1. Part-1: EC2 with ELB and ASG

Objective: Learn how to create a scalable and highly available web application environment using Amazon EC2 instances, ELB, and ASG.

Approach:

Launch EC2 Instances: Start by launching two or more EC2 instances. These instances will run a simple web application (e.g., a "Hello World" page or any basic web service).

Configure Load Balancer: Set up an Elastic Load Balancer (ELB) to distribute incoming web traffic across your EC2 instances. This step ensures high availability and fault tolerance.

Set Up Auto Scaling Group (ASG): Create an ASG that uses the launched EC2 instances. Configure ASG policies to automatically scale the number of instances up or down based on criteria like CPU usage or network traffic.

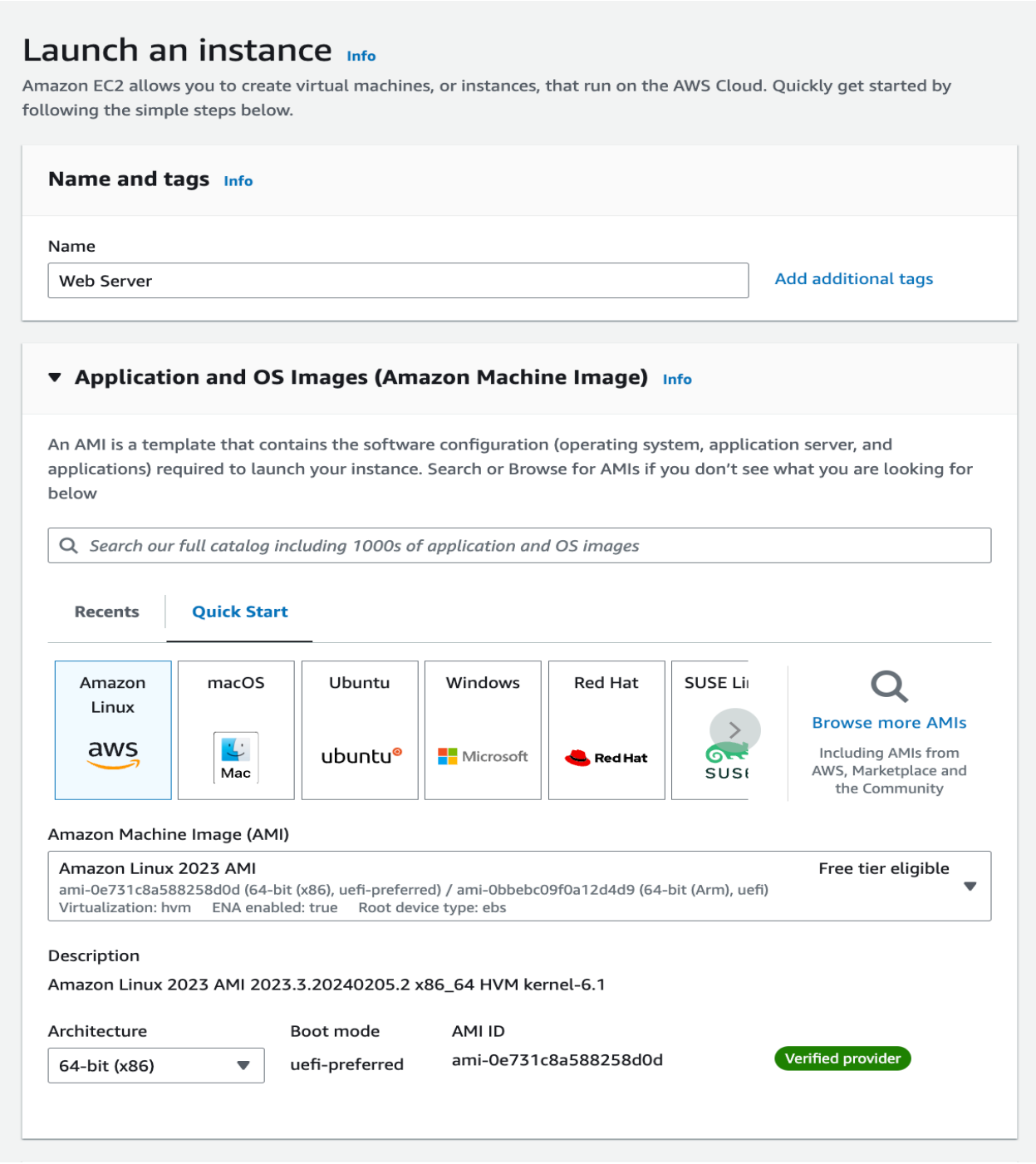
Test Your Setup: Simulate traffic to test the scaling policies and the load balancer. Observe how ASG adds or removes instances and how ELB distributes traffic.

Verify Website Functionality: Ensure that the website hosted on EC2 instances remains accessible and functional during scaling operations.

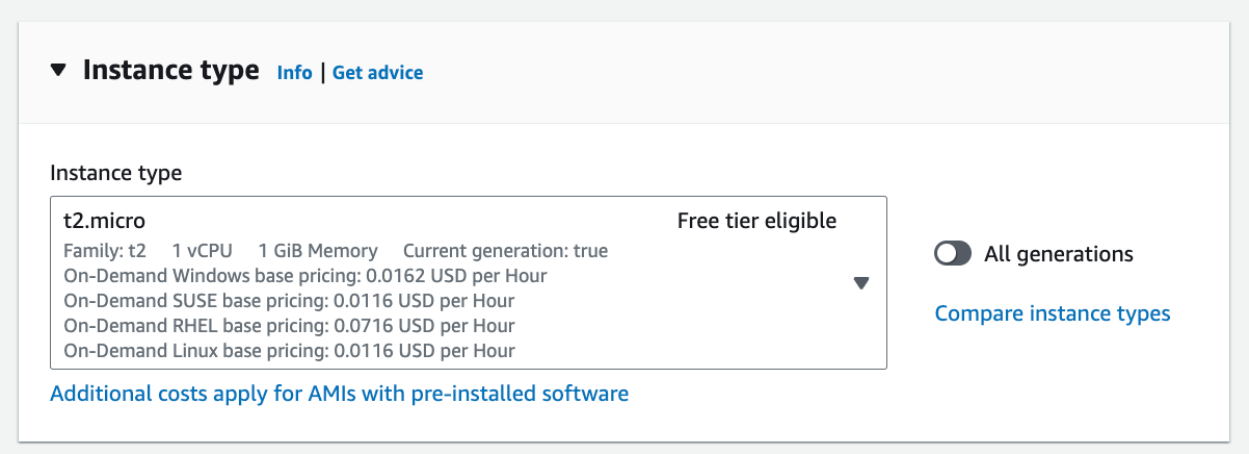
Goal: By the end of this lab, students will have a hands-on understanding of setting up a load-balanced and auto-scaled web application using AWS services.

1.1. Launch EC2 instance

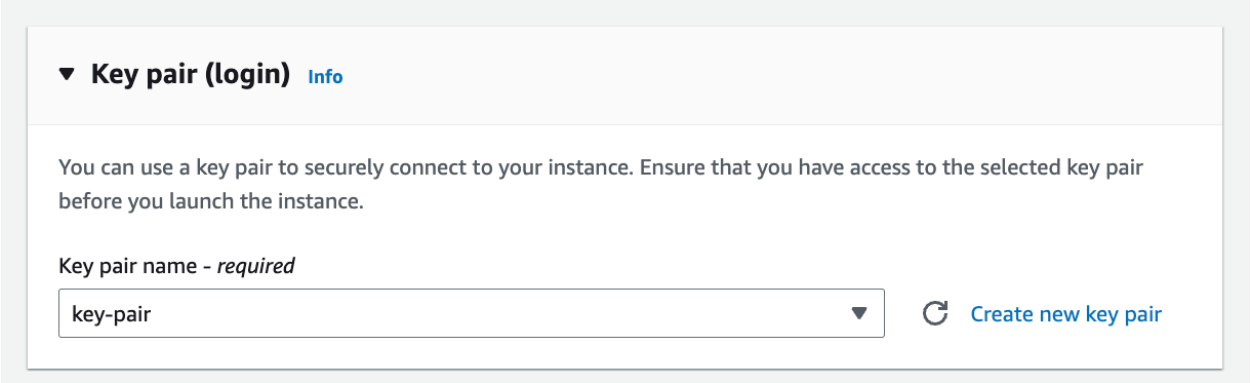
Give name of instance and select application and OS Images



T2.micro instance type is selected.



Existing key pair is selected, which was created initially.



Network settings Info

VPC - *required* Info

vpc-0d8fab51a5f972f19 (default)
172.31.0.0/16

Subnet Info

No preference

Create new subnet

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

- (•) Create security group
- ( ) Select existing security group

Security group name - *required*

web-ssh

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

web server ssh

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/8) — Remove

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info	Source Info	Description - *optional* Info
Custom	Add CIDR, prefix list or security	e.g. SSH for admin desktop

0.0.0.0/8 ×

Security group rule 2 (TCP, 80, 0.0.0.0/8) — Remove

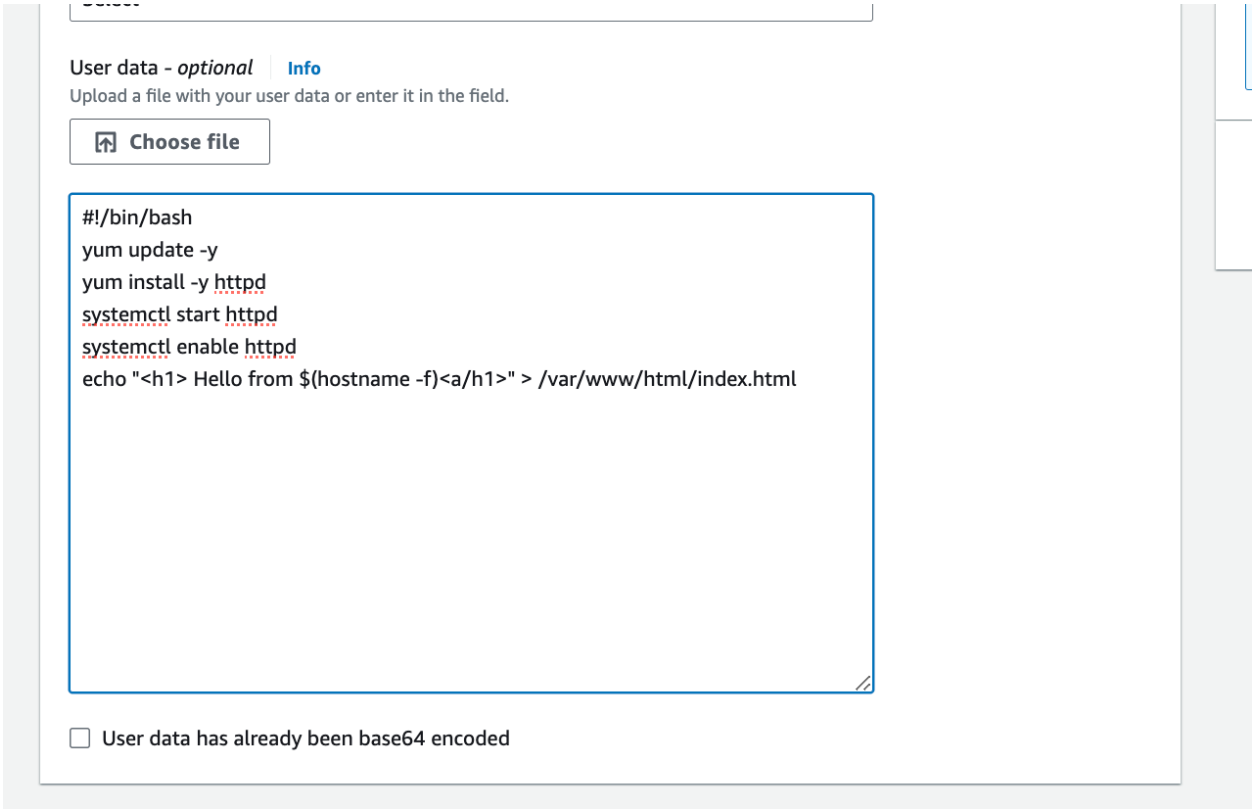
Type Info	Protocol Info	Port range Info
HTTP	TCP	80

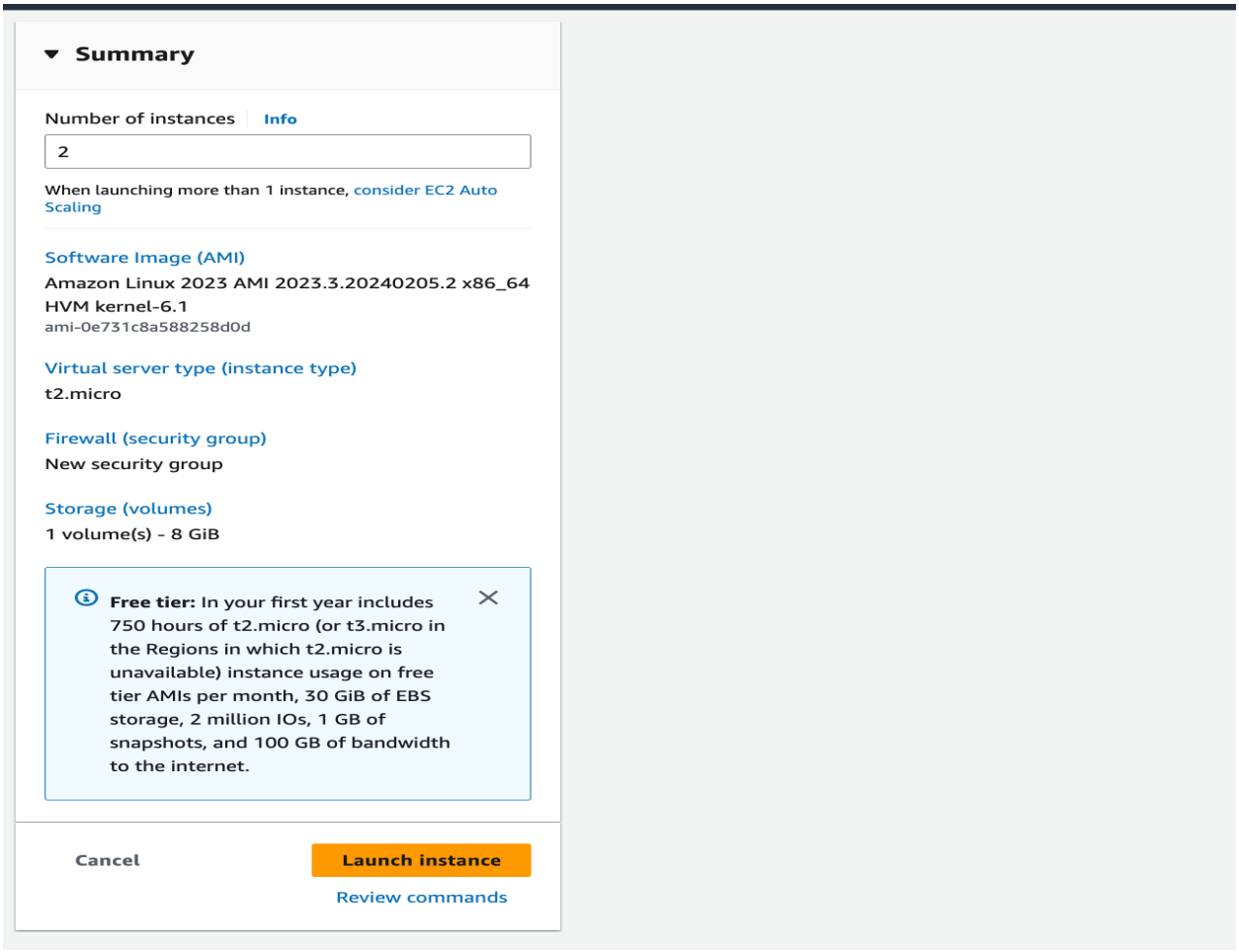
Source type Info	Source Info	Description - *optional* Info
Custom	Add CIDR, prefix list or security	e.g. SSH for admin desktop

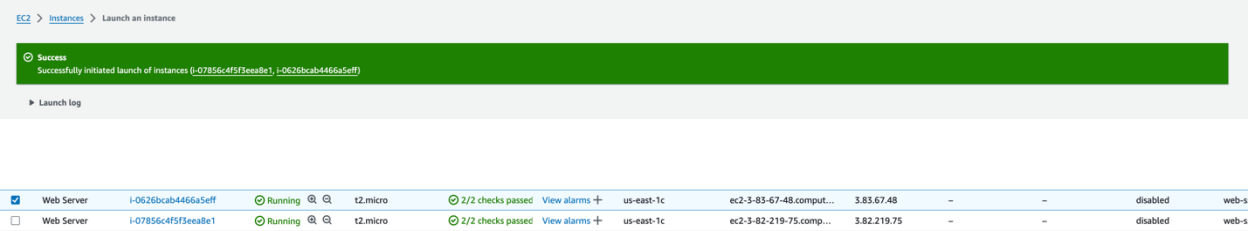
0.0.0.0/8 ×

Add security group rule

At Advanced Detail section below script is attached and rest configuration are left as it is.

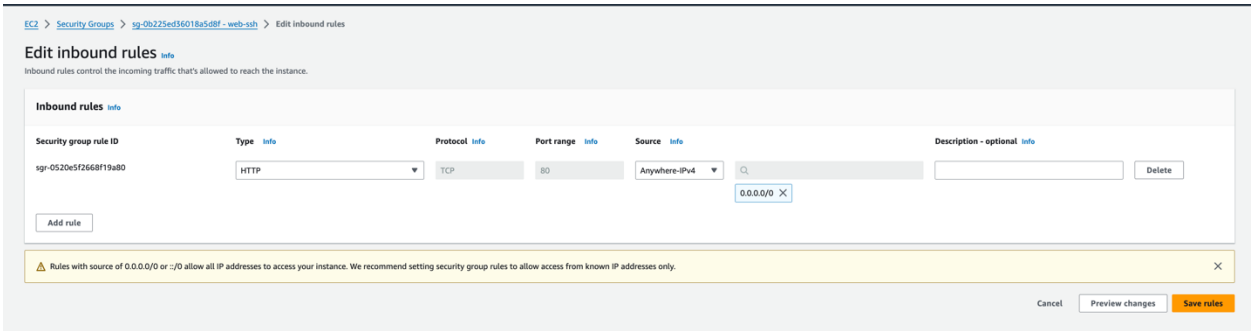



Summary
Number of instances Info
2
When launching more than 1 instance, consider EC2 Auto Scaling
Software Image (AMI)
Amazon Linux 2023 AMI 2023.3.20240205.2 x86_64 HVM kernel-6.1
ami-0e731c8a588258d0d
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.
Cancel
Launch instance
Review commands

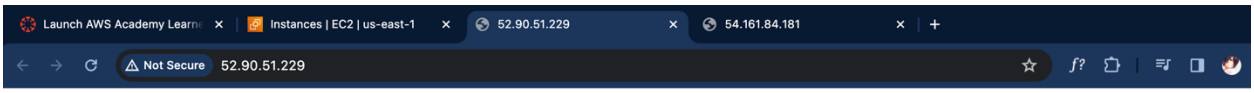


Ec2 instance with same configuration is created.

The created security group inbound rule is updated as the ec2 instance was not working while trying to run from Public IPv4 address of the created EC2 instance.



After updating inbound rule, the public IPv4 address of both ec2 instances worked in web browser.

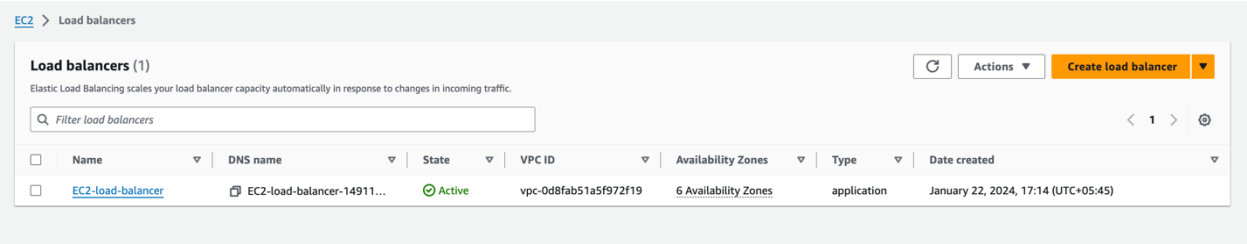


Hello from ip-172-31-83-149.ec2.internal

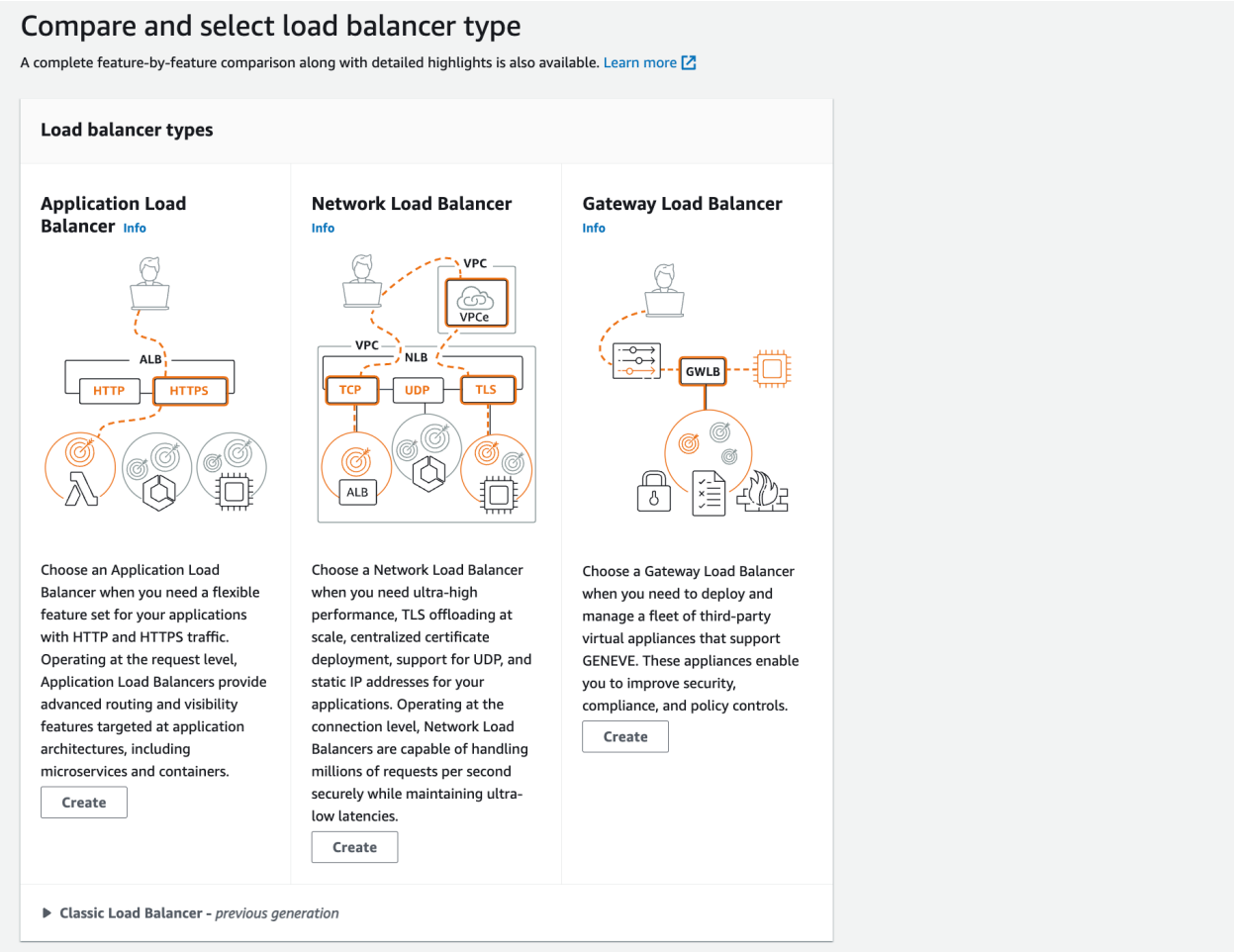
Hello from ip-172-31-91-151.ec2.internal

1.2. Configure Load Balancer

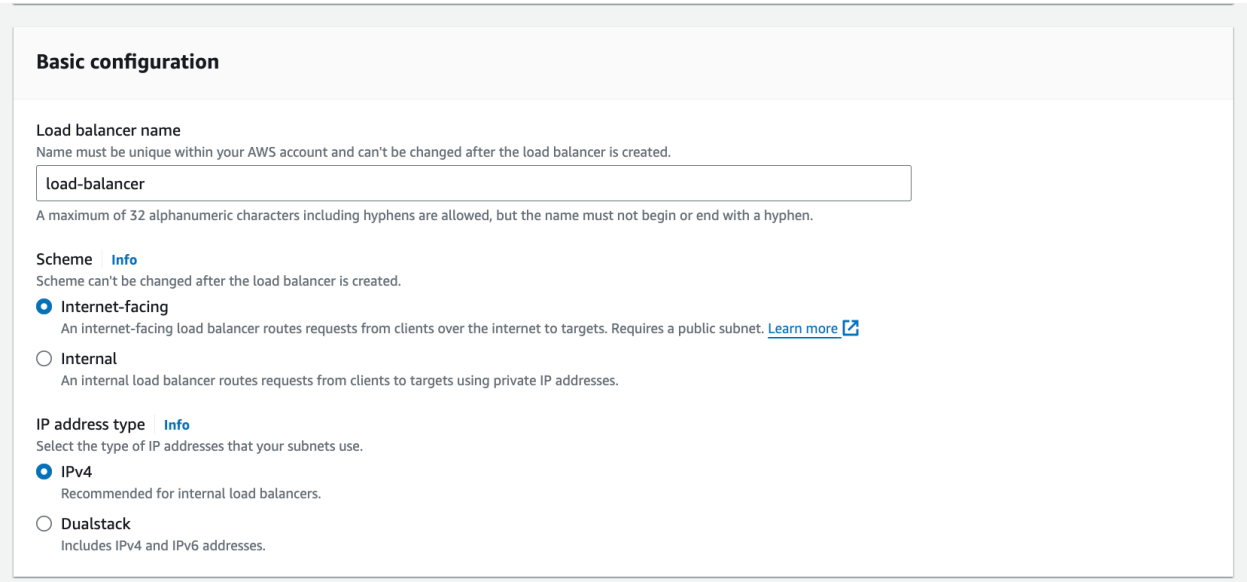
Navigate to load balancers screen and click on “Create load balancer” and create a new load balancer.



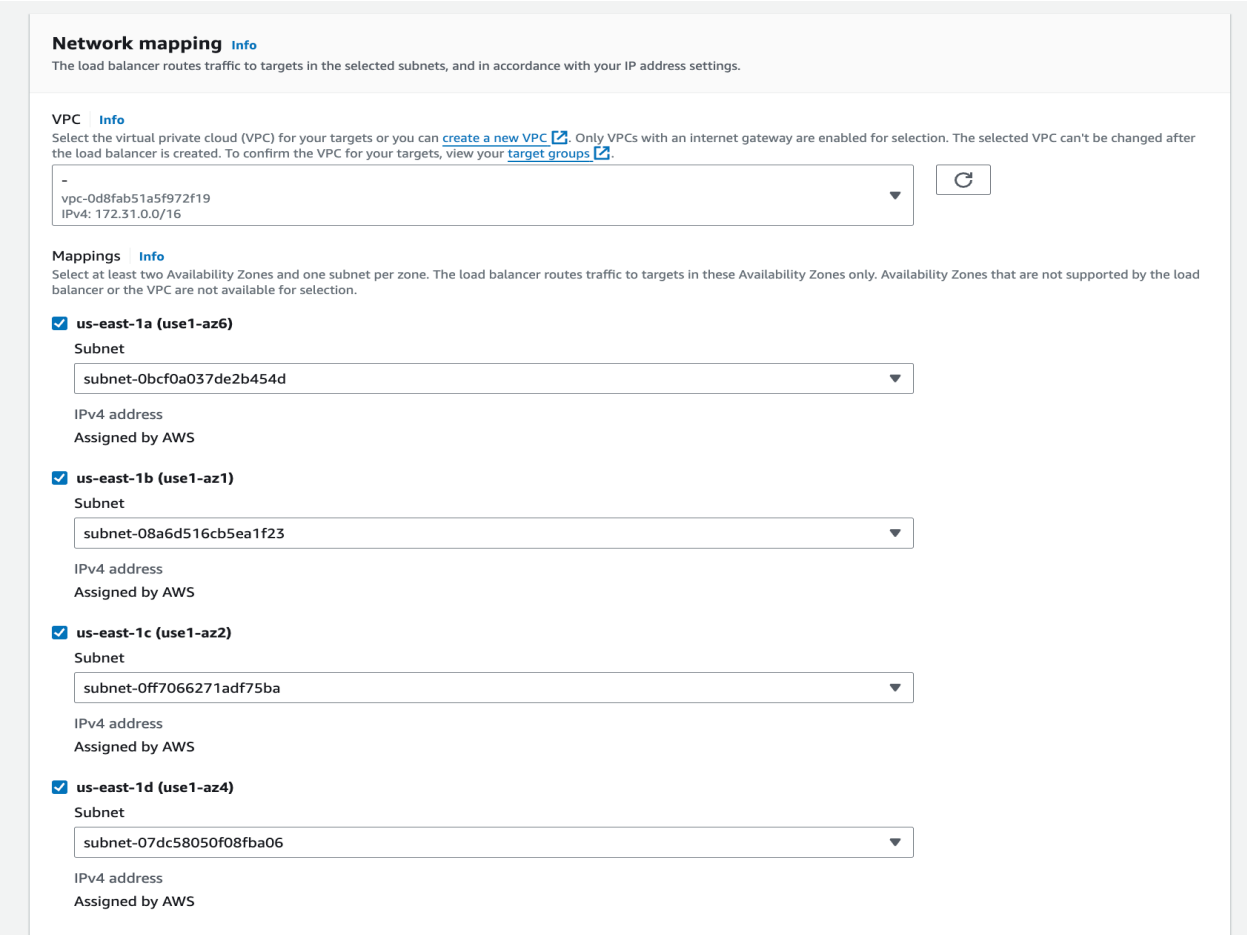
For this, lab Application Load balancer is created.



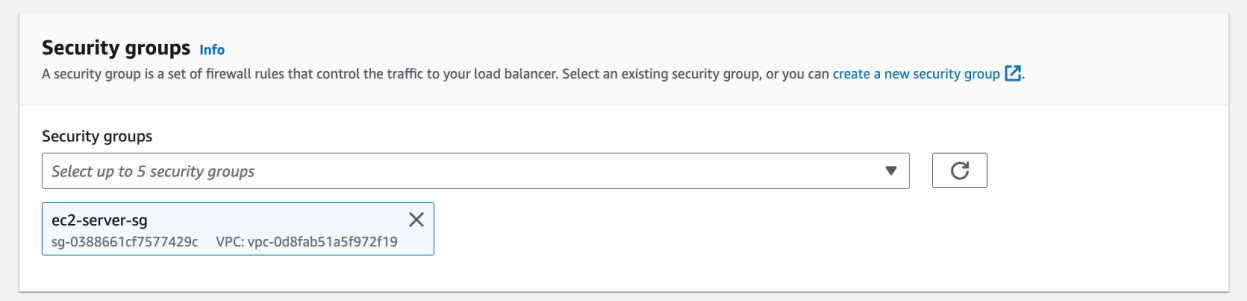
Required configuration are provided.



Default VPC and all the available zone in mappings are selected.



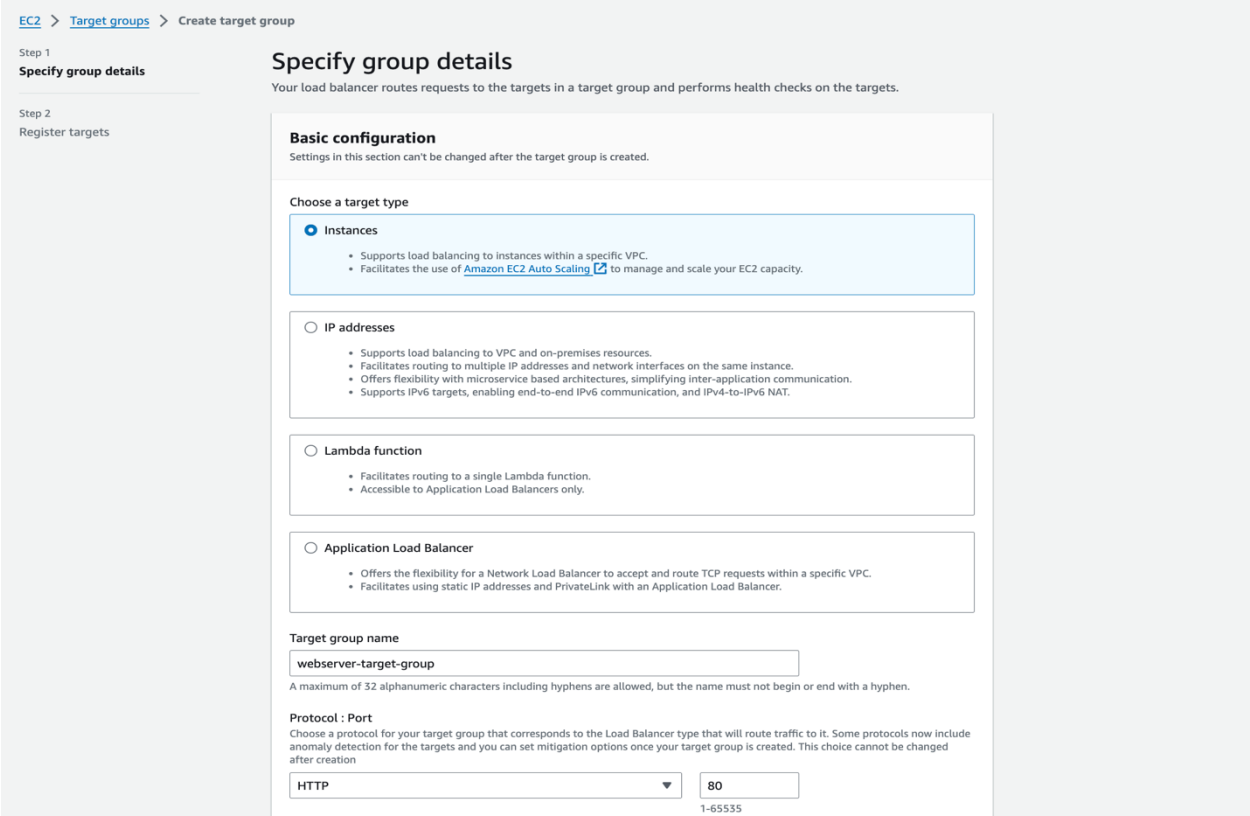
Existing security group is selected.



