**LOADBALANCER TASKS**

1. Configure Classic Load balancer.

**🔹 Step 1: Go to EC2 → Load Balancers**

1. Sign in to the **AWS Management Console**
2. Navigate to **EC2**
3. On the left, choose **Load Balancers**
4. Click **Create Load Balancer**
5. Choose **Classic Load Balancer**

**🔹 Step 2: Configure Load Balancer Basics**

* **Name**: my-classic-lb
* **Create LB inside**: Select your **VPC**
* **Listeners**:
  + Default: HTTP (80) → HTTP (80)
  + You can add HTTPS (443) if needed (requires SSL cert)
* Click **Next: Assign Security Groups**

**🔹 Step 3: Assign Security Group**

* Choose an **existing security group** or create a **new one**
* Make sure to allow:
  + Port 80 (HTTP) or 443 (HTTPS) from 0.0.0.0/0 (or restrict as needed)
  + Any ports your backend instances need

Click **Next: Configure Health Check**

**🔹 Step 4: Configure Health Check**

* **Ping Protocol**: HTTP
* **Ping Port**: 80
* **Ping Path**: / *(or /health if your app supports it)*
* Advanced Settings (optional but recommended):
  + **Response Timeout**: 5
  + **Interval**: 30
  + **Unhealthy threshold**: 2
  + **Healthy threshold**: 5

Click **Next: Add EC2 Instances**

**🔹 Step 5: Register EC2 Instances**

* Select the EC2 instances you want to load balance
* They should be:
  + In the **same VPC**
  + In a **running** state
  + Have required ports **open in their security groups**

Click **Next: Add Tags**

**🔹 Step 6: Add Tags (Optional)**

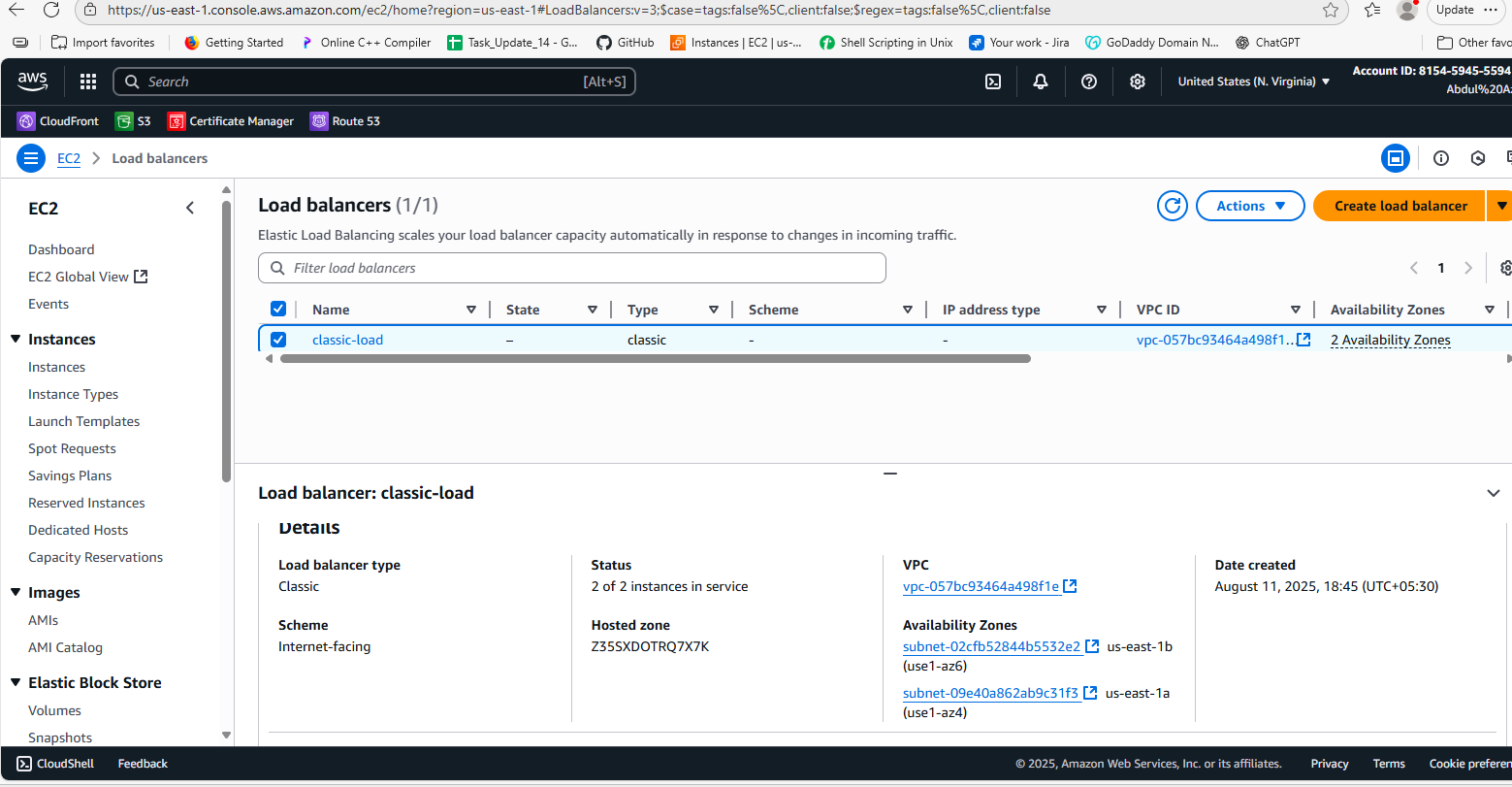
* Add key-value pairs for tagging (e.g., Name: WebLoadBalancer)

