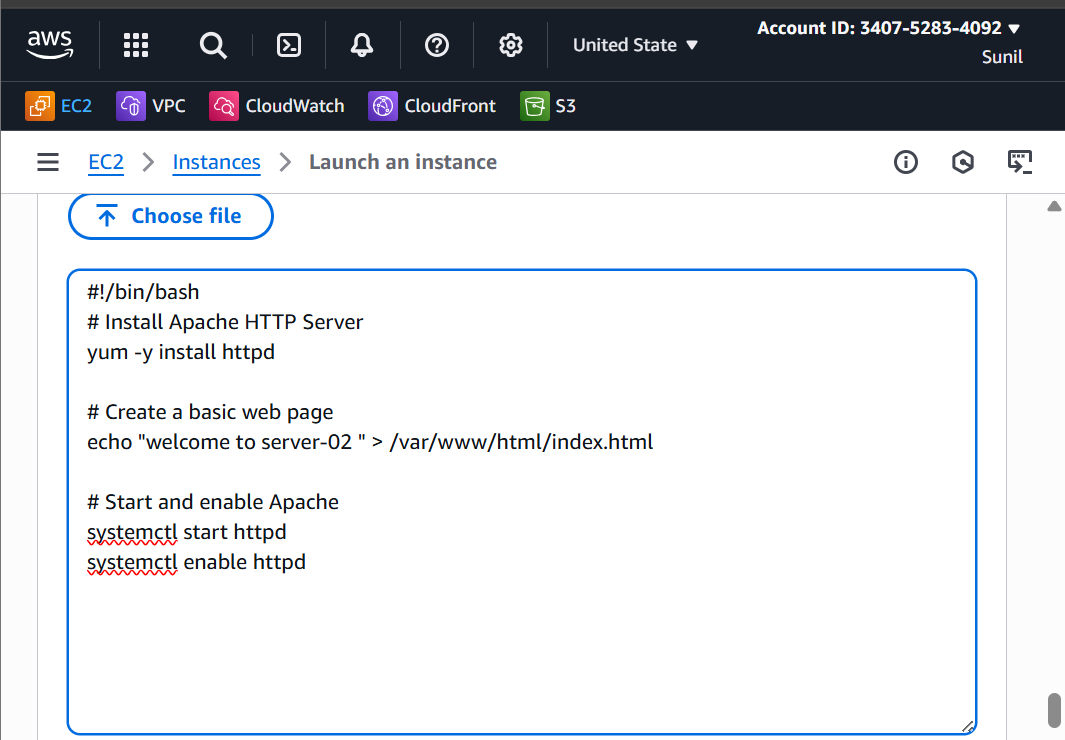
**Load balancers.**

**1) Configure Classic Load balancer.**

Step 1:

**Launch two EC2 instances** running in the same AWS region but in **different Availability Zones**.

EC2 instances must have a **web server**

****

Step 2:

**Create Load Balancer**.

Choose Load Balancer Type

Select **Classic Load Balancer**. And click create

Step 3 — Configure Basic Settings

Choose Internet-facing (for public access) or Internal (for private access).

**Listeners:**

Default HTTP: HTTP 80 → HTTP 80

**VPC:** Select the VPC where your EC2 instances are located. **Availability Zones:** Select at least **two AZs** and their subnets.

Step -4

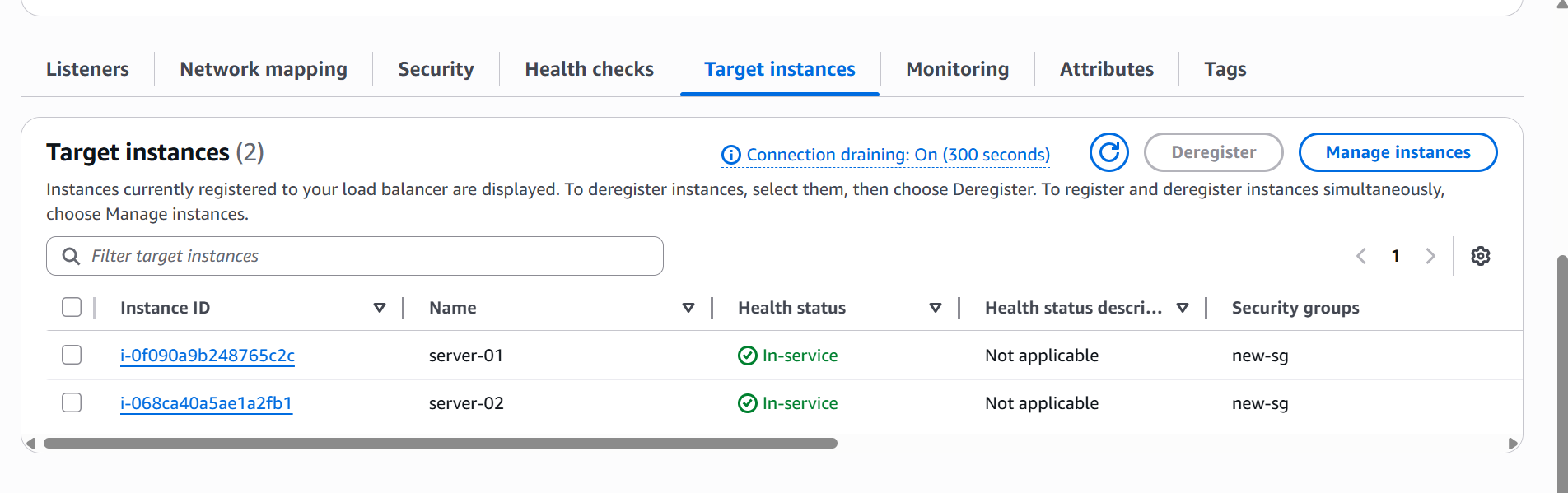
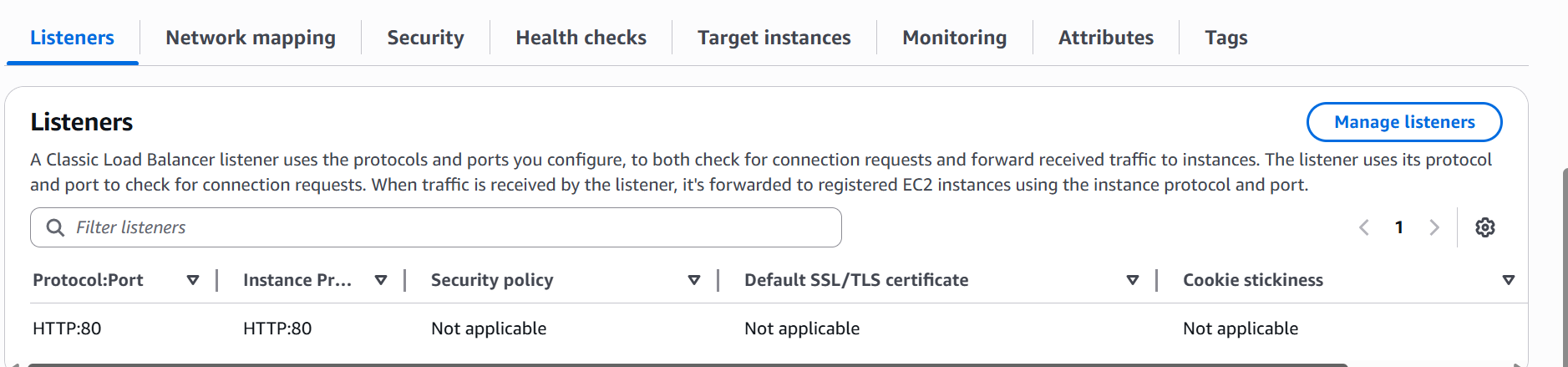
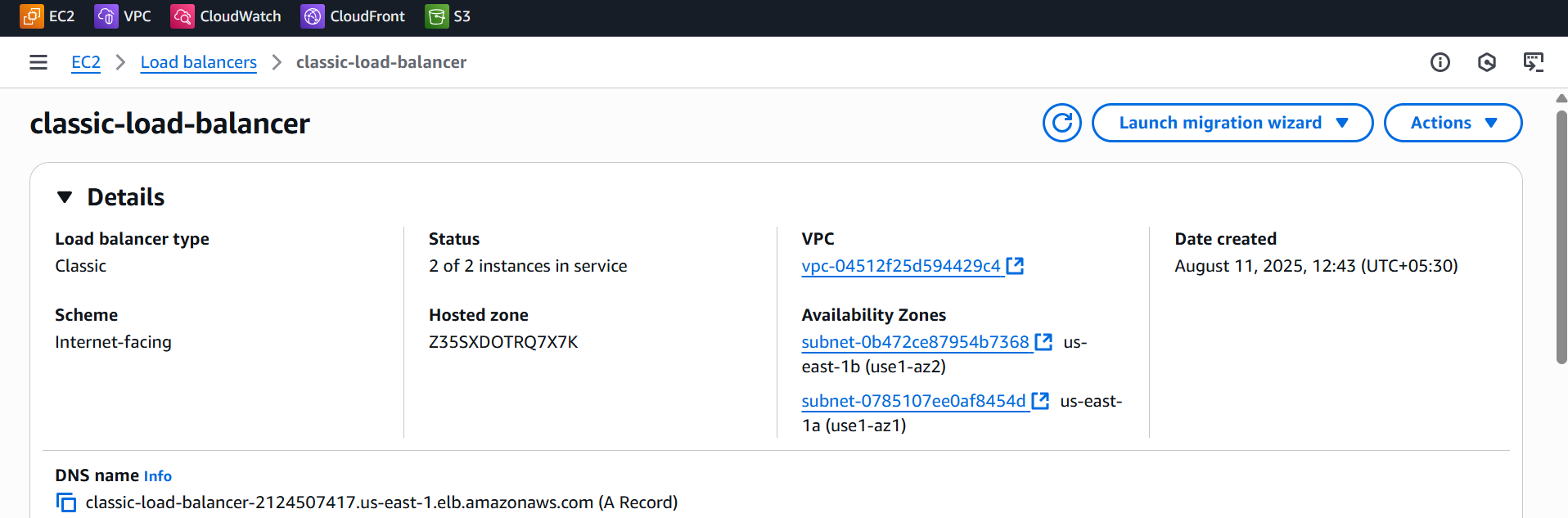
Register EC2 Instances

Select the EC2 instances you want behind the load balancer.

Ensure these instances are in the **running** state and have the web server running.

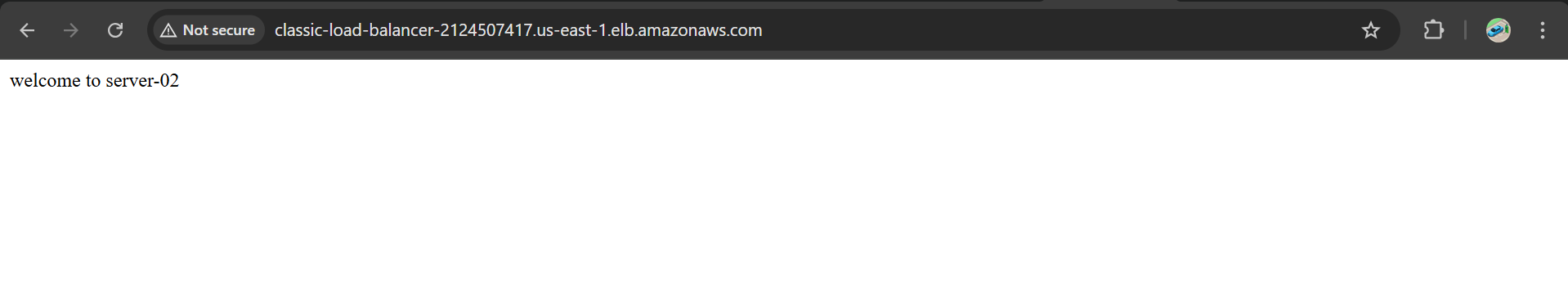
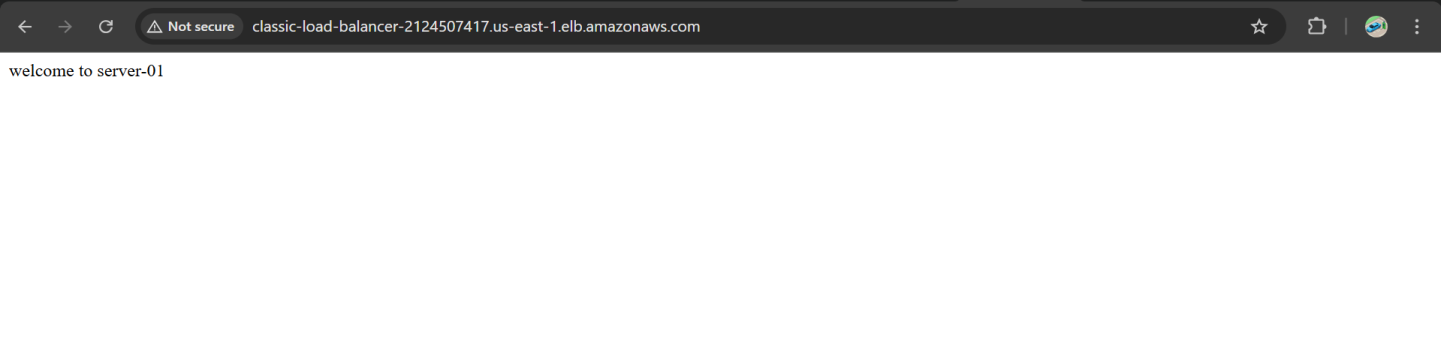
Review settings → Click **Create**.

Wait for AWS to provision the load balancer.



Step 5 — Test

Once the LB is **Active**, note its **DNS name** (ex:- ALB-02-1330657953.us-east-1.elb.amazonaws.com).

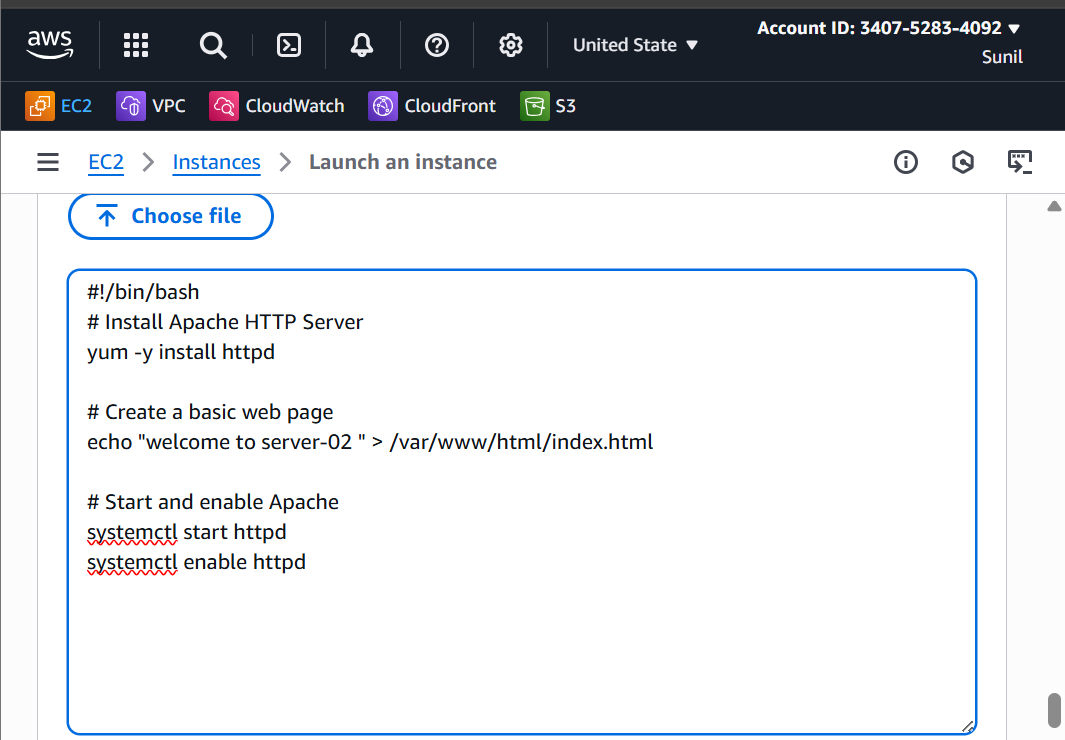
Open it in a browser — it should serve responses from your EC2 instances

**2) Configure Application Load balancer.**

Step 1:

**Launch two EC2 instances** running in the same AWS region but in **different Availability Zones**.

EC2 instances must have a **web server**

****

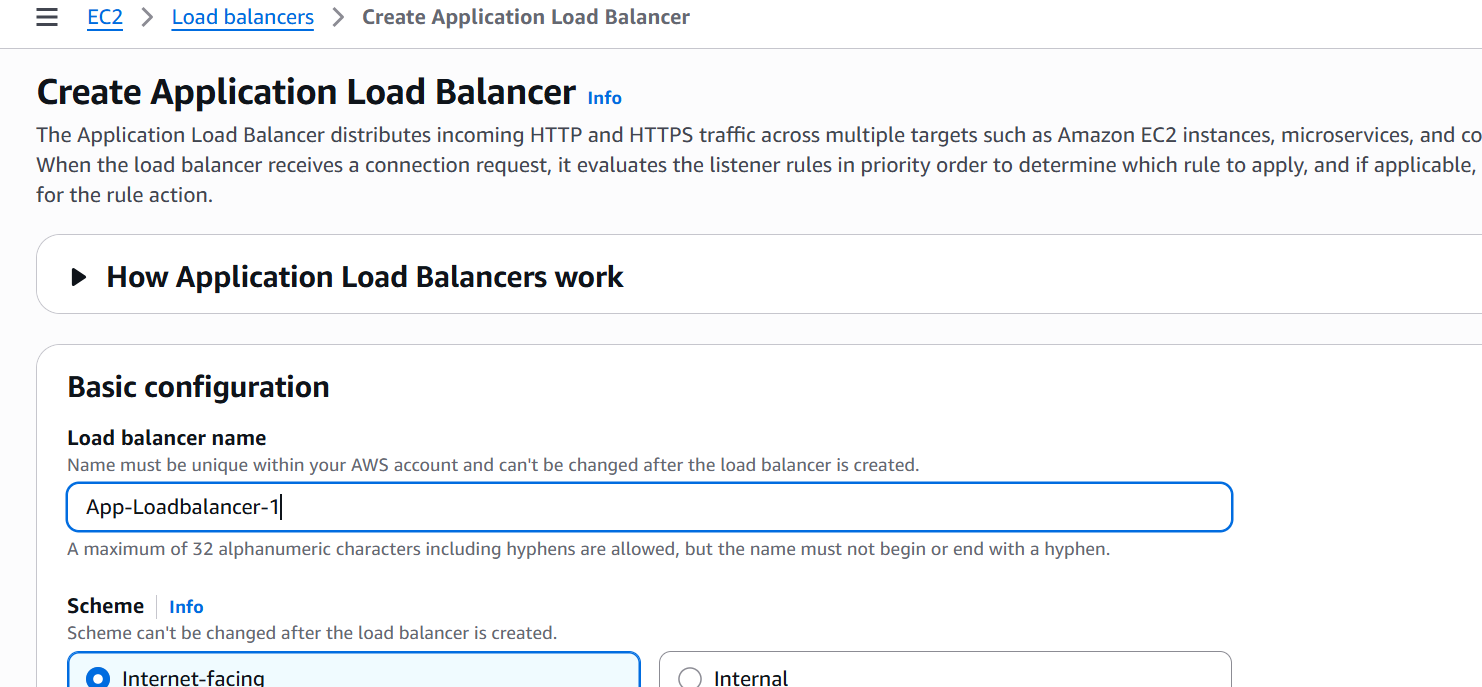
Step 2:

**Create Load Balancer**.

Choose Load Balancer Type

Select **Application Load Balancer**.

Click **Create**.



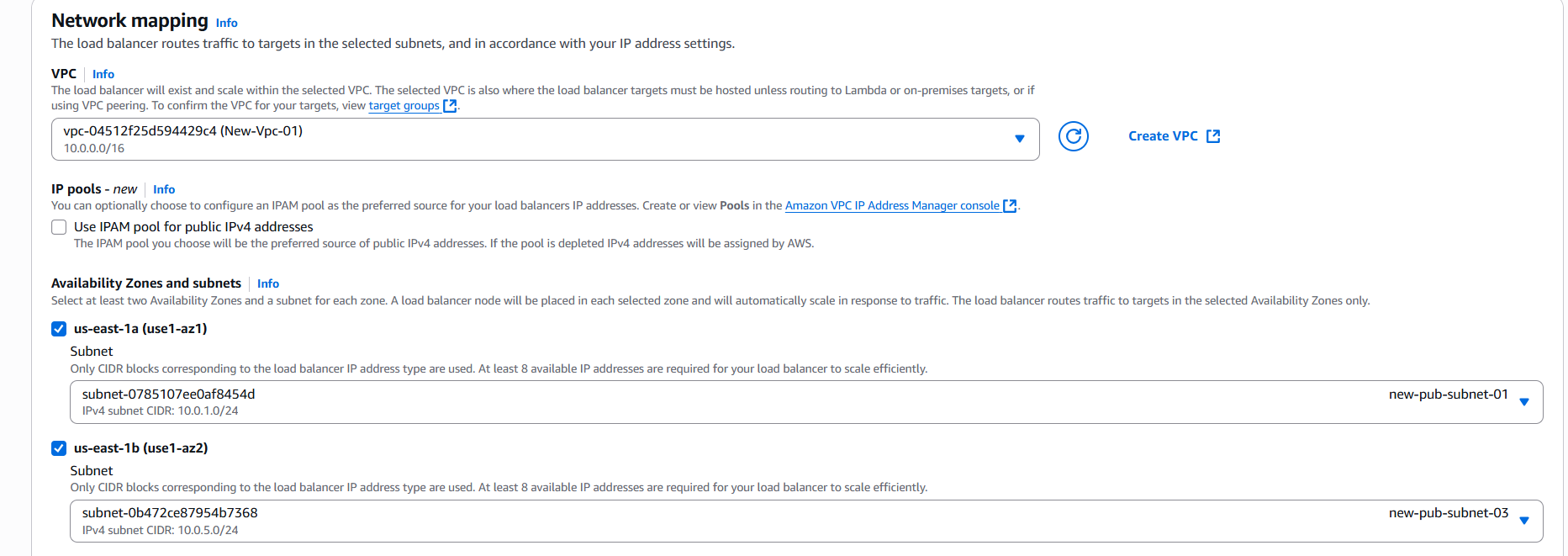
Step 3 — Basic Configuration

* Internet-facing *(for public traffic)*

**Listeners:** HTTP (port 80)

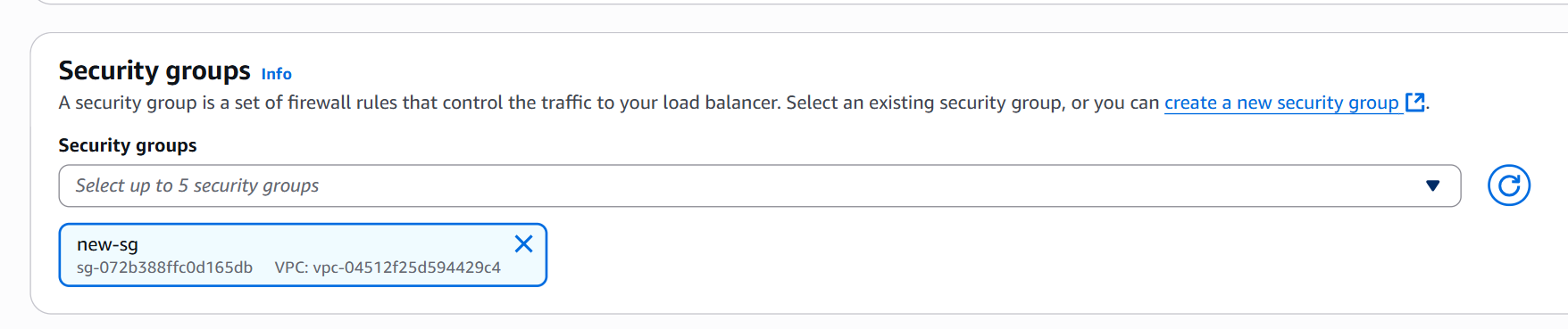
**VPC: Select your VPC.**

**Availability Zones: Select at least two subnets in different AZs.**

****

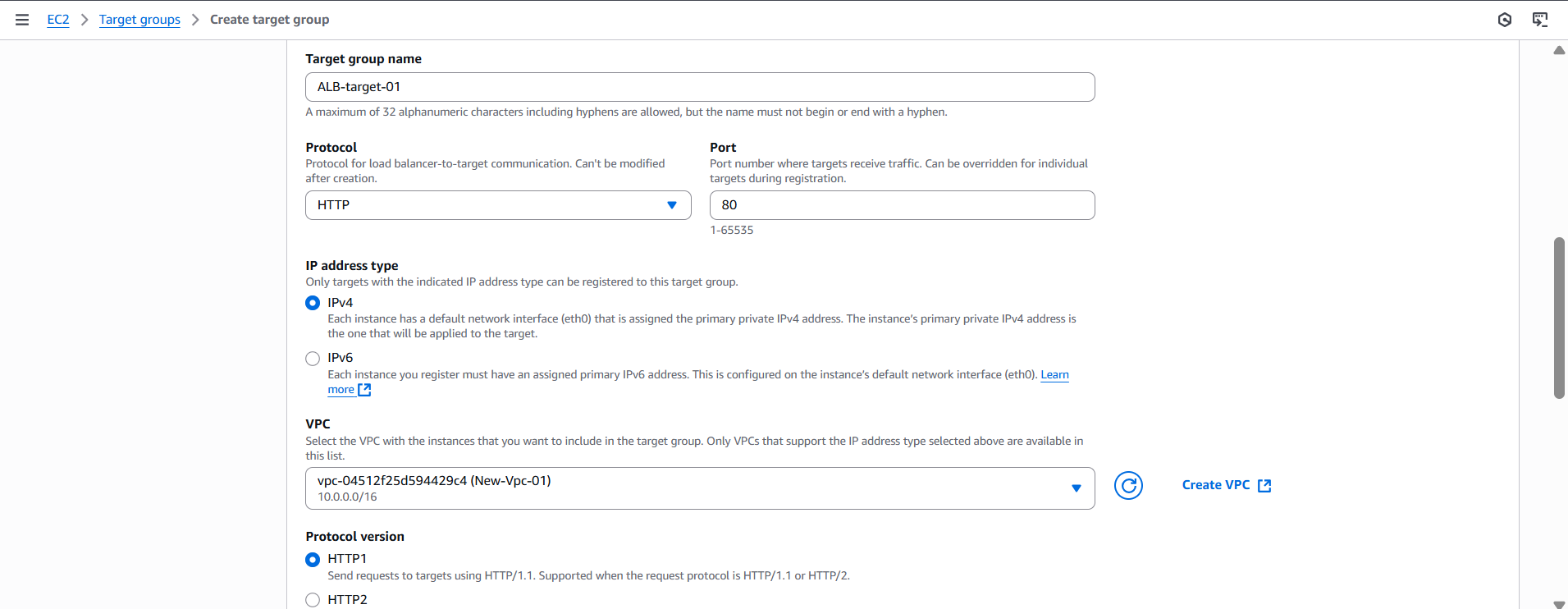
## **Step 4 — Security Groups**

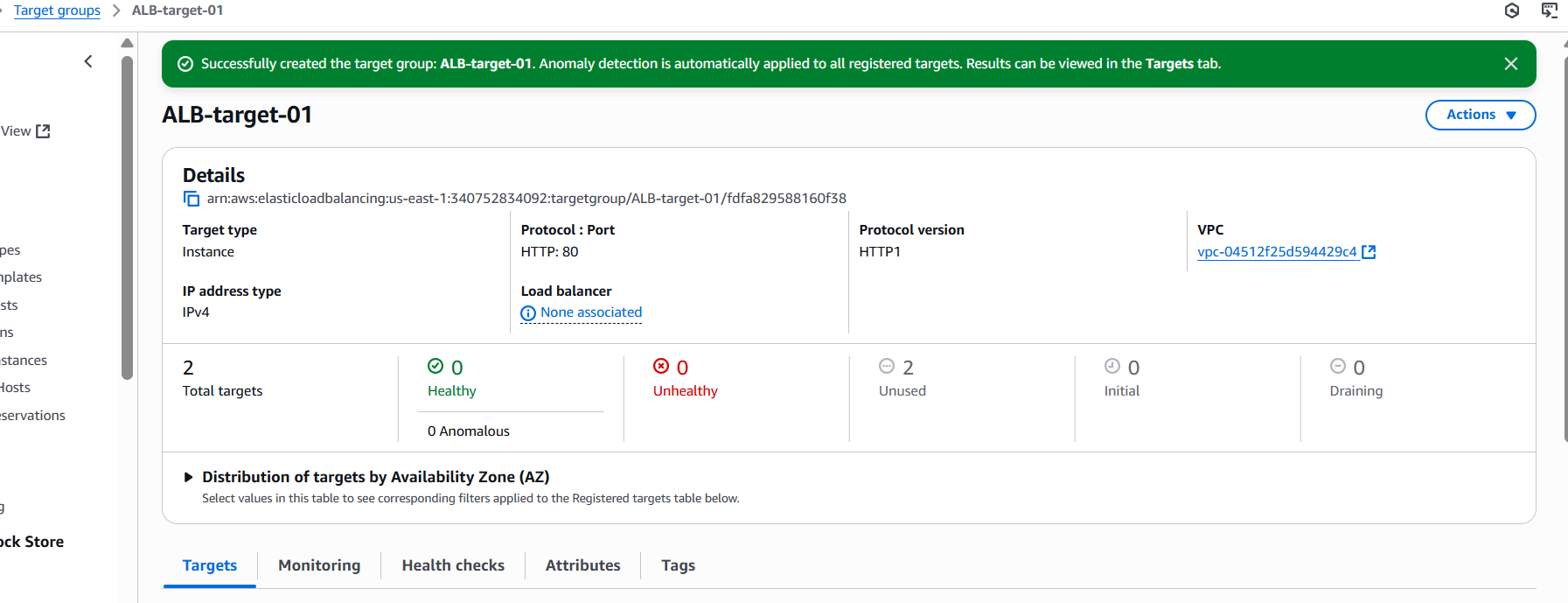
* Assign a Security Group that allows:
  + Inbound: Port 80 (and 443 if HTTPS).
  + Outbound: Allow to your EC2 instances.

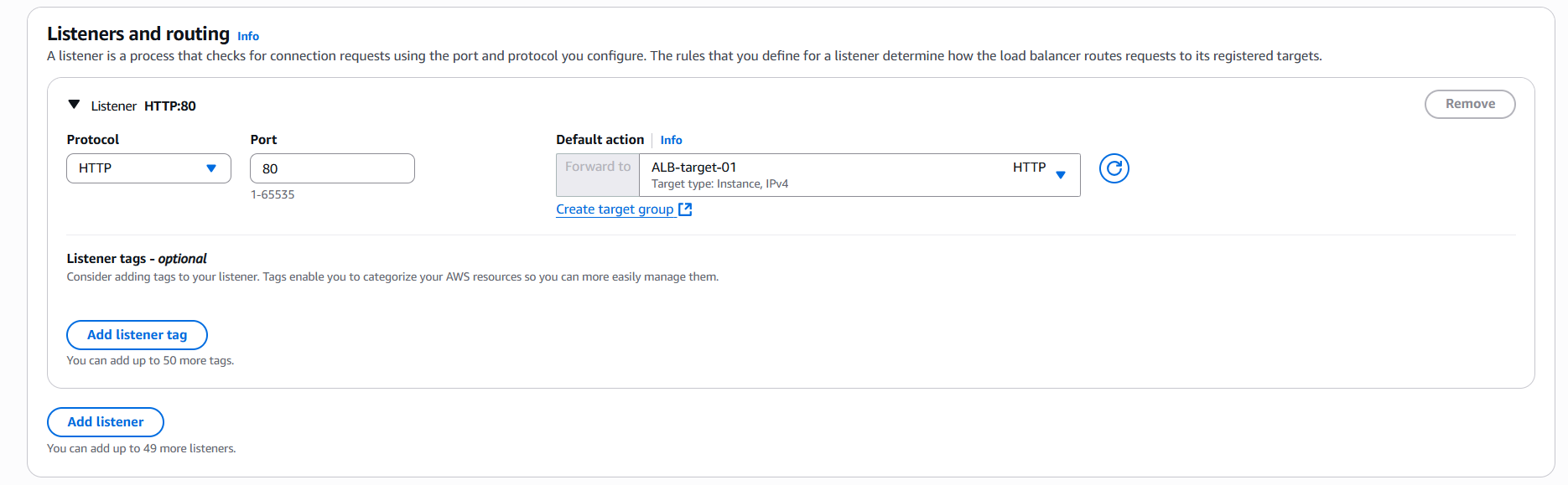


## **Step 5 — Create Target Group**

* **Target group name:** my-target-group.
* **Target type:** instance
* **Protocol:** HTTP.

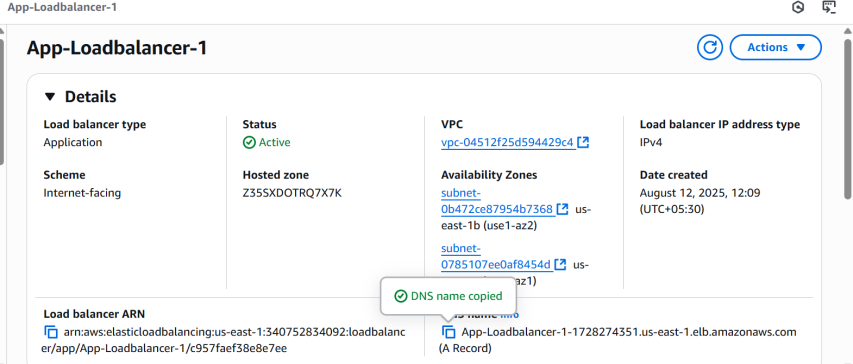






## **Step 6— Review and Create**

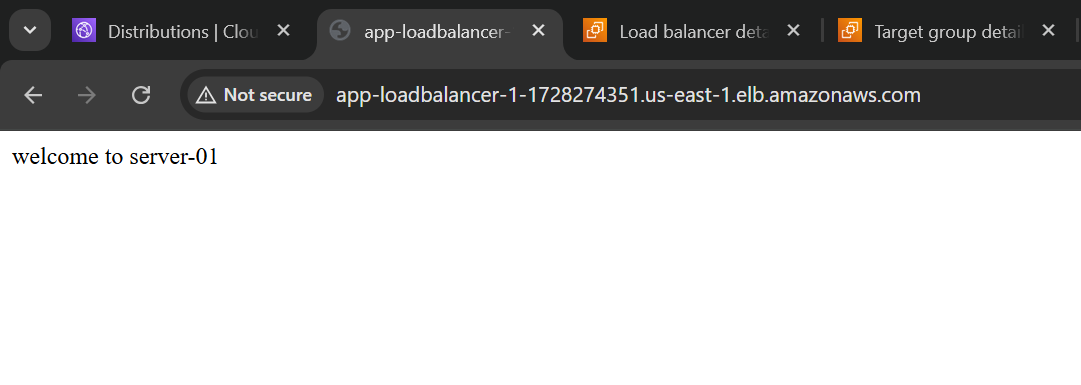
* Review the configuration.
* Click **Create load balancer**.
* Wait until the ALB’s status changes to **Active**.

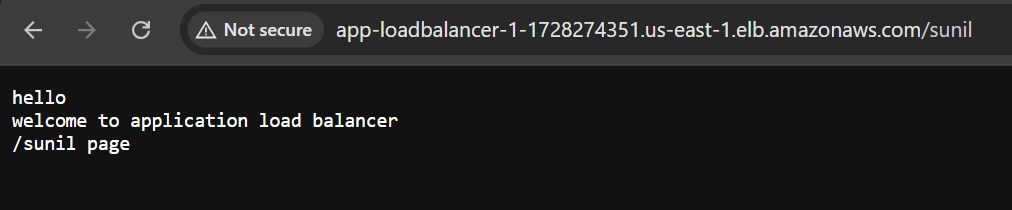


**Step 7 — Test**

**Copy the ALB DNS name ( App-Loadbalancer-1-1728274351.us-east-1.elb.amazonaws.com).**

Open it in your browser → you should see responses from your EC2 instances.

****

****

**3) Configure Network Load balancer.**

Step 1 —

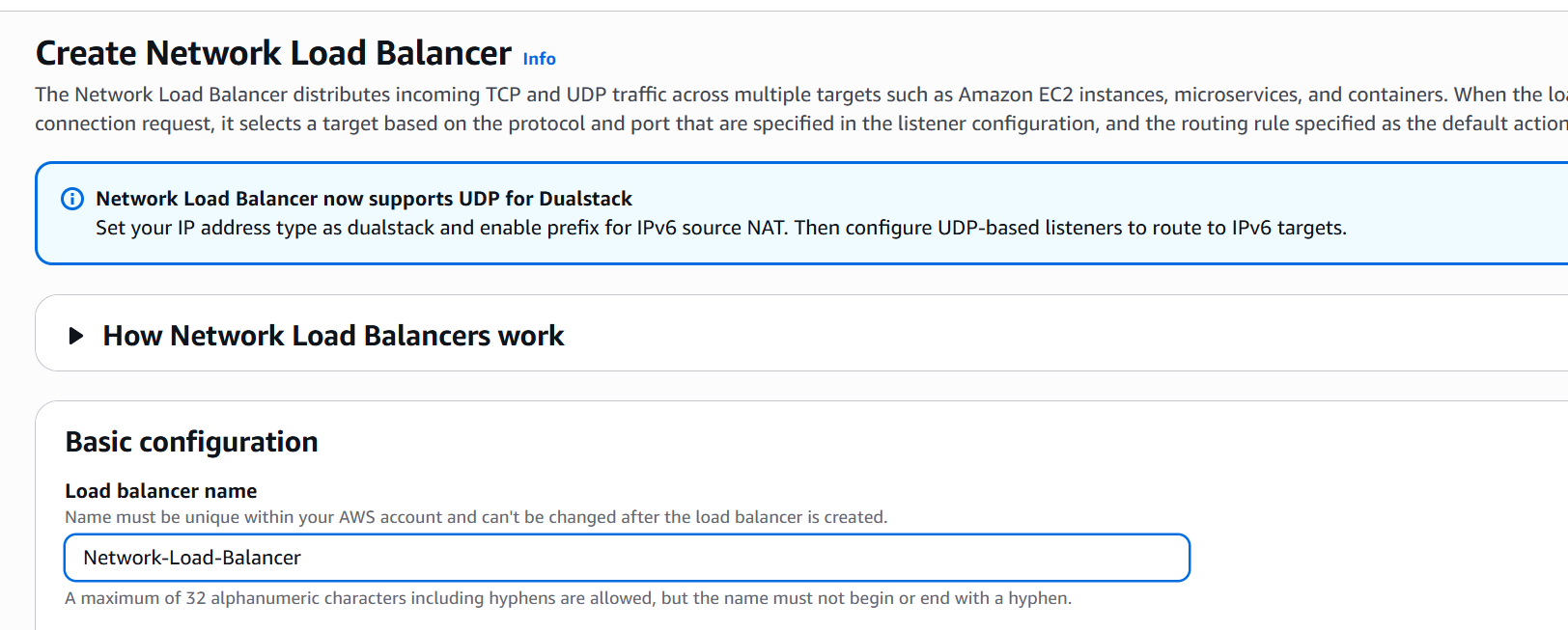
Before creating an NLB, make sure you have:

**Two or more EC2 instances** in the same VPC, ideally in **different Availability Zones**.

## **Step 2 — Go to Load Balancers**

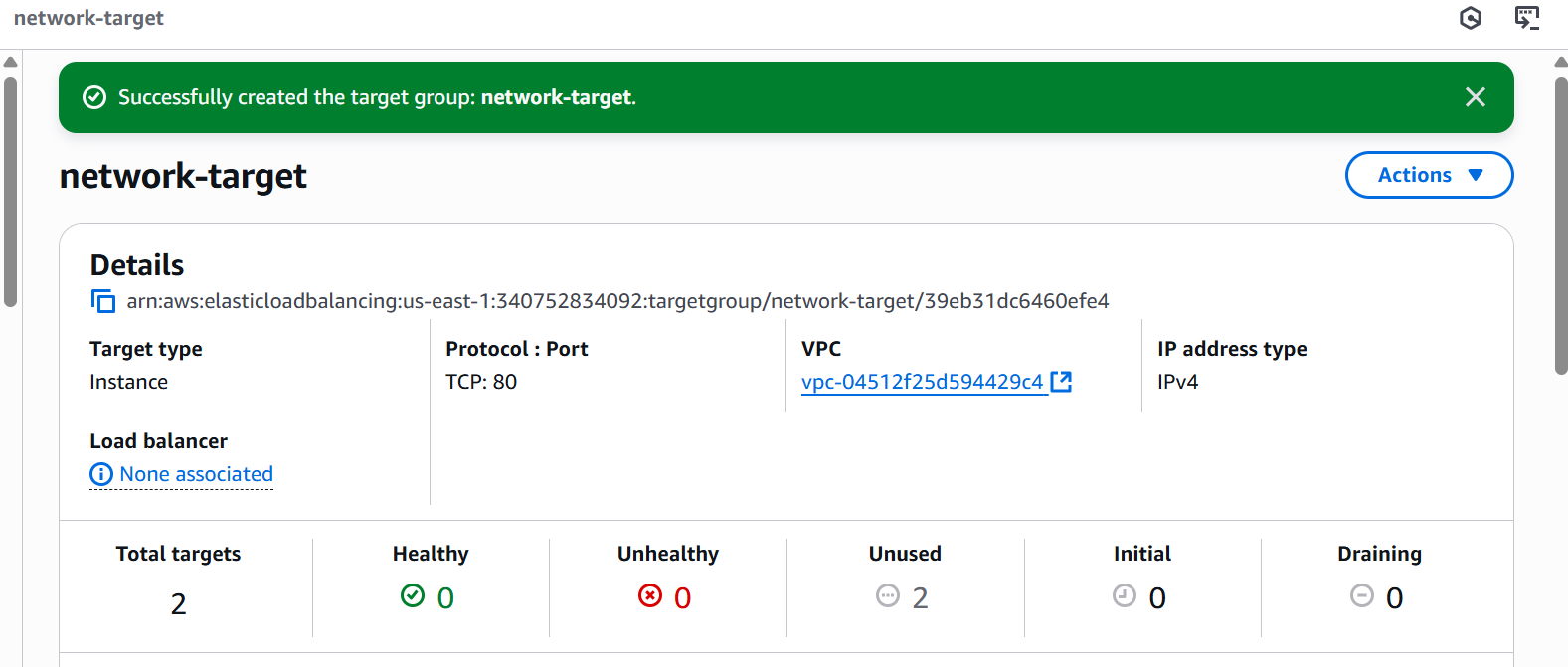
1. Open **AWS Console** → **EC2 service**.
2. On the left panel, click **Load Balancers**.
3. Click **Create Load Balancer**.

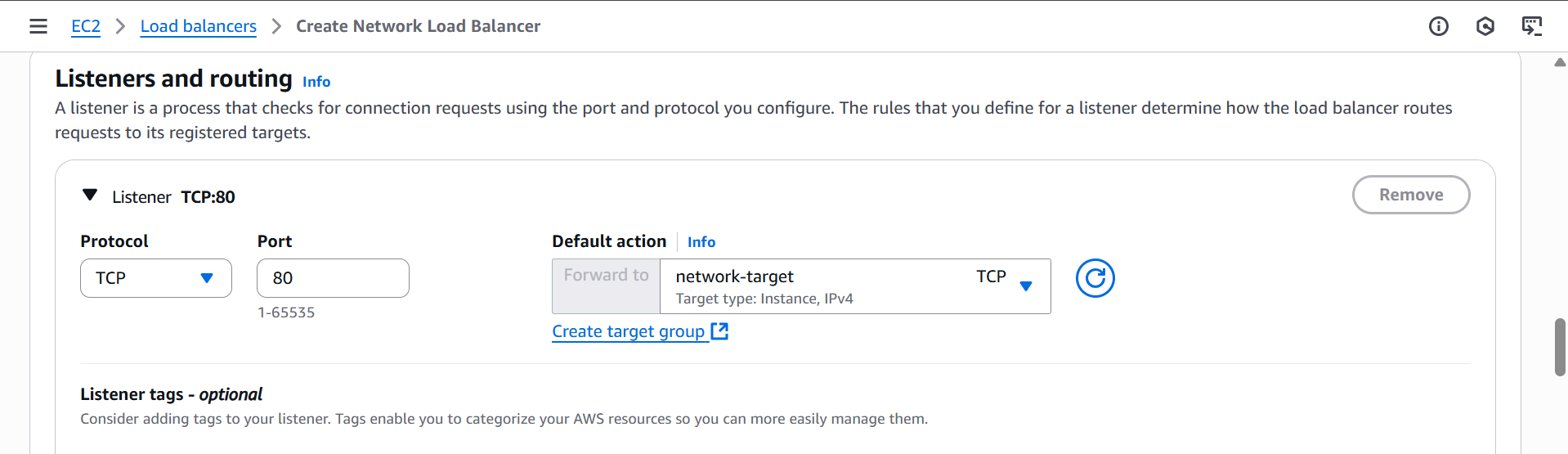
## **Step 3 — Choose Load Balancer Type**

* Select **Network Load Balancer**.
* Click **Create**.
* 

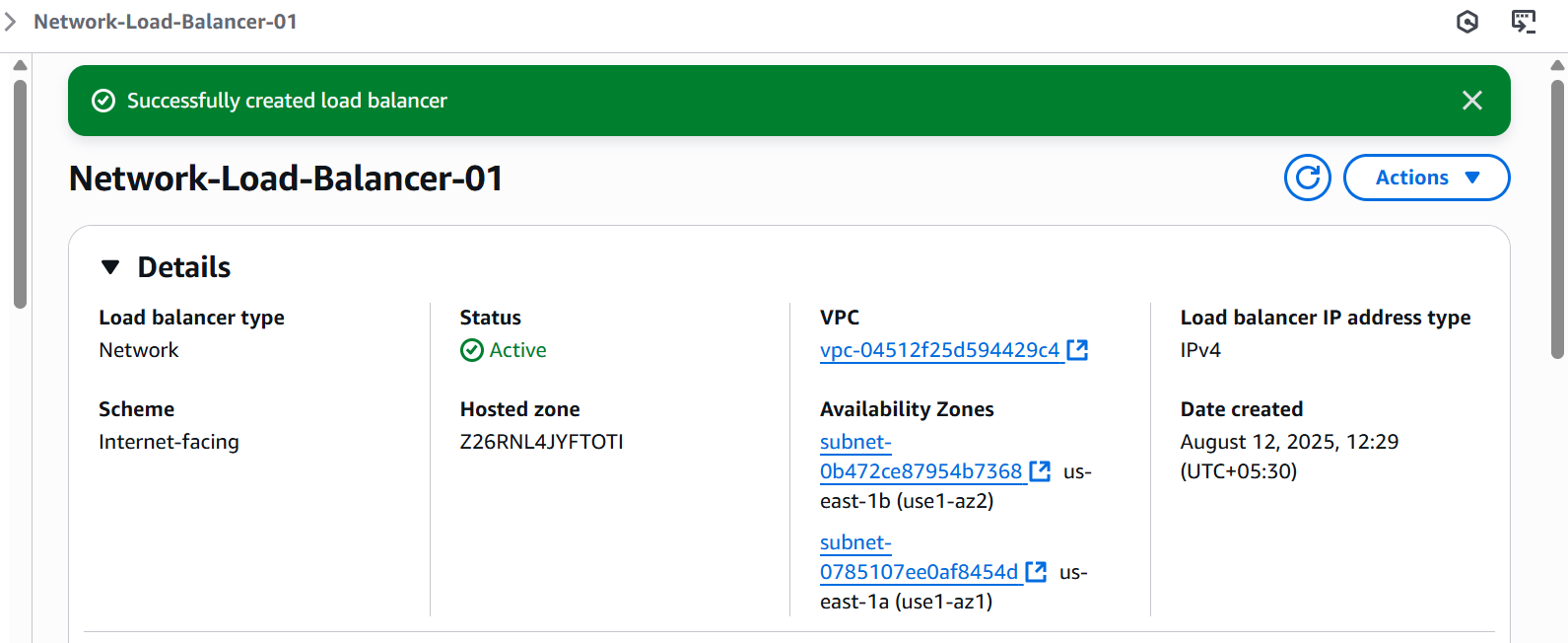
## **Step 4 — Create Target Group**

* **Target group name:** network-target
* **Target type:** Instance



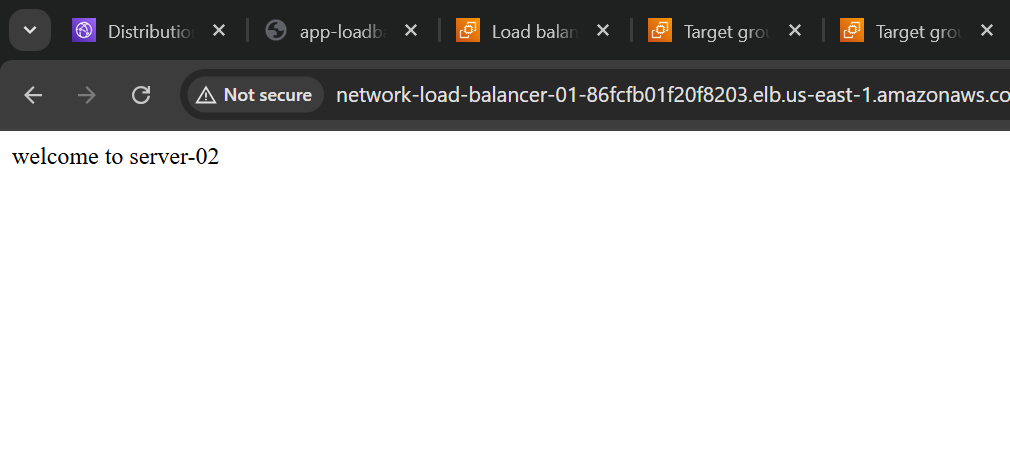


## **Step 5— Review and Create**

* Review all settings.
* Click **Create load balancer**.
* Wait until its status is **Active**.
* 

## **Step 6 — Test**

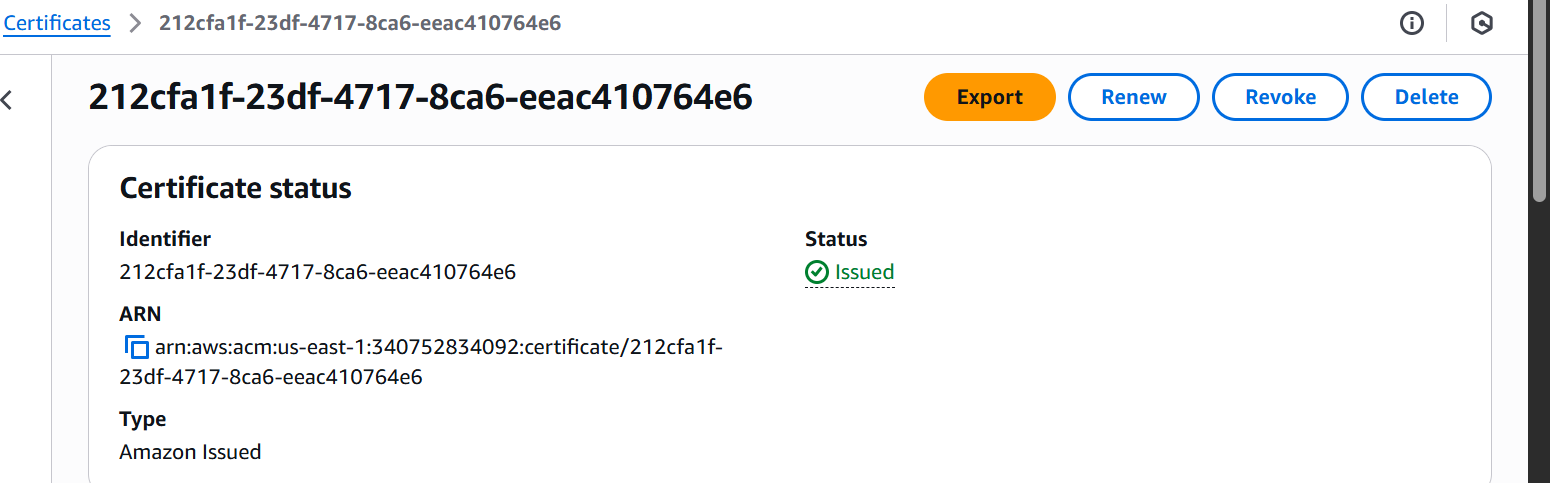
* Copy the **DNS name** of the NLB (e.g., Network-Load-Balancer-01-86fcfb01f20f8203.elb.us-east-1.amazonaws.com).
* Connect to it using your service’s protocol:



**4) Attach SSL for application load balancer.**

### ****Get an SSL/TLS Certificate in ACM****

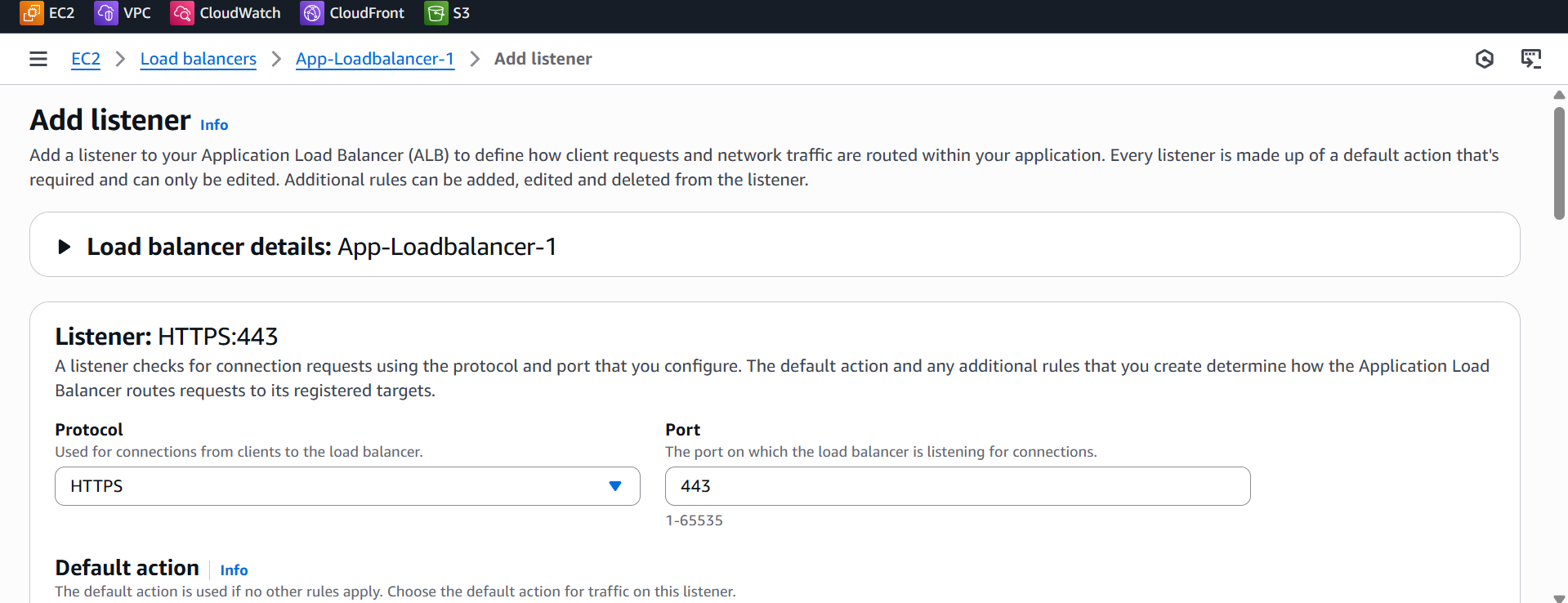
1. **Go to AWS Certificate Manager (ACM) in the AWS Console.**
2. **Click Request a certificate → Request a public certificate.**
3. **Enter your domain name(s) (e.g.,: sunilcloud.shop).**
4. Wait until the certificate status shows **Issued**.



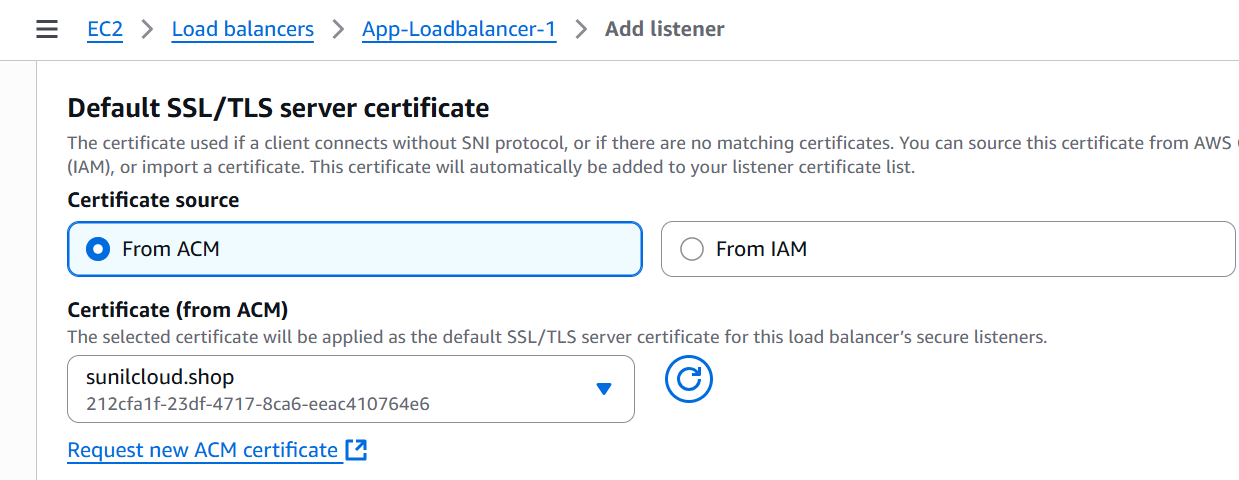
### ****Attach SSL Certificate to ALB****

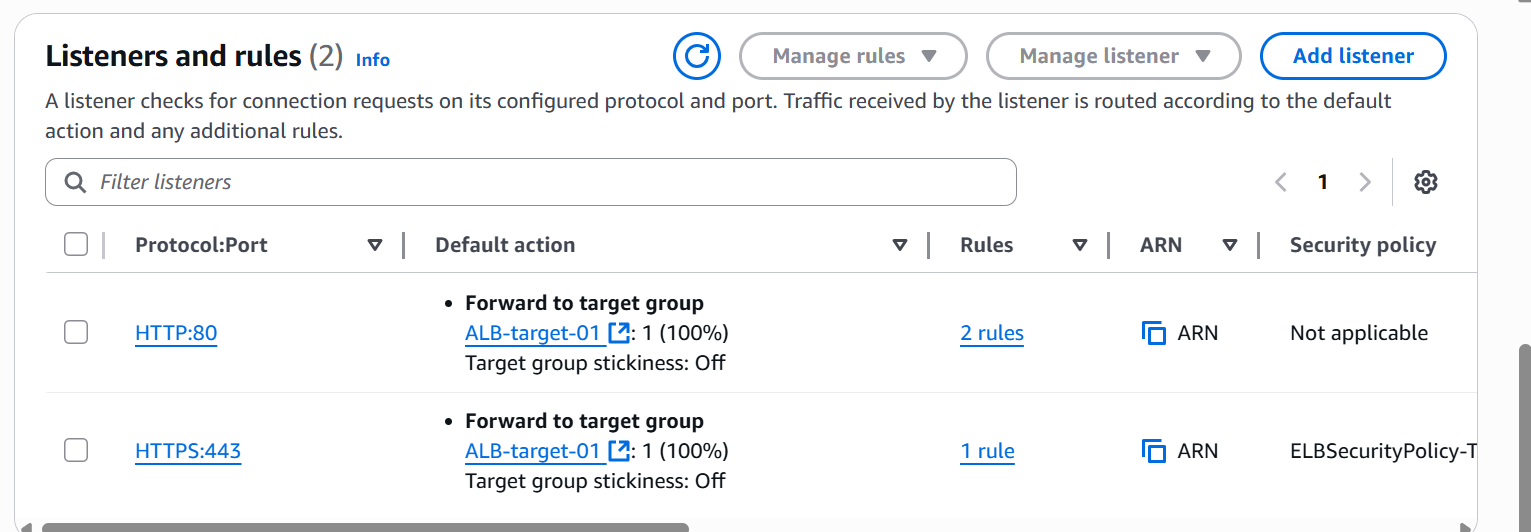
1. Go to **EC2 → Load Balancers** in AWS Console.
2. Select your **Application Load Balancer**.
3. Go to the **Listeners** tab.
4. If you don’t have an HTTPS listener:

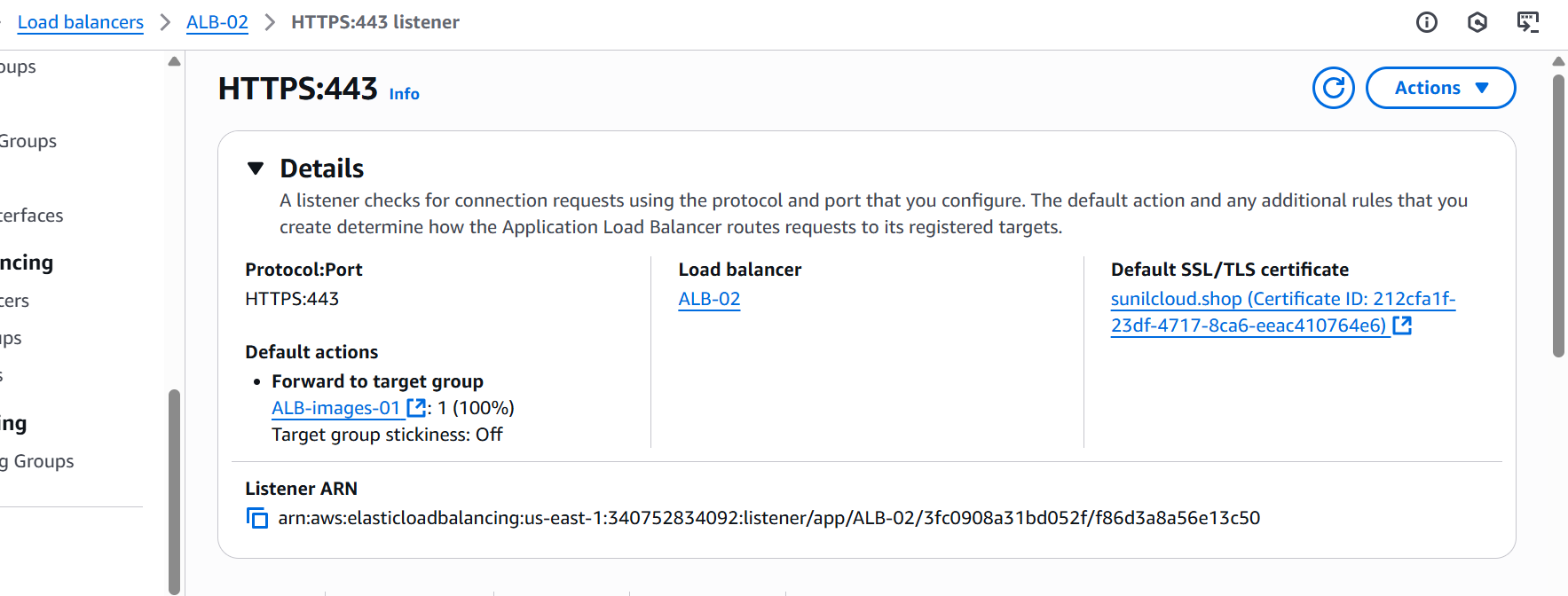
Click **Add listener** → **HTTPS (Port 443)** → **Add action** (forward to your target group).



1. Under **Default SSL Certificate**, choose the ACM certificate you created.
2. Save the changes.





****

**5) Map Application load balancer to R53.**

Step-1 Go to **Route 53 → Hosted zones**.

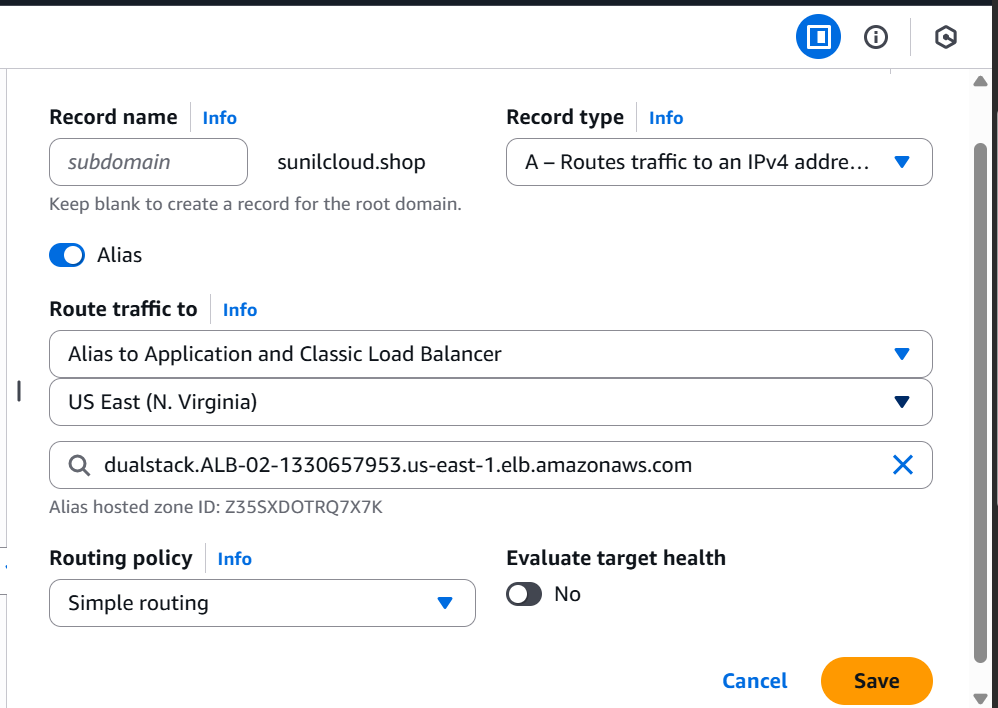
Click your domain name (e.g., sunilcloud.shop).

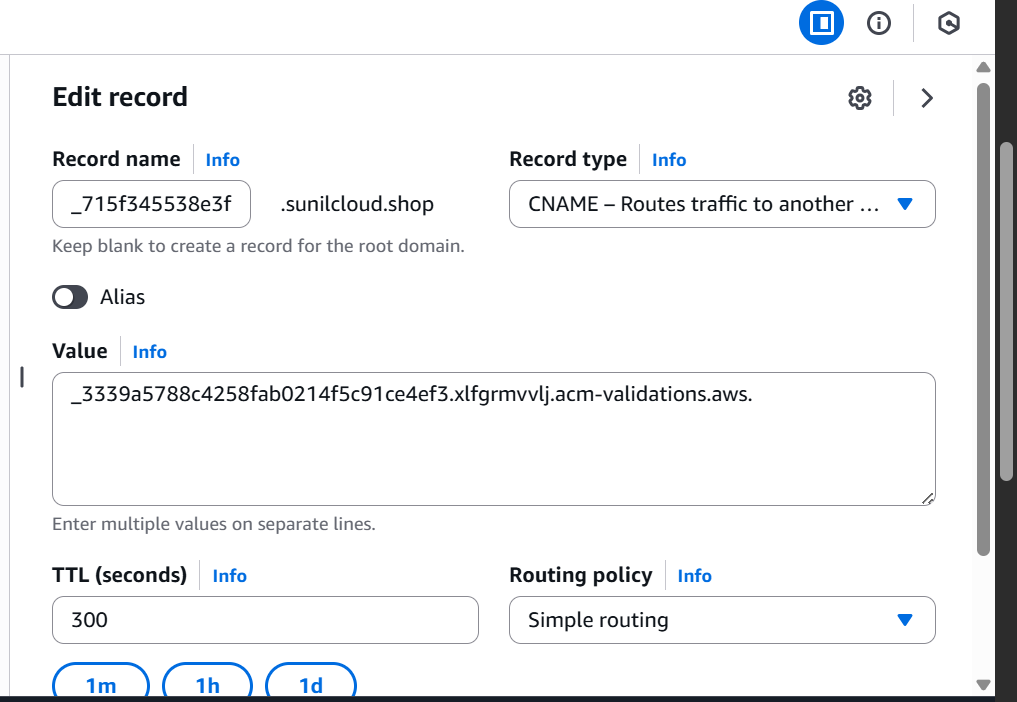
**Create record**.

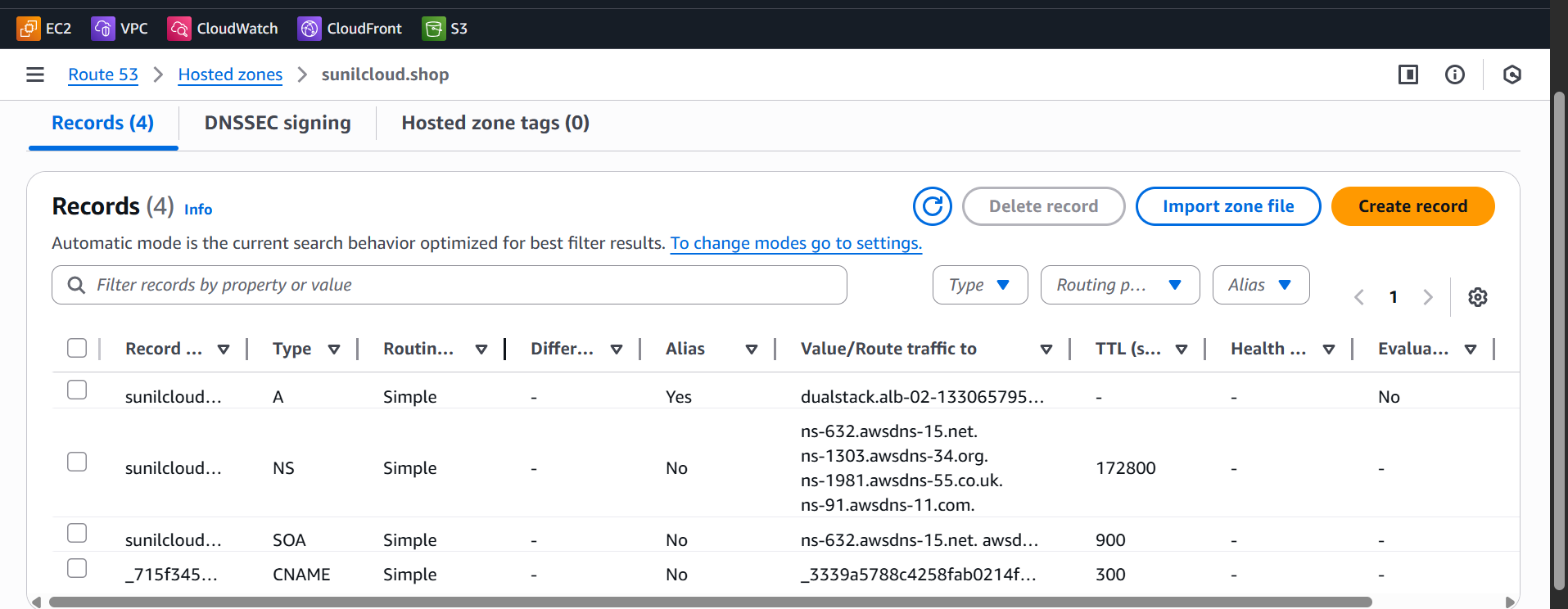
Choose:

* **Record name**: Leave blank for root domain (example.com) or type subdomain (www).
* **Record type**: **A — IPv4 address** (&Cname).
* **Alias**: **Yes**.
* **Route traffic to**: **Alias to Application Load Balancer**.
* **Region**: Select the same region as your ALB.
* **Choose load balancer**: Select your ALB from the dropdown.

**Save record**.

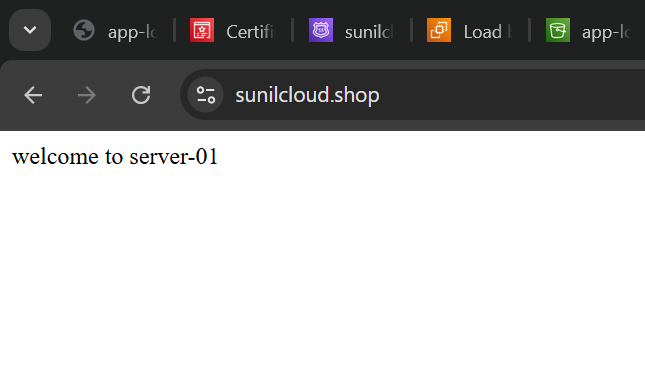
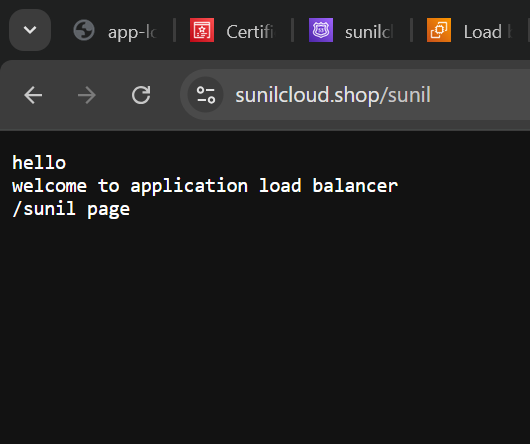
****

****

****

Test

With yourDomainname In the webite

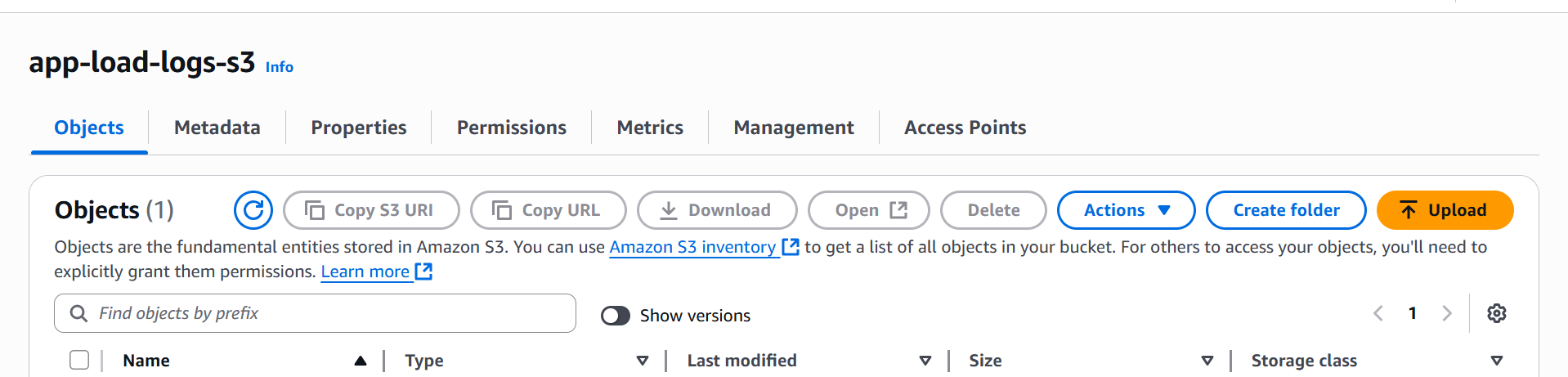
 

**6) Push the application load balancer logs to s3**

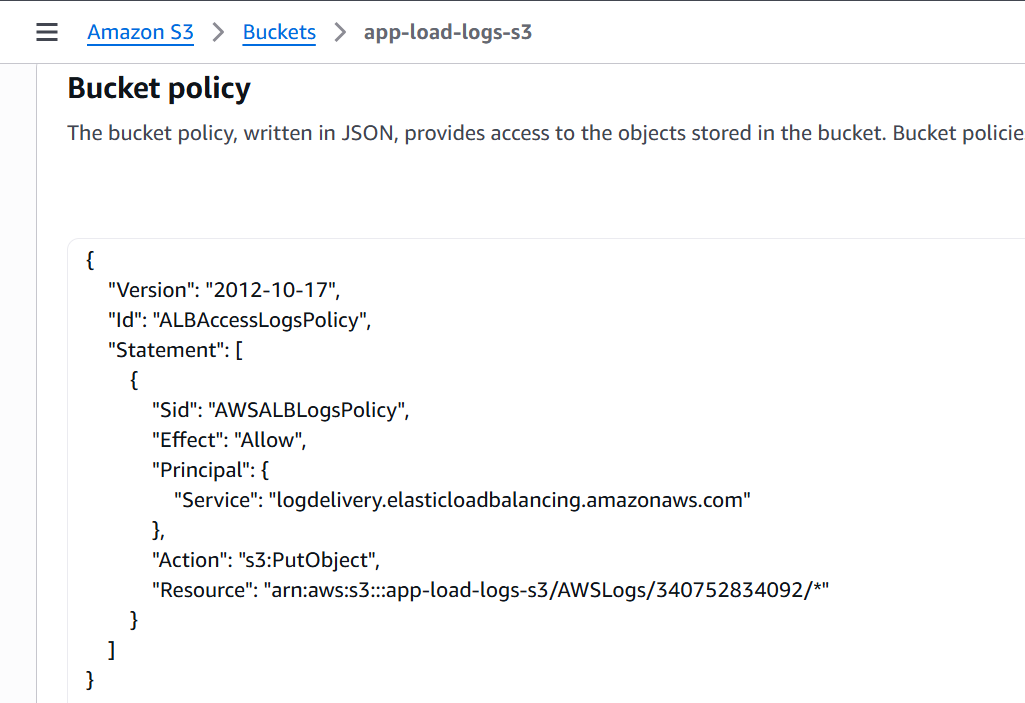
Step 1 — Create or Use an S3 Bucket

Go to **AWS S3 Console**.

Create a bucket (example: app-load-logs-s3).



Step 2 — Add Bucket Policy for ALB Logging



Step 3 — Enable Access Logging in ALB

Go to AWS EC2 Console → Load Balancers.

Select your ALB.

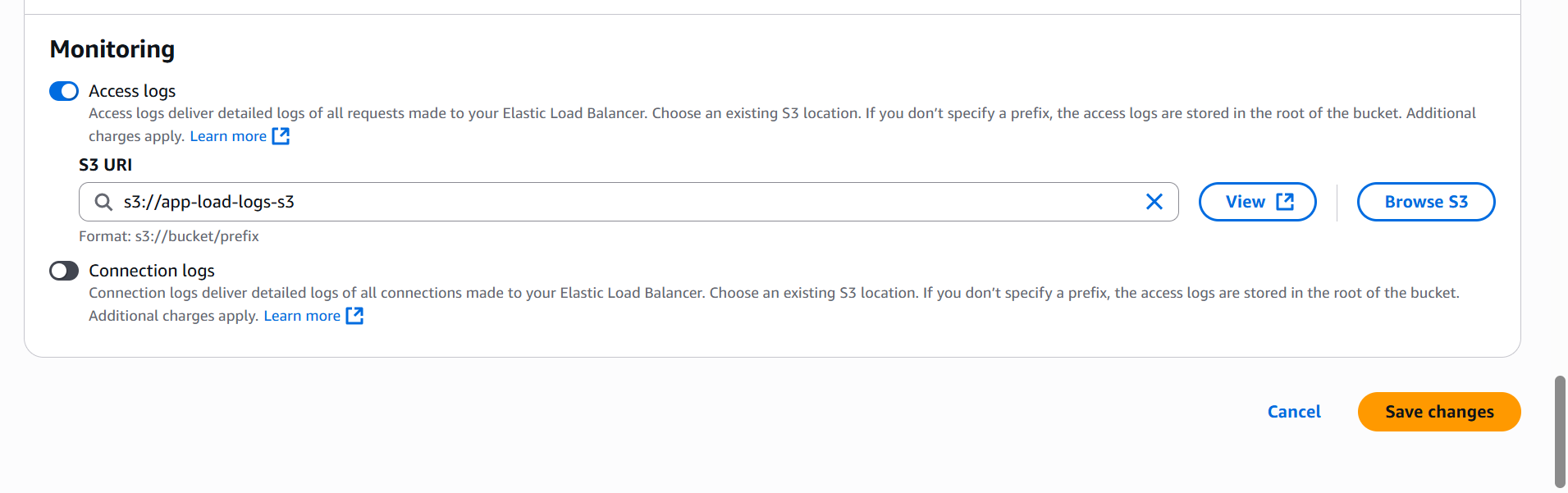
Go to Attributes tab.

Find Access logs → Click Edit.

Turn Access logs ON.

S3 location: s3://<bucket-name>

Save changes.



## **Step 4 — Wait & Verify**

* Logs are delivered every **5 minutes or so**.