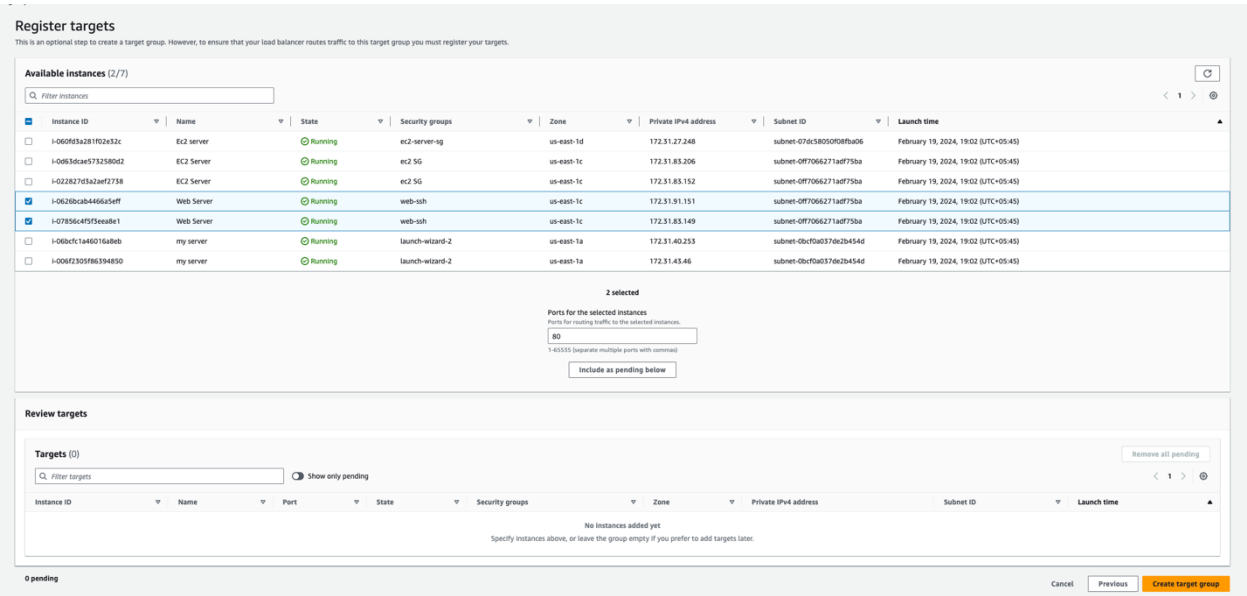
In Listeners and routing section, target group is to be selected. For this purpose, new target group is created.

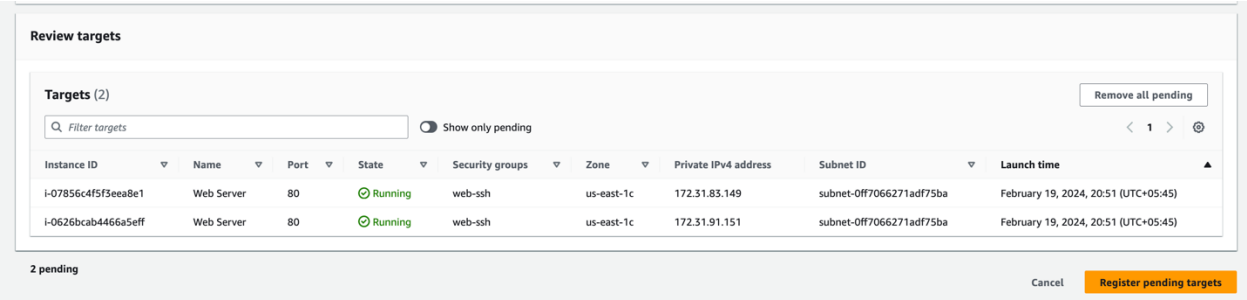
1.2.1. Target group creation

Basic configuration for target group is provided. Here, target group instance is chosen and for rest default configuration are kept as it is.

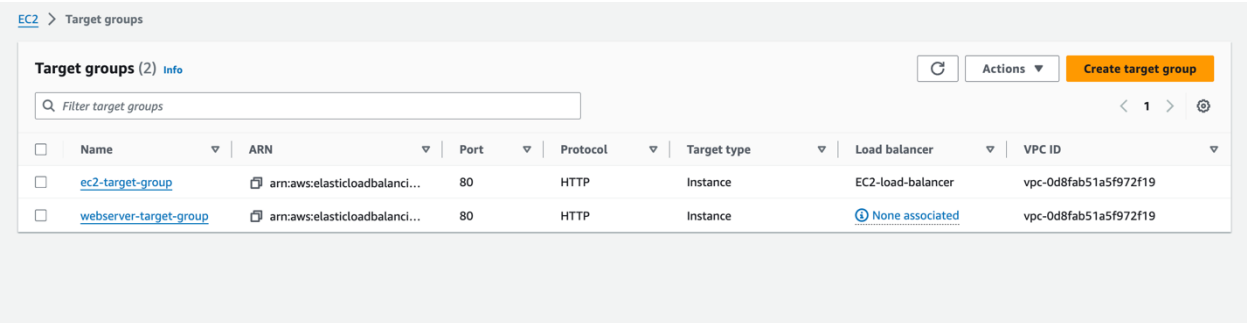


After that, target EC2 instances are registered in this newly created target group.

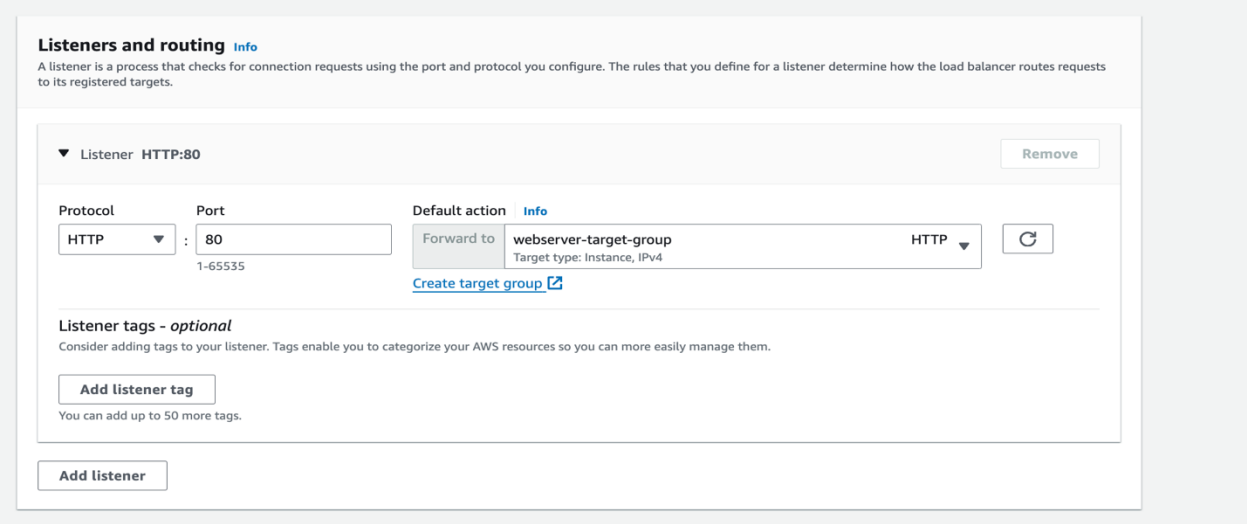




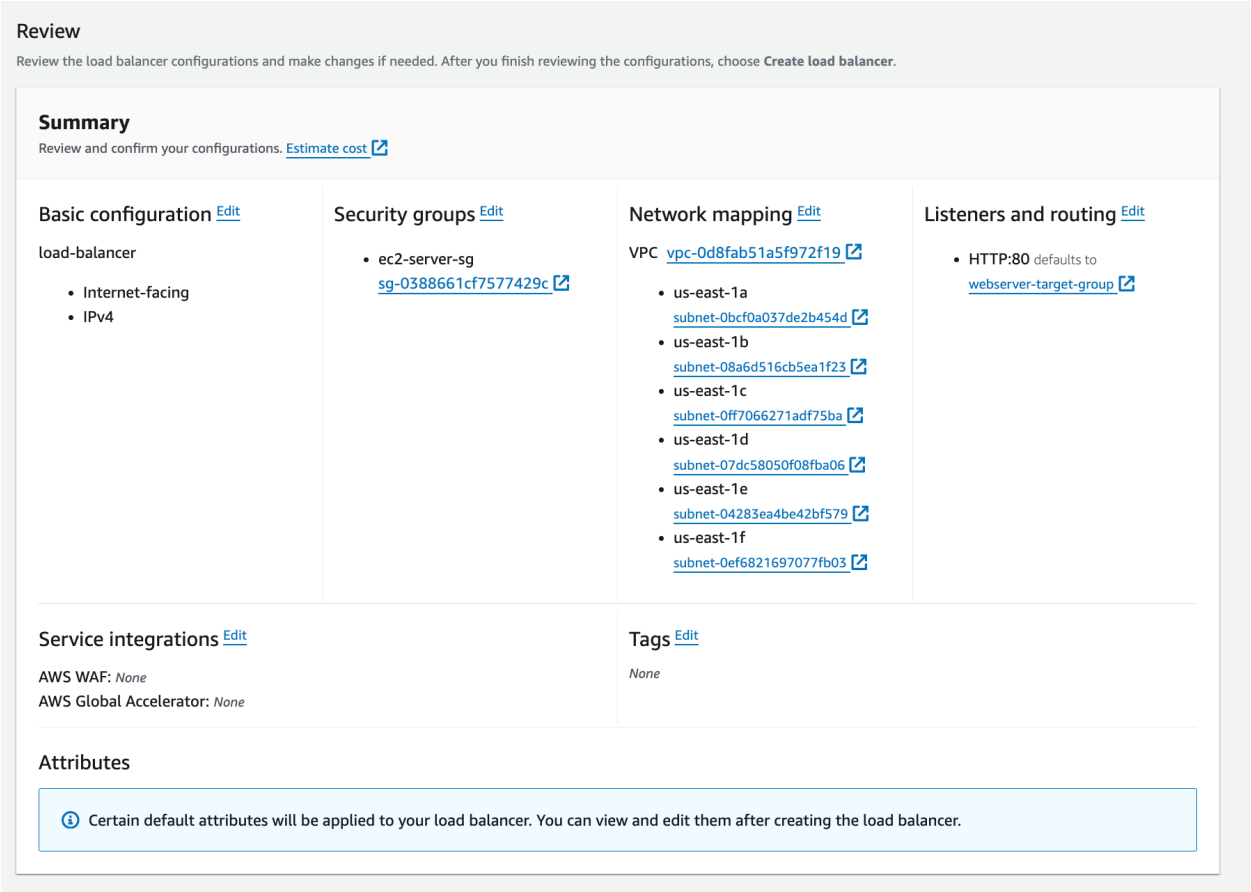
Here, the created target group can be viewed in target group lists.



Now, the load balancer configuration can be continued. In Listeners and routing section, recently created target group is selected.

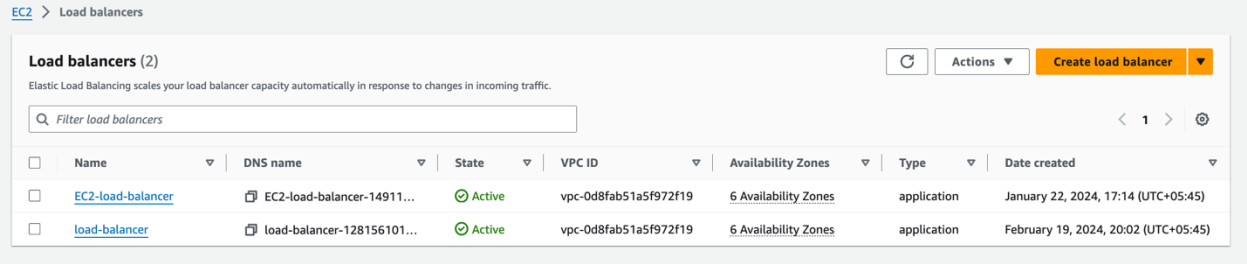


The summary of configuration is reviewed and after confirmation load balancer is created.

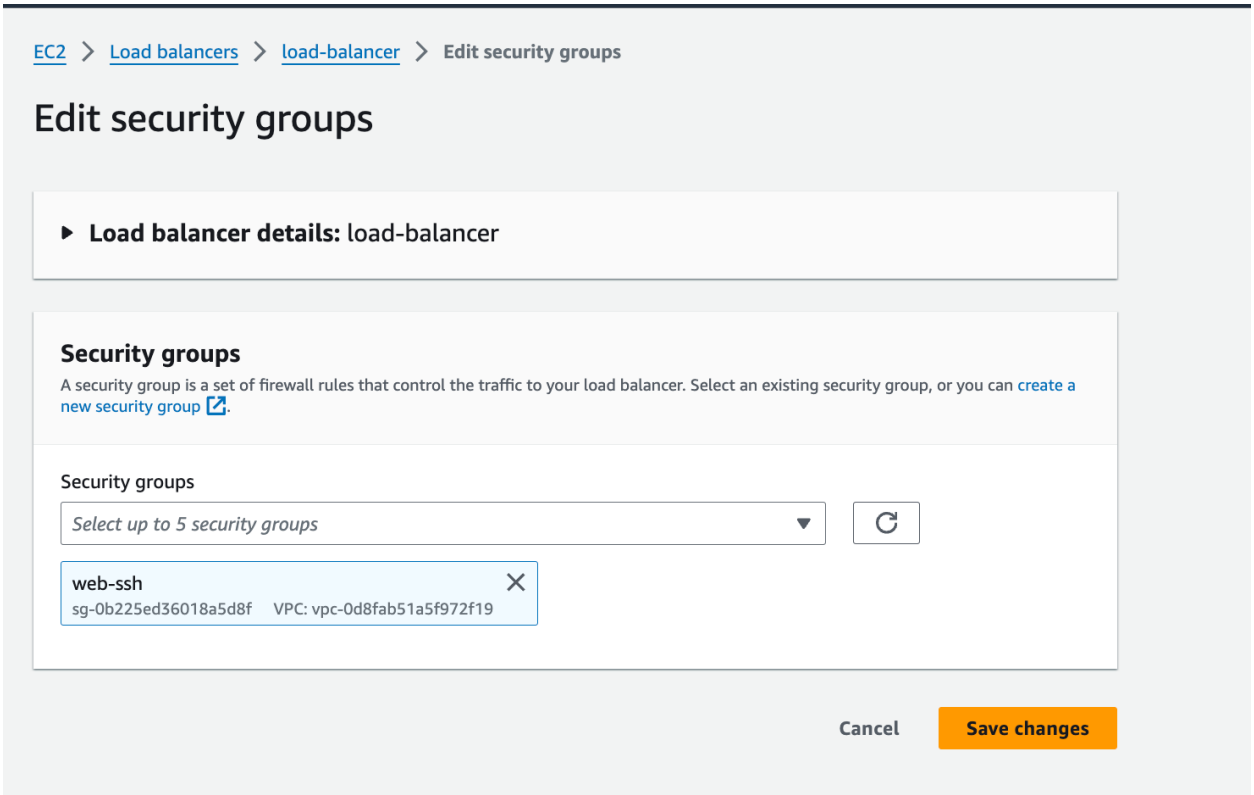


Testing load balancer

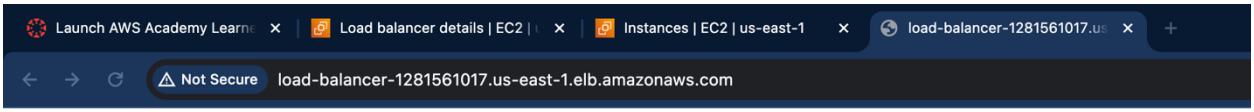
After the created load balancer ”load-balancer” is in active state,



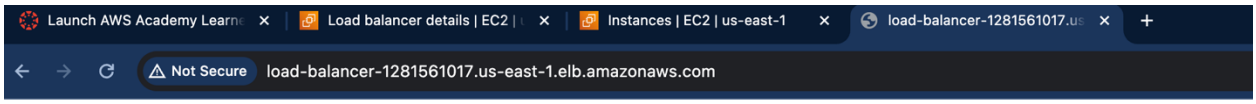
Security group was updated to another security group “web-ssh” which contains http inbound rule, so that the instance can accessed.



After that, the DNS name of load balancer is copied and accessed in web browser. When the URL is refreshed, multiple times it runs different instance of the specified target group to maintain the load.



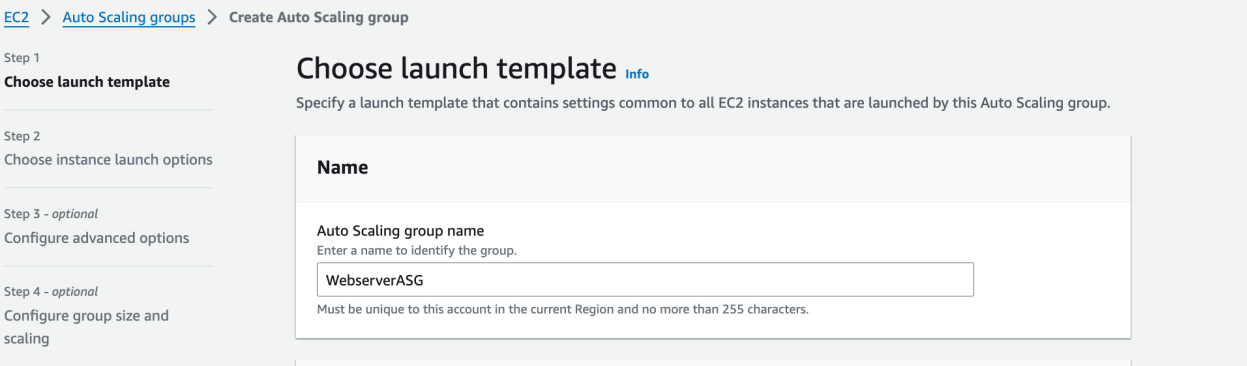
Hello from ip-172-31-91-151.ec2.internal



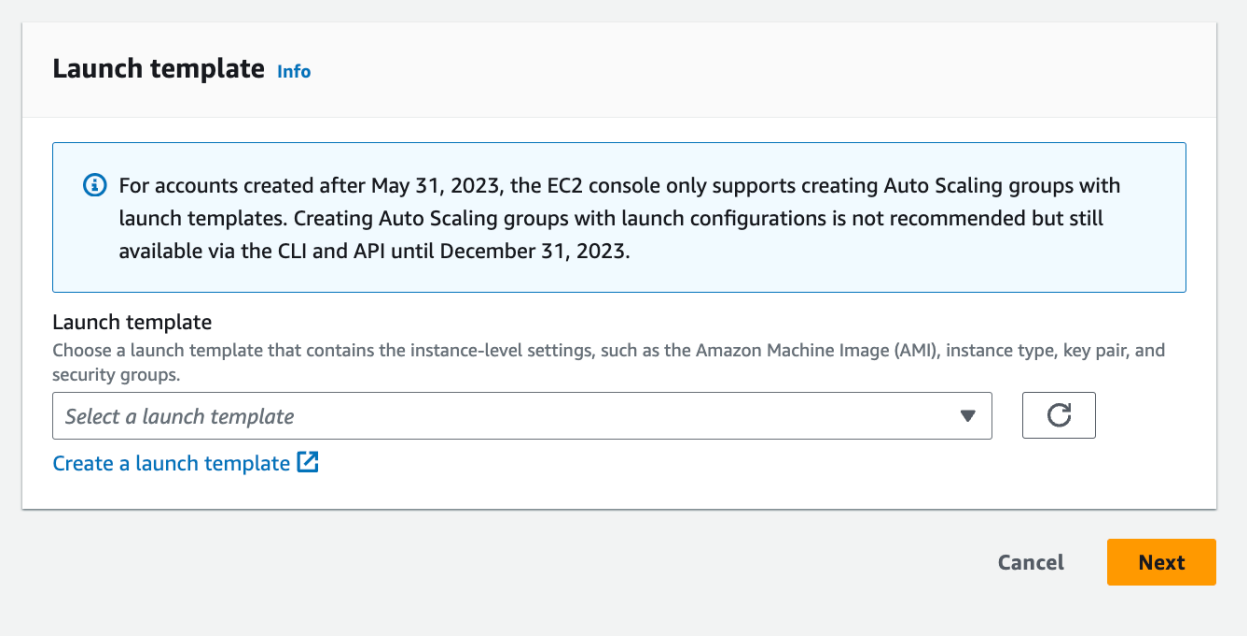
Hello from ip-172-31-83-149.ec2.internal

1.3. Set up Auto Scaling Group

Create new auto scaling group.

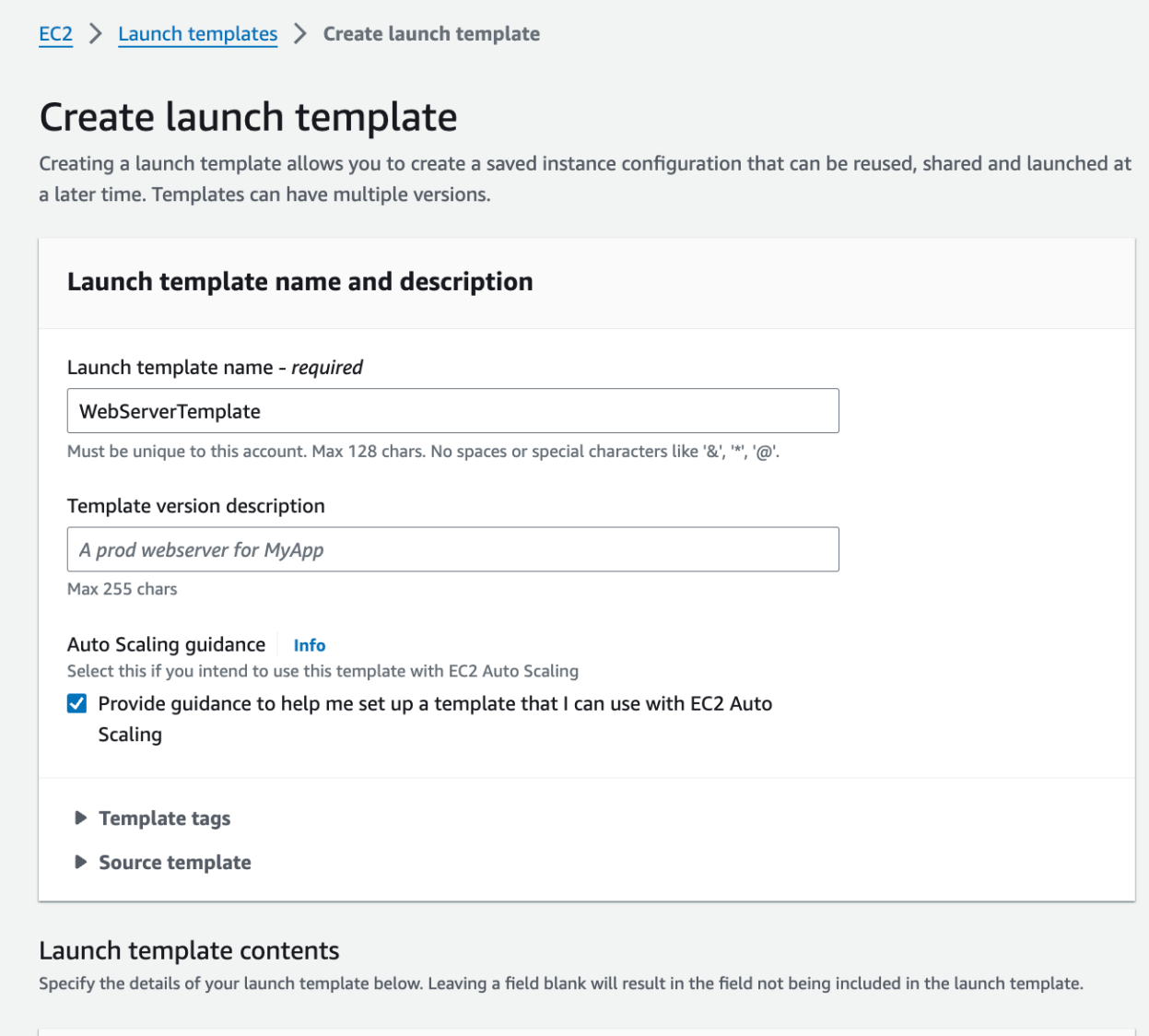


In launch template section, “Create a launch template” is selected and new template is created.

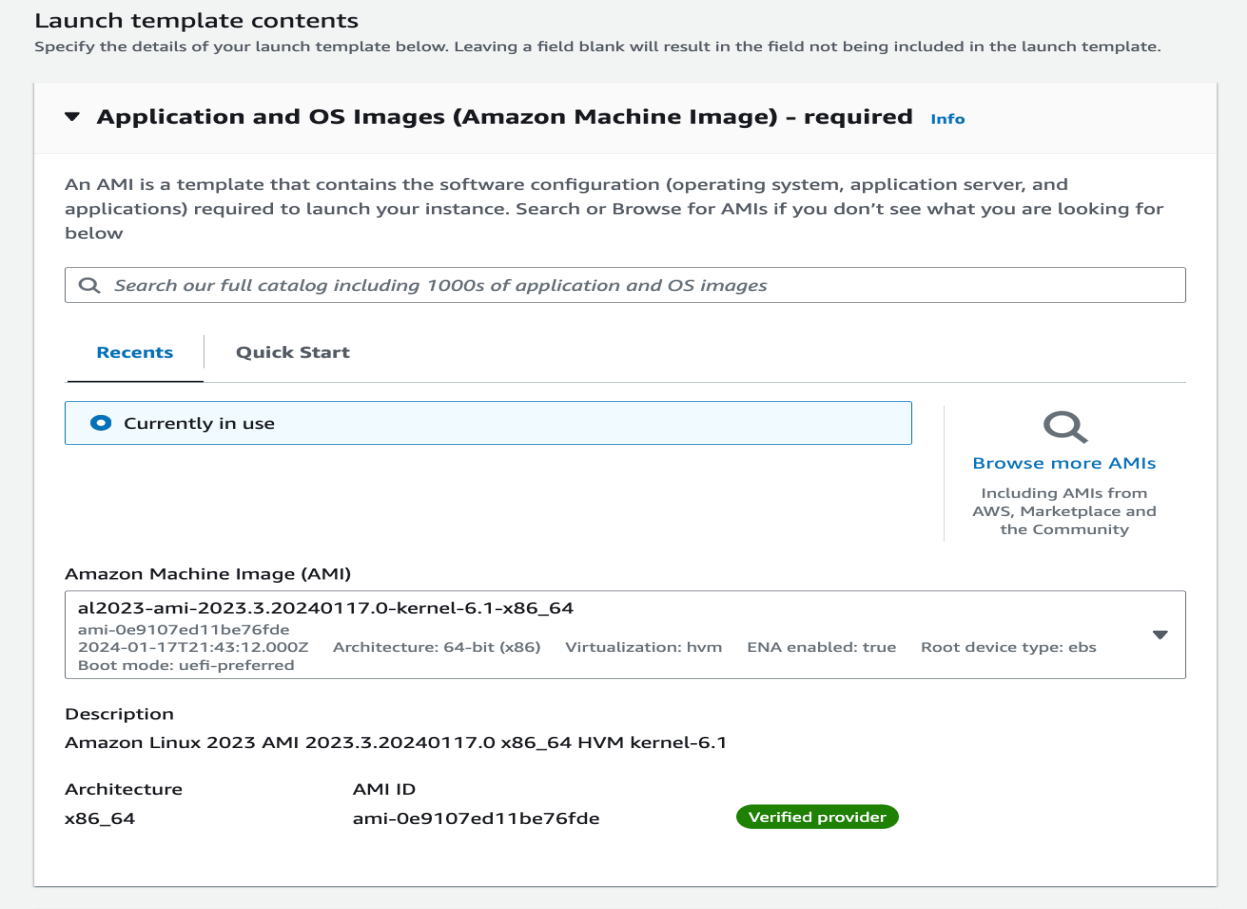


1.3.1. Launch Template

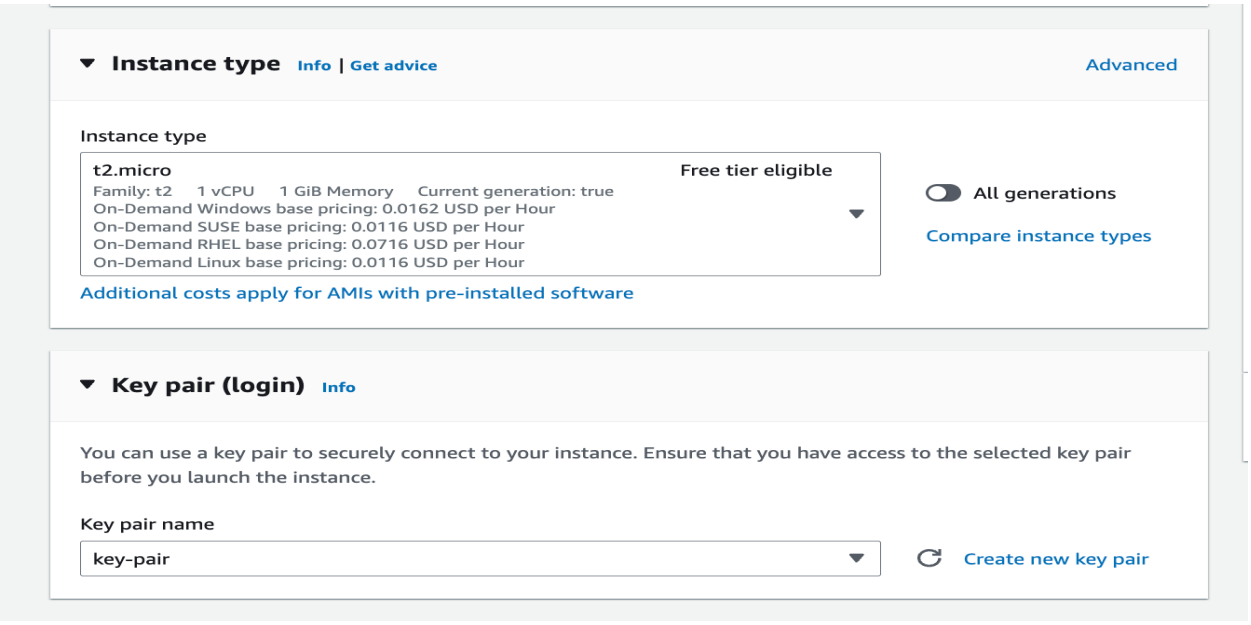
Required configuration is given to create launch template.



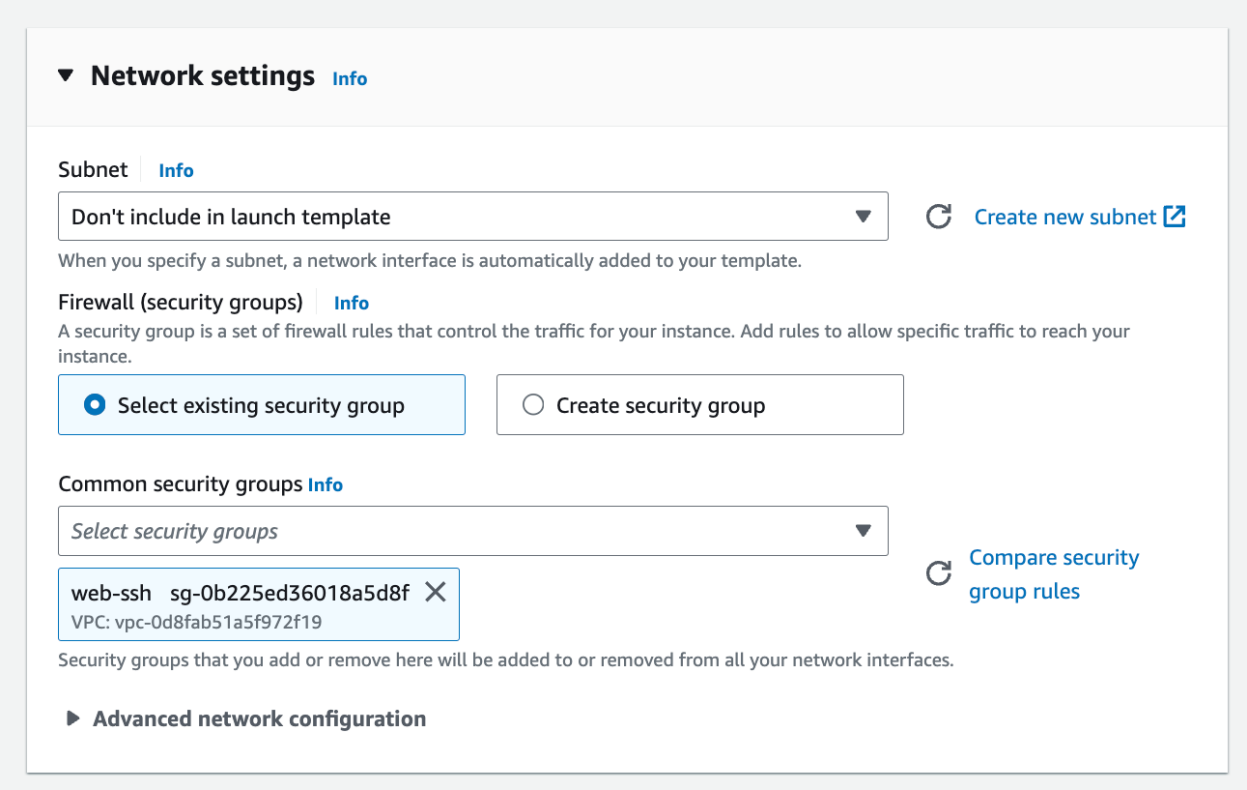
For Application and OS Images, currently in use is selected.



For instance type, t2.micro is selected and for key pair, existing key pair is selected.

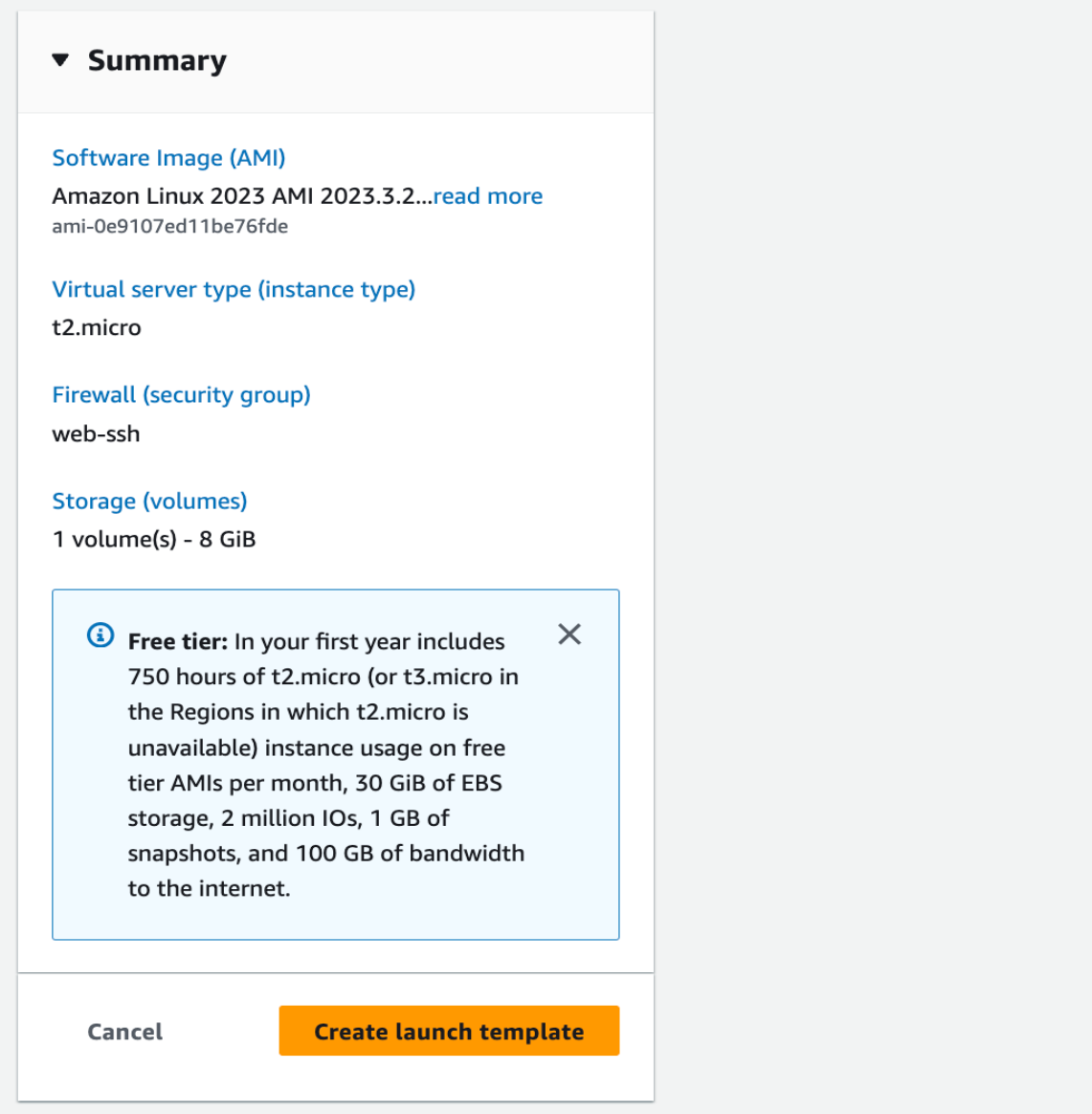


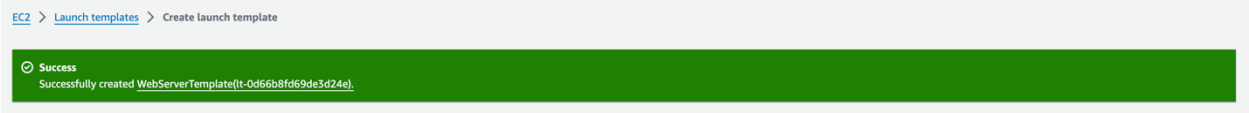
In network setting section existing security group is selected and other configurations are kept as default.



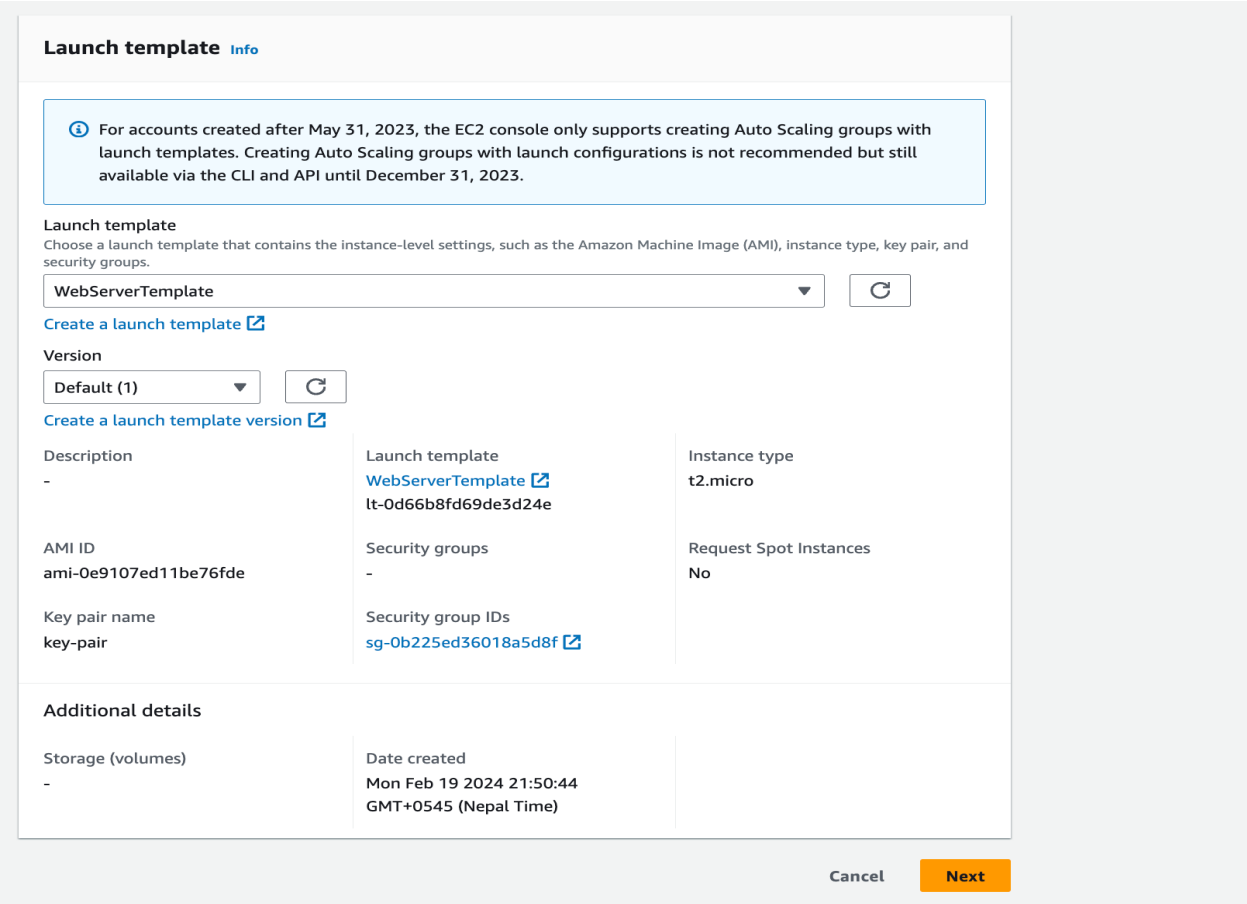
Bash script is added in advanced detail section.

After confirming the configuration, template is launched .

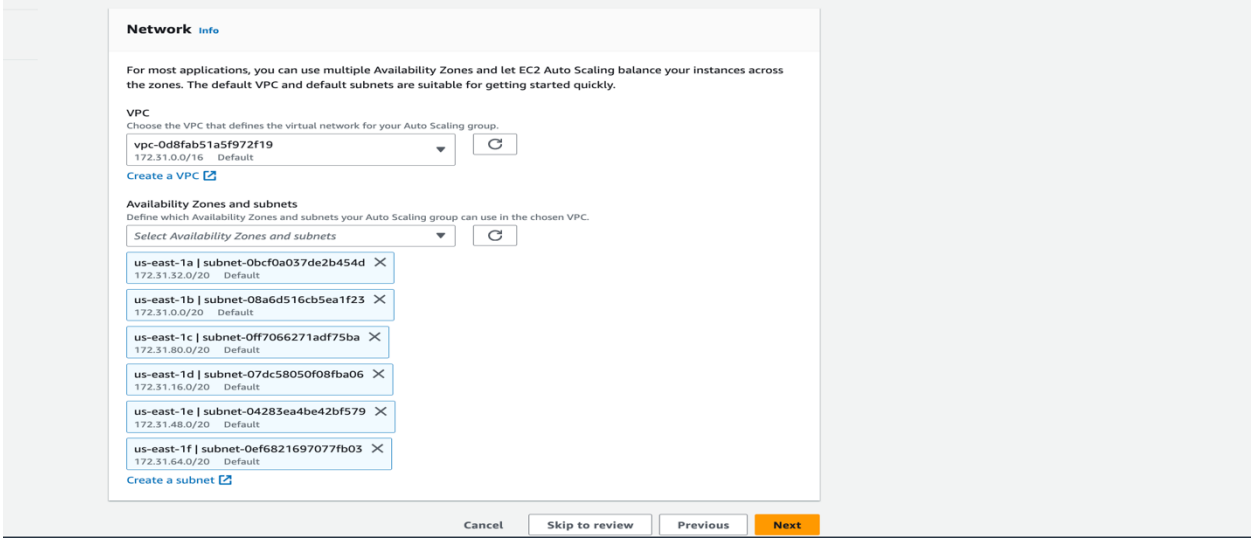




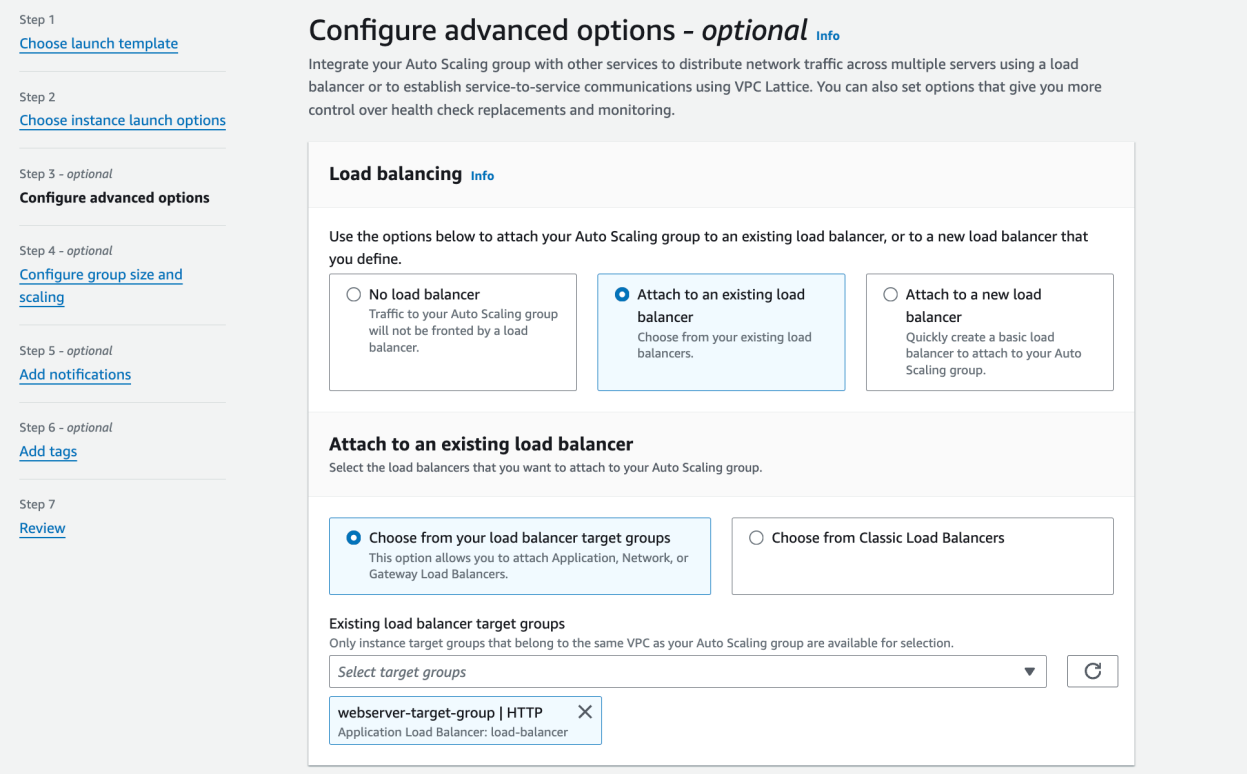
Now, the configuration of auto scaling is continued. Here newly created launch template is selected.



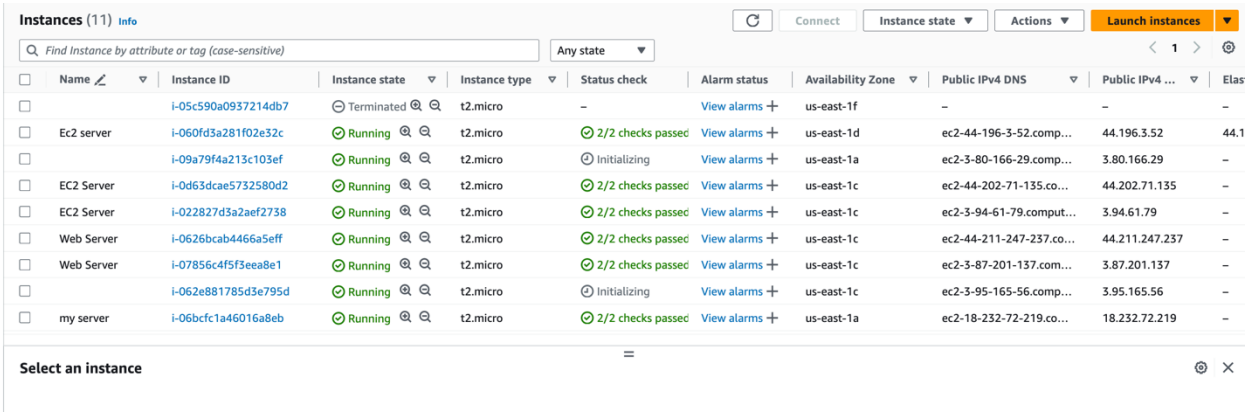
In network section default VPC and all the subnets are selected



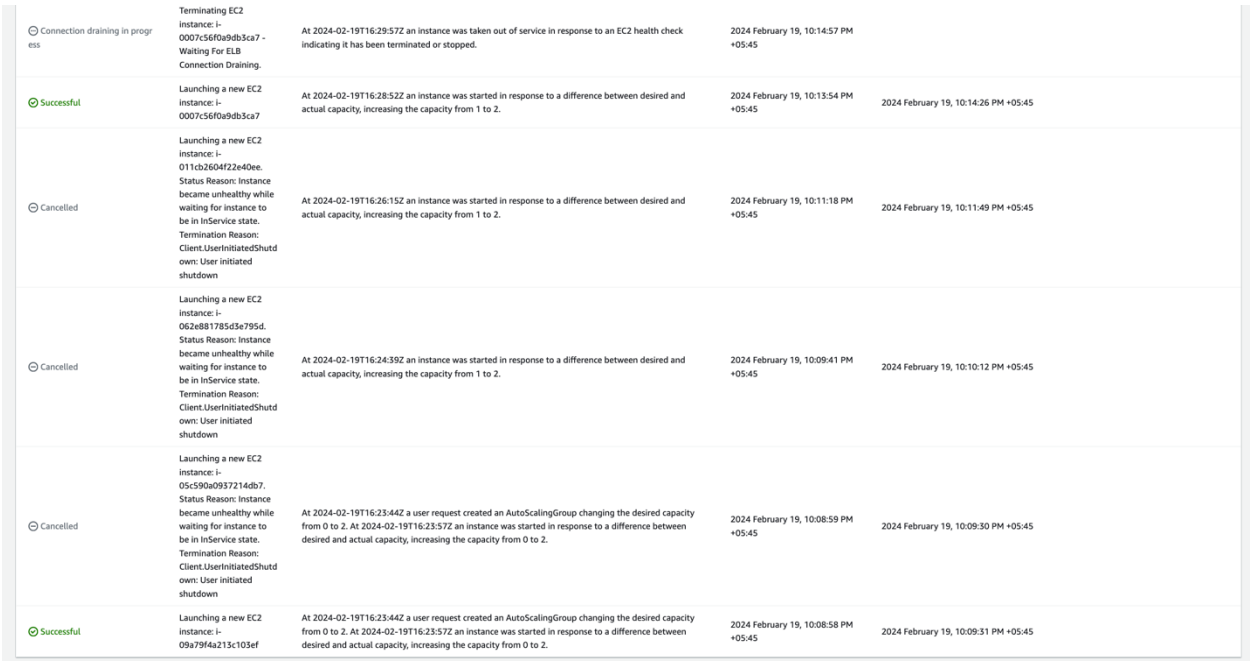
Advanced options configurations is done. Firstly, in load balancing configuration, existing load balancer is chosen and then it's target group is selected. Rest advanced configuration is kept as default.



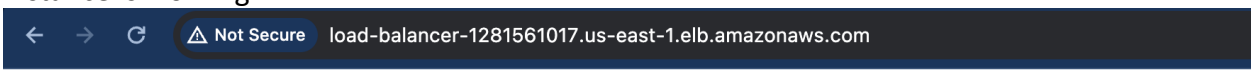
Now, group size and scaling are configured. Here, desired capacity of Group size is set as 2. In scaling, min desired capacity is set as 2 and max desired capacity as 4. After reviewing the configuration auto scaling group is created.



As we have defined 2 desired capacity, 2 EC2 instance of same target group is always maintained based on the created launch template. In the activity tab of auto scaling, we can observe the activity history of the EC2 instances.



When DNS of load balancer is refreshed even for multiple times the, it can be observed that EC2 instance is working.

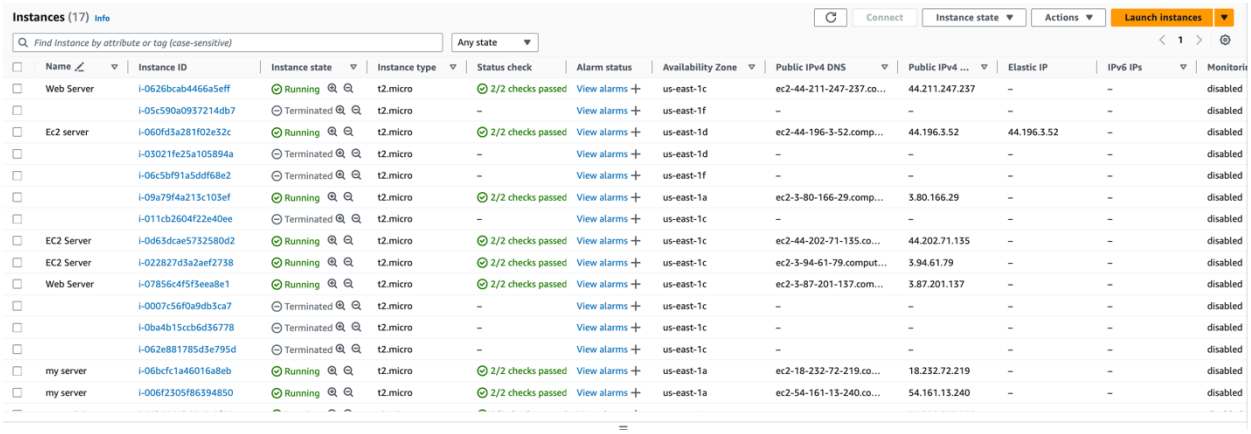


Hello from ip-172-31-91-151.ec2.internal

Hello from ip-172-31-43-204.ec2.internal

Hello from ip-172-31-83-149.ec2.internal

It can be observed that, how auto scaling adds and removes EC2 instances to manage the traffic.



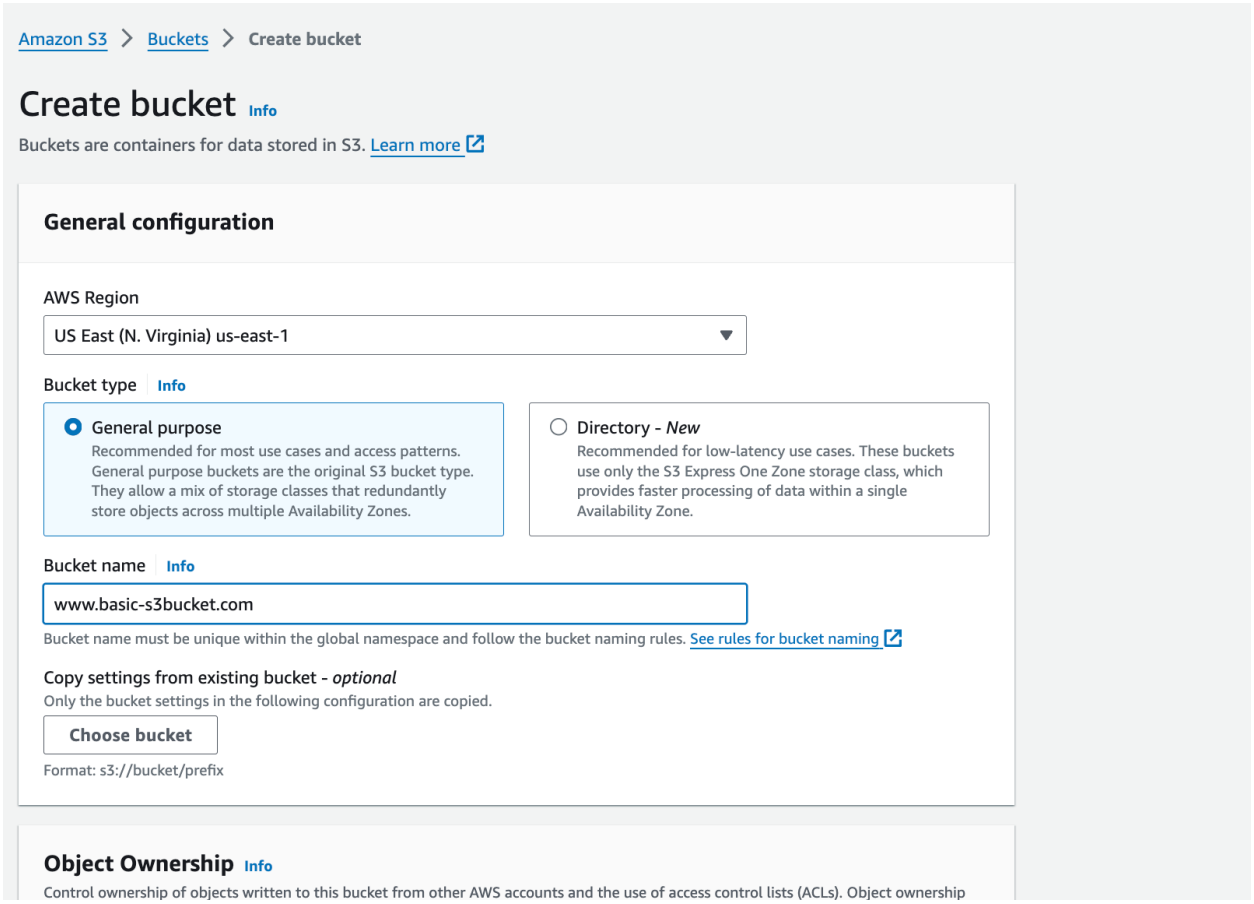
Part 2: Hosting a Static Portfolio Website on S3

Objective: Learn to host a static website (such as a personal portfolio) on Amazon S3.

Approach:

1. **Create an S3 Bucket**: Start by creating a new S3 bucket. Configure the bucket for website hosting, which includes setting permissions to make the content publicly accessible.
2. **Upload Website Files**: Upload the static files of your portfolio website (HTML, CSS, JavaScript, images) to the S3 bucket.
3. **Configure DNS**: Use Amazon Route 53 or another DNS service to point a domain name to the S3 bucket. This makes the website accessible via a user-friendly URL.
4. **Enable Additional Features** (Optional): Implement features like HTTPS for secure access and CloudFront for content delivery optimization.

Goal: Students will understand how to use S3 for hosting static websites, manage bucket permissions, and integrate with other AWS services for a complete web hosting solution.


Amazon S3 > Buckets > Create bucket
Create bucket Info
Buckets are containers for data stored in S3. Learn more
General configuration
AWS Region
US East (N. Virginia) us-east-1
Bucket type Info
General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.
Directory - New
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.
Bucket name Info
www.basic-s3bucket.com
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming
Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix
Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership

Amazon S3 > Buckets > www.basic-s3bucket.com > Edit static website hosting

Edit static website hosting Info

Static website hosting

Use this bucket to host a website or redirect requests. Learn more

Static website hosting

- ○ Disable
- ◉ Enable

Hosting type

- ○ Host a static website
  Use the bucket endpoint as the web address. Learn more
- ◉ Redirect requests for an object
  Redirect requests to another bucket or domain. Learn more

Host name

basic-s3bucket

Target bucket website address or personal domain

Protocol - *Optional*

- ○ none
- ◉ http
- ○ https

Cancel Save changes

Route 53 > Hosted zones > Create hosted zone

Create hosted zone Info

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name Info

This is the name of the domain that you want to route traffic for.

basic-s3bucket

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Description - *optional* Info

This value lets you distinguish hosted zones that have the same name.

The hosted zone is used for...

The description can have up to 256 characters. 0/256

Type Info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

- (•) **Public hosted zone**
  A public hosted zone determines how traffic is routed on the internet.
- ( ) **Private hosted zone**
  A private hosted zone determines how traffic is routed within an Amazon VPC.

Tags Info

Apply tags to hosted zones to help organize and identify them.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel | Create hosted zone

Route 53 > Get started

Get started Info

Choose your starting point

- Register a domain
  Register the name, such as example.com, that your users use to access your application.
  
- Transfer domain
  You can transfer domain names to Route 53 that you registered with another domain registrar.
  
- Create hosted zones
  A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.
  
- Configure health checks
  Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.
  
- Configure traffic flow
  A visual tool that lets you easily create policies for multiple endpoints in complex configurations.
  
- Configure resolvers
  A regional service that lets you route DNS queries between your VPCs and your network.
  

Cancel Get started