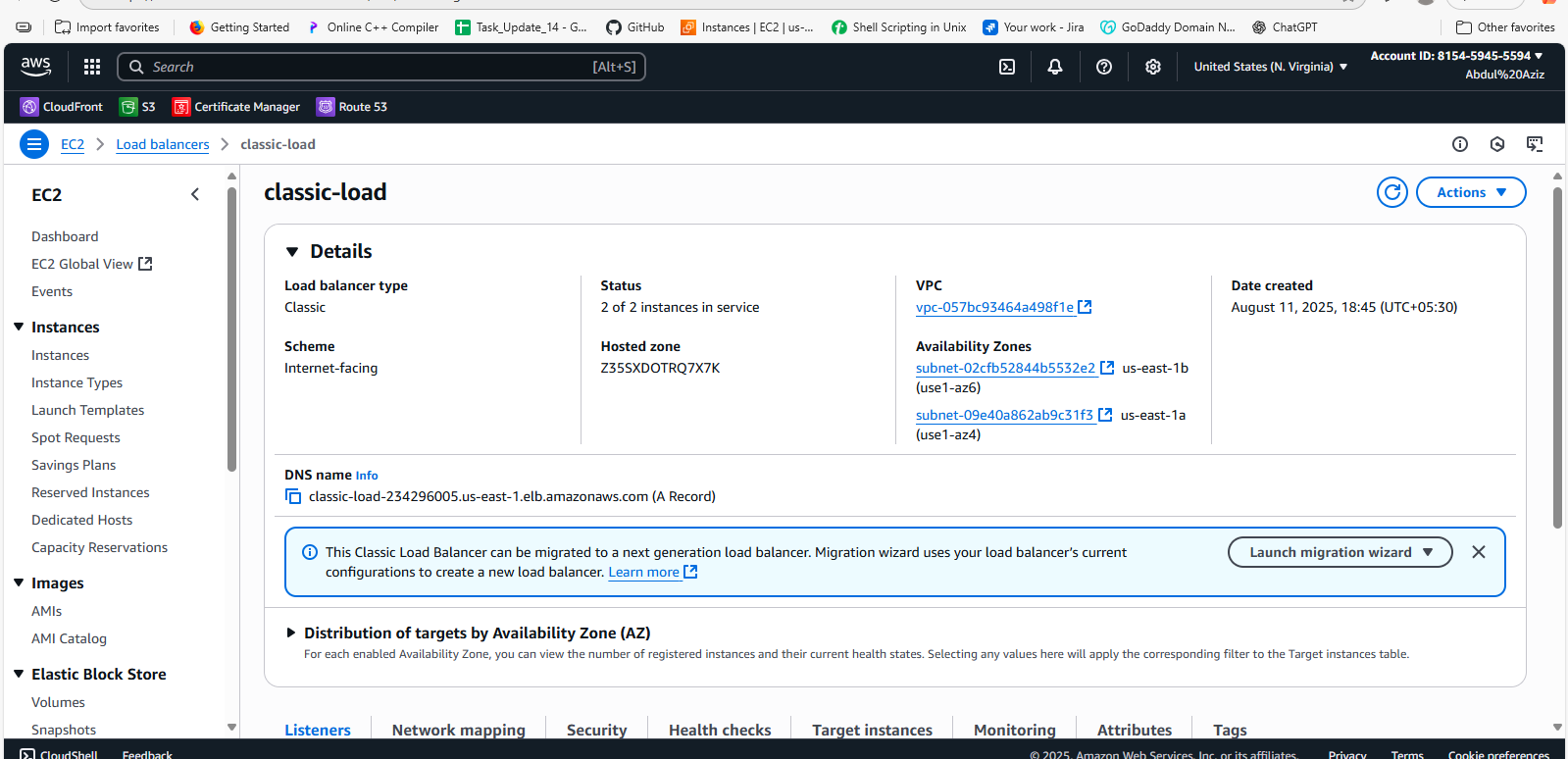
Click **Next: Review and Create**

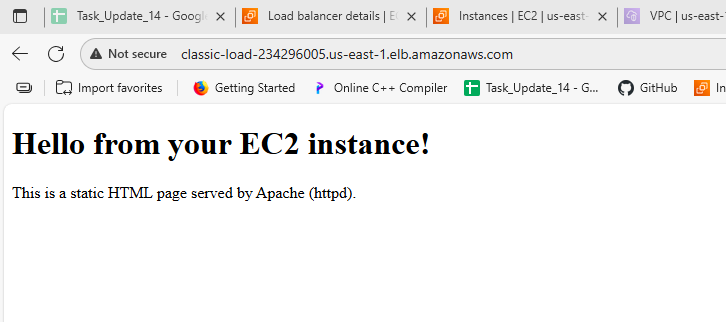
**🔹 Step 7: Review and Launch**

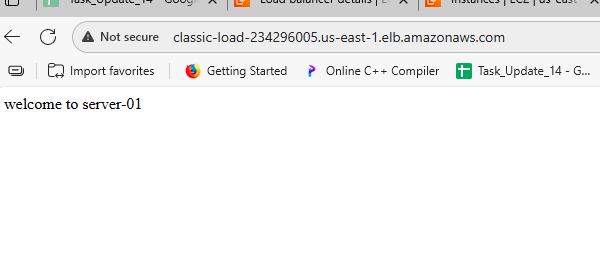
* Review all settings
* Click **Create**

✅ Done! Your Classic Load Balancer will start provisioning.









1. Configure Application Load balancer.

**🔸 Step 1: Go to Load Balancers**

1. Go to the **AWS Management Console**
2. Navigate to **EC2** → **Load Balancers** (left menu)
3. Click **Create Load Balancer**
4. Choose **Application Load Balancer**

**🔸 Step 2: Basic Configuration**

* **Name**: my-alb
* **Scheme**: Choose one:
  + Internet-facing (for public apps)
  + Internal (for private/internal use)
* **IP address type**: IPv4 or Dualstack (if you want IPv6)

Click **Next: Network Mapping**

**🔸 Step 3: Network Mapping**

* **VPC**: Select your VPC
* **Availability Zones**:
  + Select **at least 2** AZs for high availability
  + For each AZ, select a **public subnet**

Click **Next: Configure Security Groups**

**🔸 Step 4: Configure Security Groups**

* Select an existing security group **OR** create a new one
* Ensure it allows:
  + **HTTP (port 80)** and/or **HTTPS (port 443)**
  + Optional: Restrict by IP for security

Click **Next: Configure Routing**

**🔸 Step 5: Configure Routing (Target Group)**

This is where you define where the ALB sends traffic.

* **Target group name**: my-target-group
* **Target type**: Instance (for EC2) or IP (for IP-based targets)
* **Protocol**: HTTP or HTTPS
* **Port**: 80 (or the port your app listens on)
* **Health checks**:
  + **Path**: / or /health (your app must respond to this)
  + **Protocol**: Usually HTTP

Click **Next: Register Targets**

**🔸 Step 6: Register Targets**

* Select the **EC2 instances** you want to send traffic to
* Choose the correct **port** (e.g., 80)
* Click **Add to registered**

Click **Next: Review**

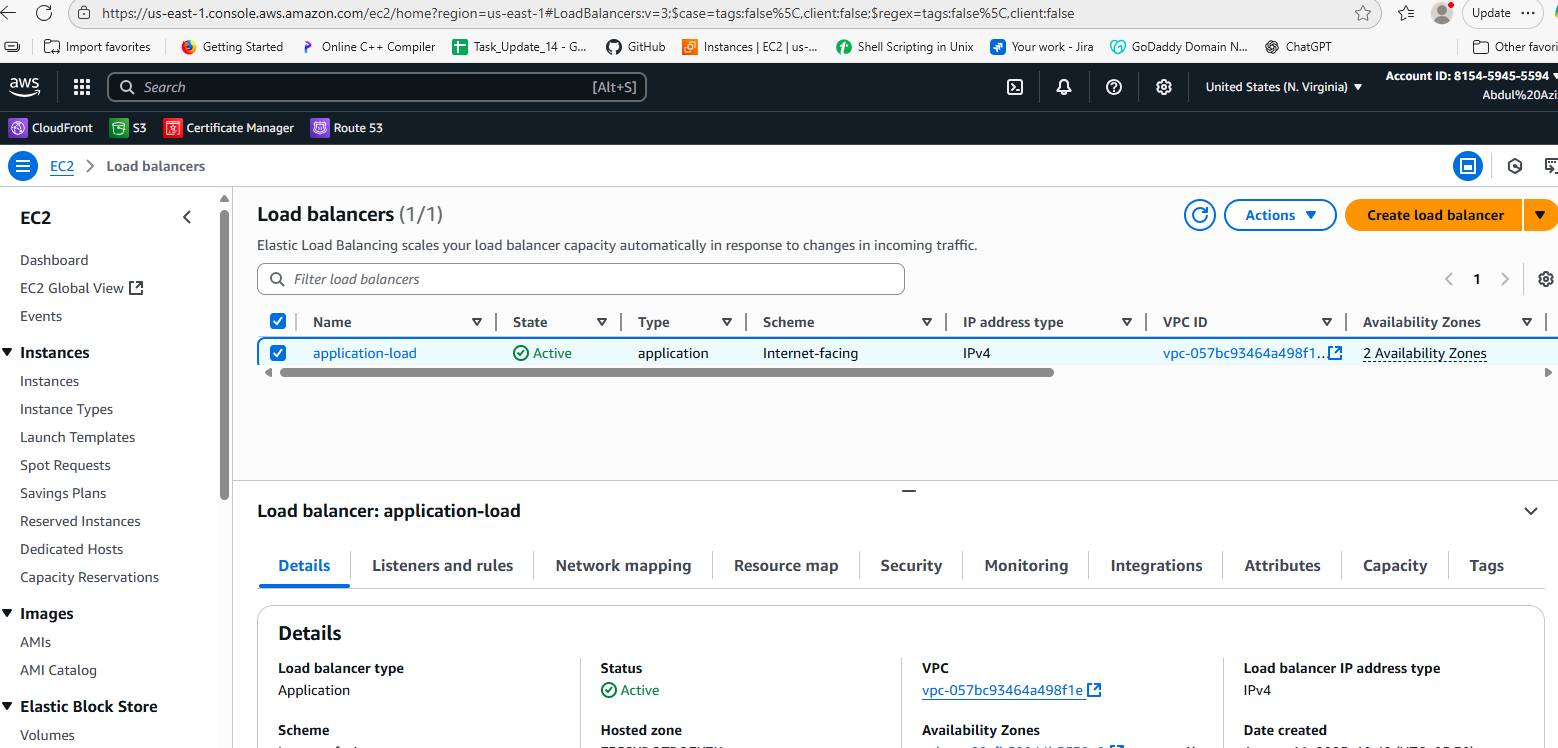
**🔸 Step 7: Review and Create**

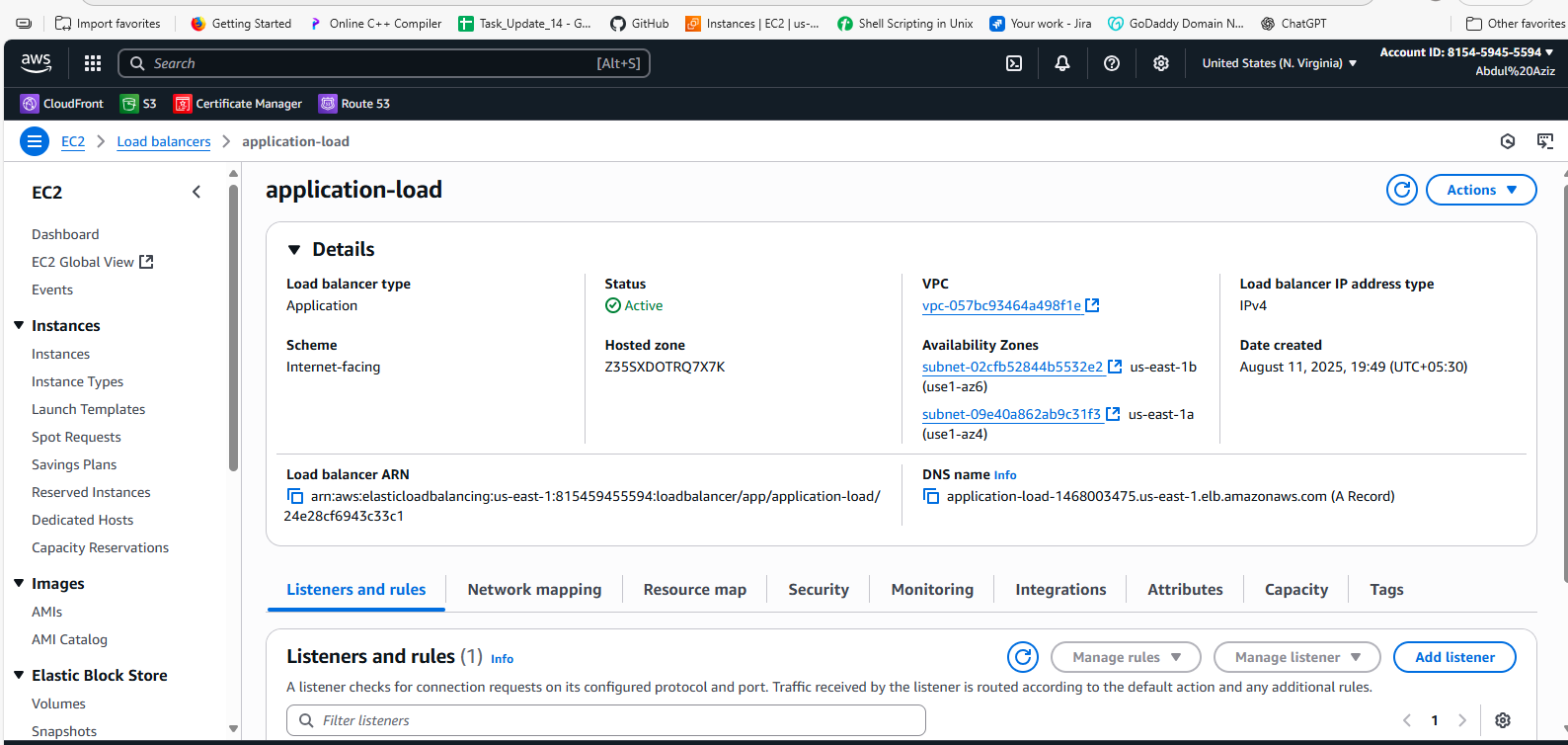
* Review all settings:
  + ALB name, listeners, security group, target group, etc.
* Click **Create Load Balancer**

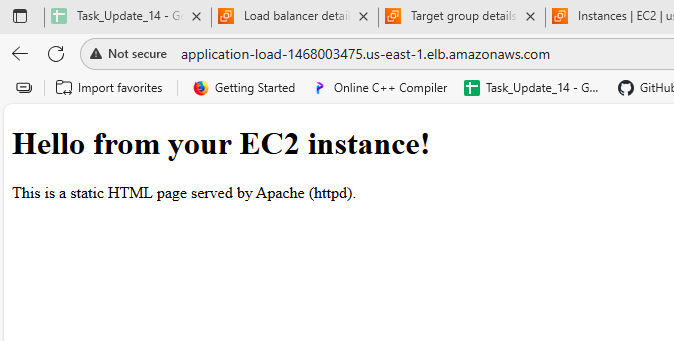
✅ You’ll see a success message once it's created.

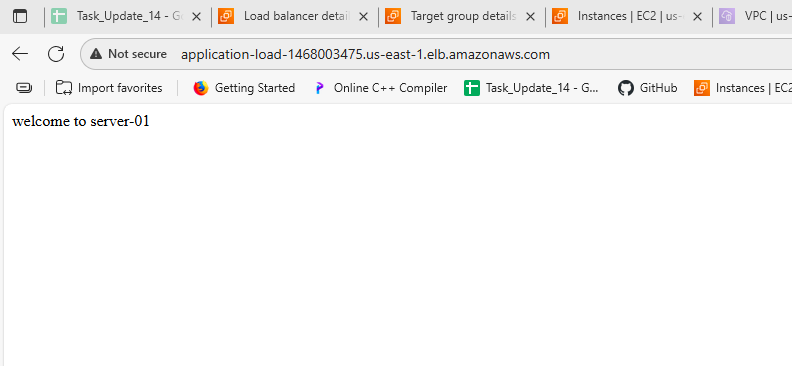
**🔸 Step 8: Test It**

* Go to your ALB in EC2 → Load Balancers
* Copy the **DNS name** (e.g., my-alb-123456.us-east-1.elb.amazonaws.com)
* Open in your browser:









1. Configure Network Load balancer.

### 🔸 Step 1: Go to Load Balancers

1. Open the **AWS Console**
2. Go to **EC2 → Load Balancers**
3. Click **Create Load Balancer**
4. Choose **Network Load Balancer**
5. Click **Create**

### 🔸 Step 2: Basic Configuration

* **Name**: my-nlb
* **Scheme**:
  + Internet-facing (for public access)
  + Internal (for internal use only)
* **IP address type**: Choose IPv4 or Dualstack (IPv4 + IPv6)

### 🔸 Step 3: Network Mapping (Subnets)

* Select your **VPC**
* Choose **at least 2 Availability Zones**
* For each AZ, select a **public subnet** (if Internet-facing)
* Optionally assign **Elastic IPs** for fixed IP addresses

### 🔸 Step 4: Configure Listeners and Routing

#### 🔹 Listener Settings:

* **Protocol**: Choose TCP, UDP, or TLS
* **Port**: Common choices are 80, 443, 22, 3306, etc.

#### 🔹 Target Group Settings:

* **Target group name**: my-nlb-targets
* **Target type**: Choose one:
  + Instance – EC2
  + IP – IP addresses
  + Lambda – (less common with NLB)
* **Protocol**: TCP or TLS
* **Port**: The port on which targets are listening (e.g., 80)
* **Health checks**:
  + Protocol: Usually TCP (simpler than HTTP)
  + Port: traffic port or specific port

Click **Next**

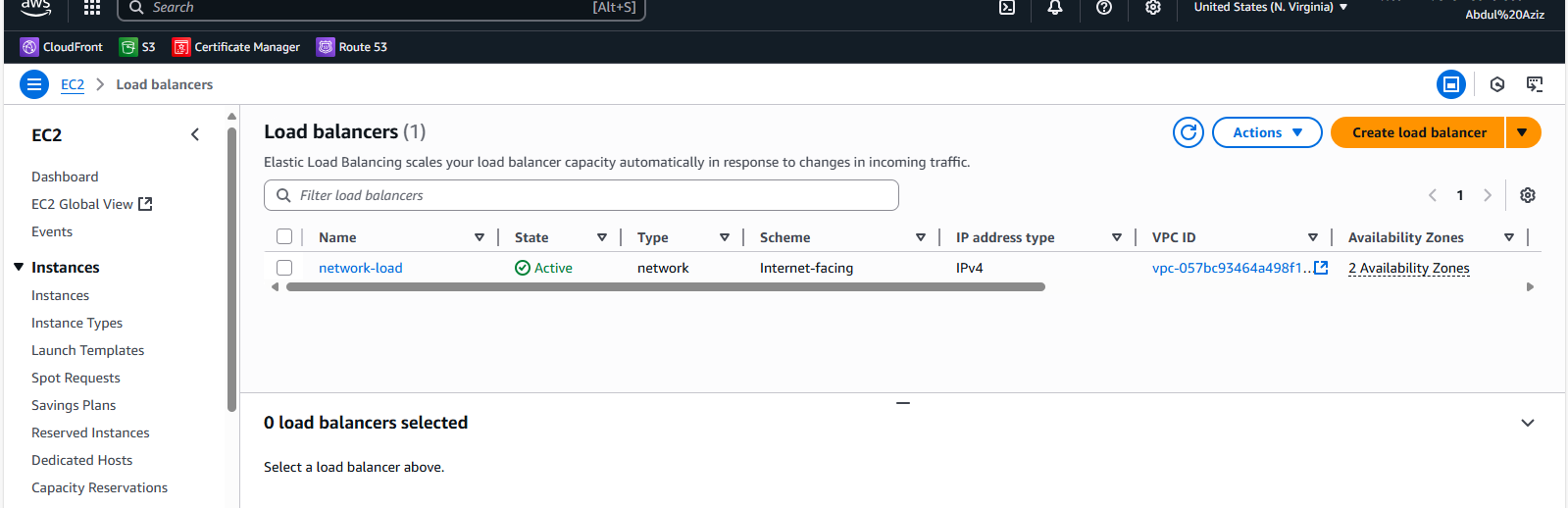
### 🔸 Step 5: Register Targets

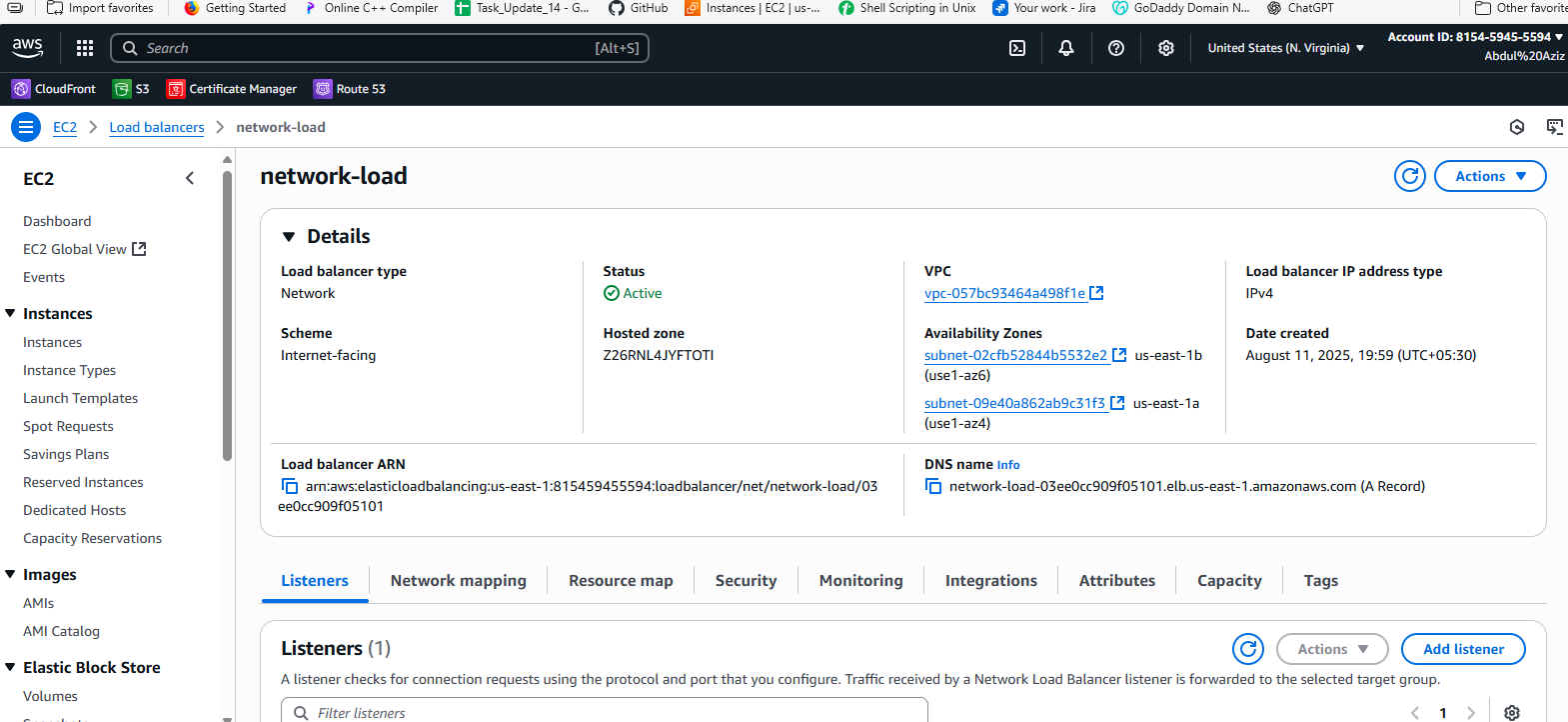
* Select EC2 instances
* Specify the port (e.g., 80)
* Click **Add to registered**

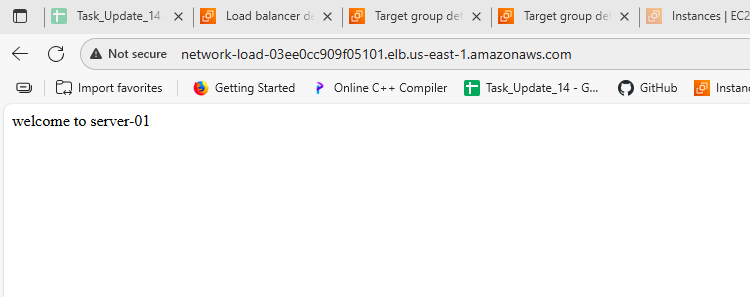
Click **Next → Review → Create**

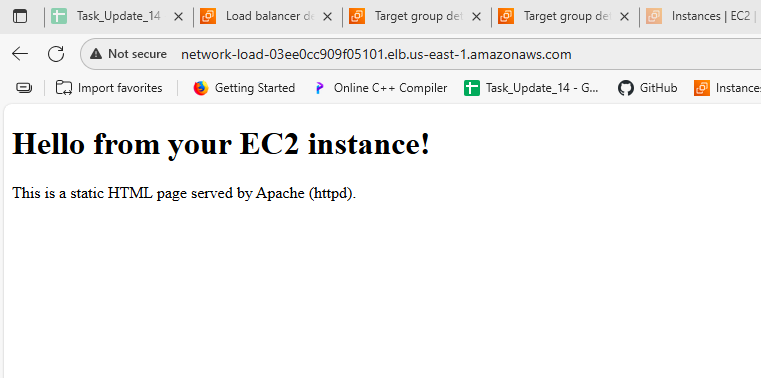
### ✅ Step 6: Test the NLB

* Go to EC2 → Load Balancers
* Find your **Network Load Balancer**
* Copy the **DNS name** (e.g., my-nlb-12345678.us-east-1.elb.amazonaws.com)
* Try connecting to it using a browser, curl, or TCP client:









1. Attach SSL for application load balancer.

**🔸 Step 1: Get an SSL Certificate from ACM**

1. Go to **AWS Console → Certificate Manager (ACM)**
2. Click **Request a certificate**
3. Choose **Public certificate**
4. Enter your **domain name** (e.g., example.com, www.example.com)
5. Choose a validation method:
   * **DNS validation** (easy if using Route 53)
   * **Email validation**
6. Follow the instructions to validate
7. Once validated, the certificate status will be **"Issued"**

**🔸 Step 2: Go to Your Application Load Balancer**

1. Go to **EC2 → Load Balancers**
2. Select your **Application Load Balancer**
3. Click on the **Listeners** tab
4. Click **Add listener** or edit existing one:
   * Protocol: HTTPS
   * Port: 443

**🔸 Step 3: Attach the SSL Certificate**

* Under **Secure listener settings**:
  + Select **"Choose an existing certificate from ACM"**
  + Pick your validated certificate from the dropdown
* Select a **security policy** (leave default or use latest: ELBSecurityPolicy-2021-06)
* Click **Save** or **Add**

**🔸 Step 4: Add Listener Rule (Optional)**

* Go to **Listeners → HTTPS (443) → View/edit rules**
* Add a rule to **forward** requests to your **target group**

Example:

* IF: Path is /
* THEN: Forward to → my-target-group

**🔸 Step 5: (Optional) Redirect HTTP → HTTPS**

1. Click on the **HTTP (80)** listener
2. Click **View/edit rules**
3. Add a rule:
   * IF: Path is /
   * THEN: **Redirect to**
     + Protocol: HTTPS
     + Port: 443
     + Status code: HTTP\_301

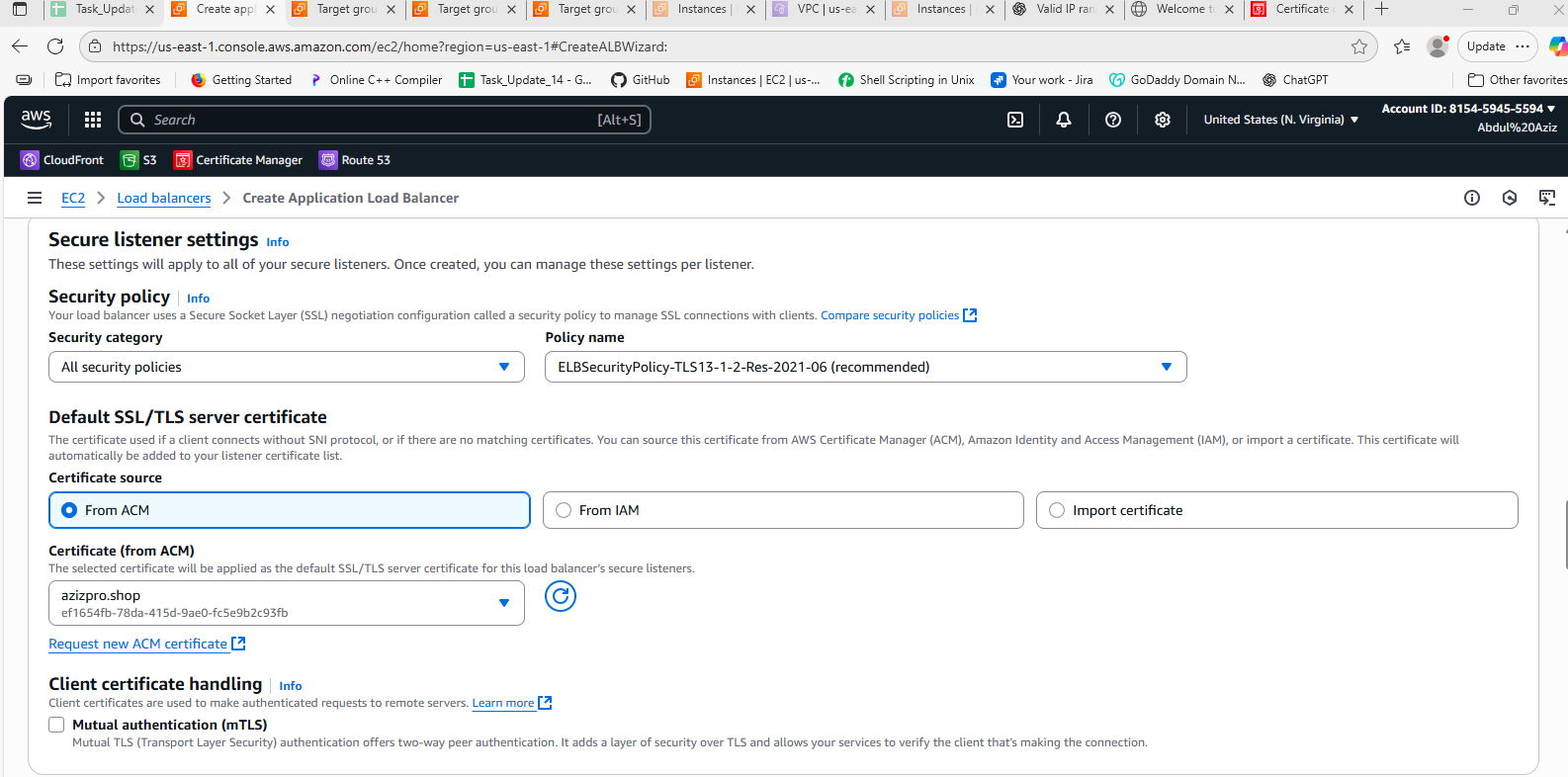
**✅ Step 6: Test Your ALB**

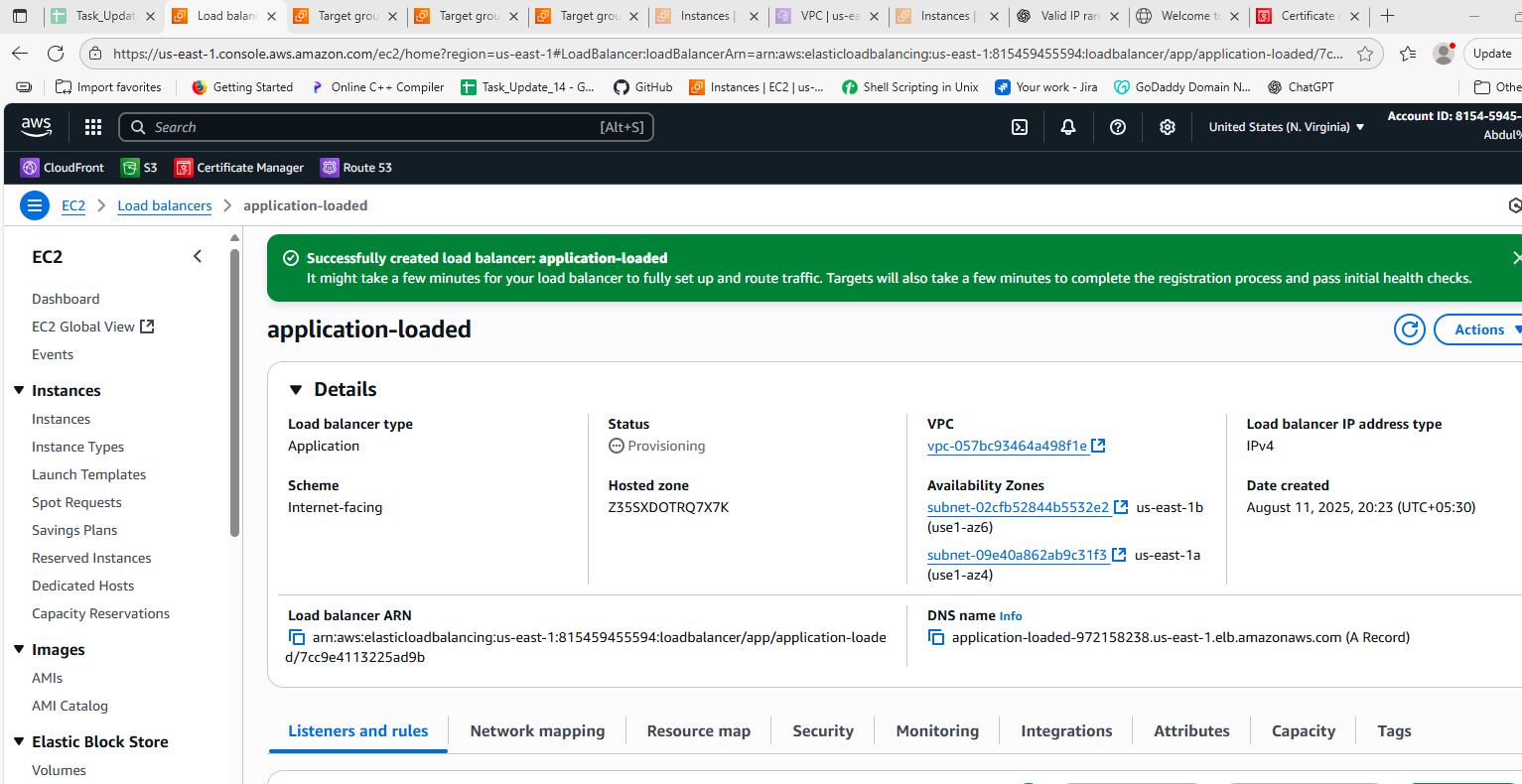
Open your browser and go to:

pgsql

CopyEdit

https://your-domain-name.com





1. Map Application load balancer to R53.

**🔸 Step 1: Get Your ALB DNS Name**

1. Go to **AWS Console → EC2 → Load Balancers**
2. Select your **Application Load Balancer**
3. Copy the **DNS name** (something like:  
   my-alb-1234567890.us-east-1.elb.amazonaws.com)

**🔸 Step 2: Go to Route 53 Hosted Zones**

1. Open **AWS Console → Route 53 → Hosted zones**
2. Click on your **domain name** (e.g., example.com)

**🔸 Step 3: Create a Record Set (Alias)**

1. Click **Create record**
2. In the form:
   * **Record name**:
     + Enter subdomain (e.g., www)
     + Leave blank if you want root domain (example.com)
   * **Record type**: Select A – IPv4 address
   * **Alias**: Toggle **On**
   * **Alias target**:
     + Click the dropdown
     + Select your **Application Load Balancer** from the list
3. Leave **Routing policy** as Simple (default)
4. Click **Create records**

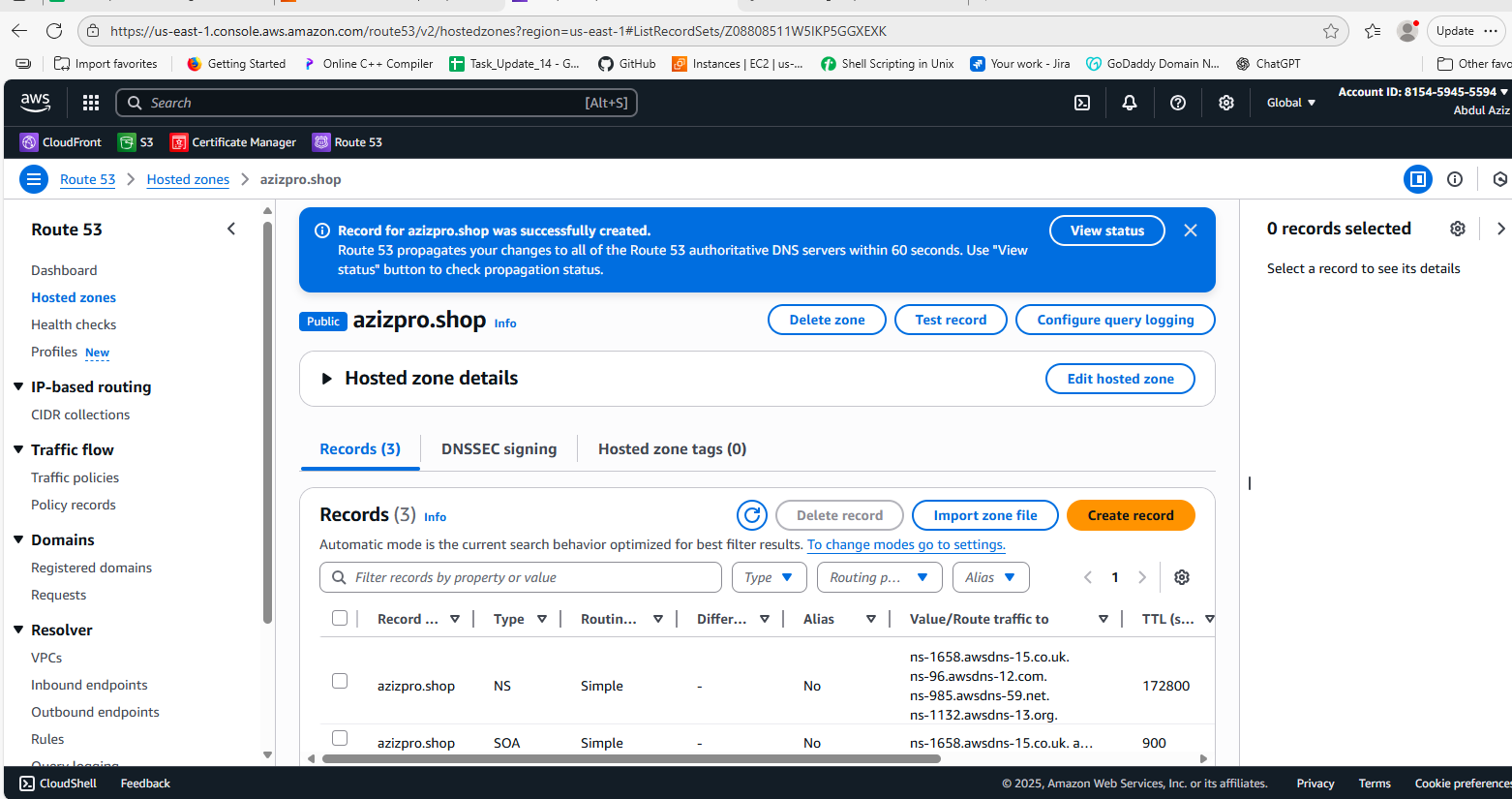
**🔸 Step 4: Test Your Domain**

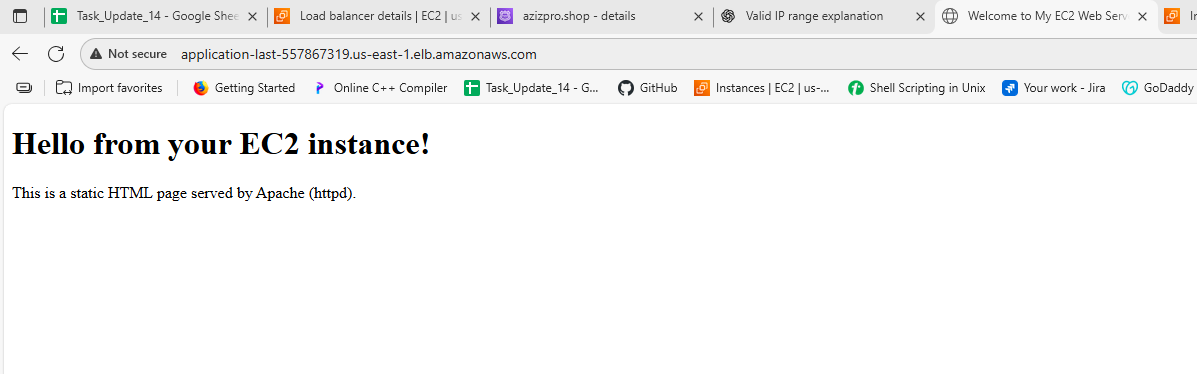
* Open browser and go to:

arduino

CopyEdit

http://www.example.com





* 1. Push the application load balancer logs to s3

**🔸 Step 1: Create an S3 Bucket (if not already)**

1. Go to **S3 Console**
2. Click **Create bucket**
3. Enter a **unique bucket name** (e.g., my-alb-logs-bucket)
4. Choose the same **Region** as your ALB
5. Leave other settings default or customize as needed
6. Click **Create bucket**

**🔸 Step 2: Add Bucket Policy (Permissions for ALB)**

1. Go to **S3 → Your Bucket → Permissions → Bucket Policy**
2. Add this policy (replace your-bucket-name and region):

json

CopyEdit

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AWSALBLoggingPermissions",

"Effect": "Allow",

"Principal": {

"Service": "logdelivery.elasticloadbalancing.amazonaws.com"

},

"Action": "s3:PutObject",

"Resource": "arn:aws:s3:::your-bucket-name/AWSLogs/\*"

}

]

}

**🔸 Step 3: Enable Access Logs on ALB**

1. Go to **EC2 Console → Load Balancers**
2. Select your **Application Load Balancer**
3. Go to the **Attributes** tab
4. Click **Edit attributes**
5. Under **Access logs**:
   * ✅ Enable access logs
   * **S3 bucket name**: enter your bucket name (e.g., my-alb-logs-bucket)
   * **Prefix (optional)**: e.g., alb-logs/
6. