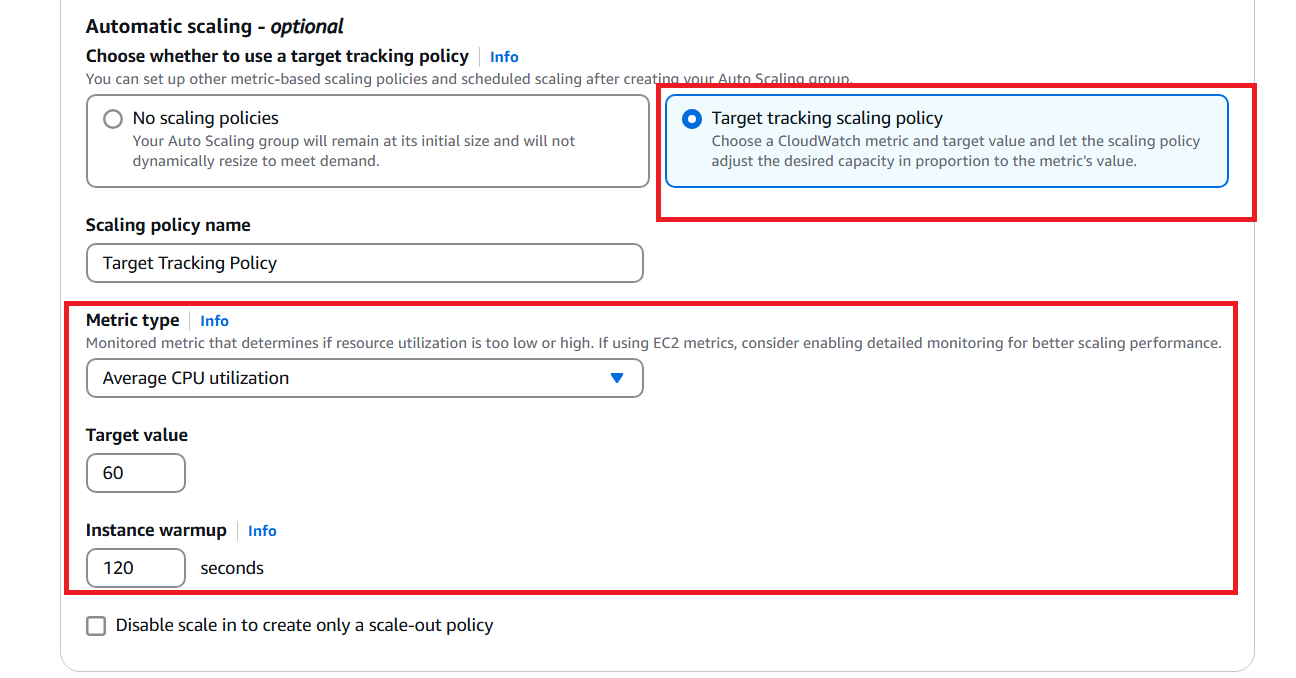
****

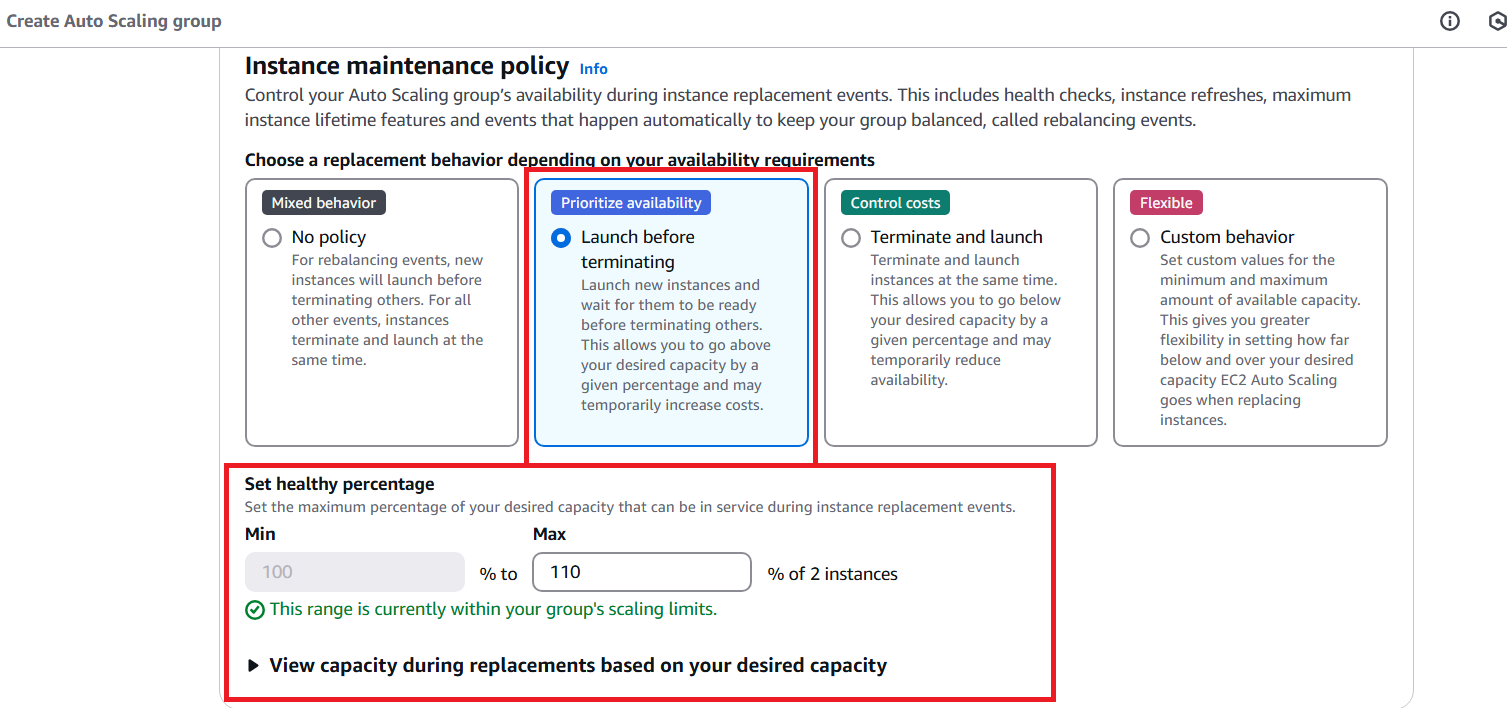
****

****

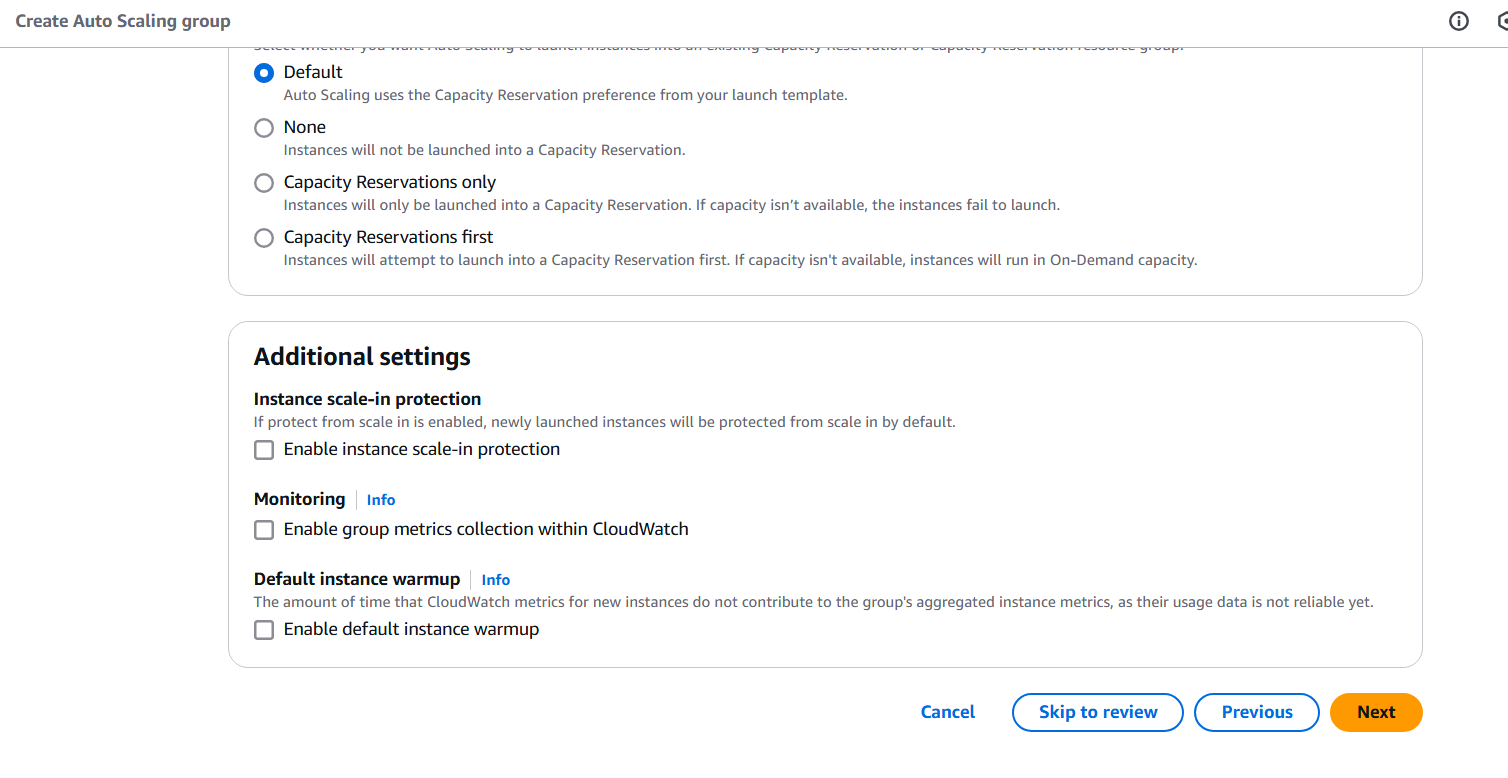
****

****

****

****

**Click on next**

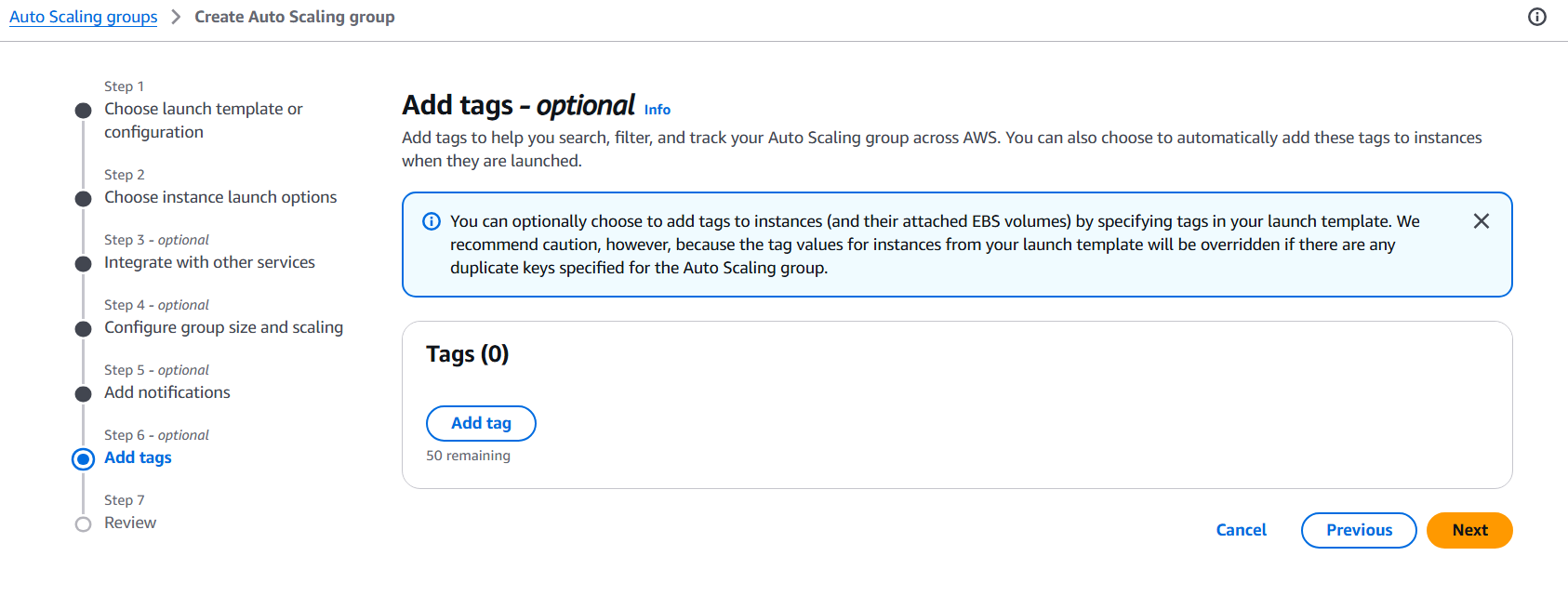
****

**Click on next**

**A screenshot of a computer

Description automatically generated**

**Click on next**

****

**Review the configuration and click on the Create Auto-Scaling group Button.**

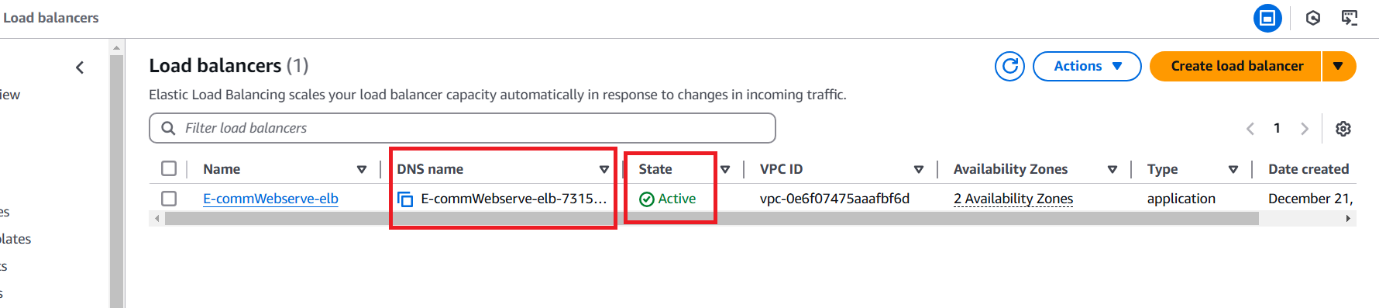
**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**You may need to wait for some time while the resources, such as EC2 instances, ELB, and target groups, are being created. Once the instances are up and running, you can access the E-Commerce application using the ELB DNS. Once the ELB Status is Active**

****

**A screenshot of a website

Description automatically generated**

**Great Job well done!  
  
You have successfully completed the project with high availability, capable of handling millions of requests.**

**Note: Do not forget to terminate the AWS resources you have created to avoid unnecessary costs.  
  
Delete the AWS ASG  
Delete the AWS ELB and Target groups  
Delete the AMI and Snapshot  
Delete the Launch template**

**Thanks**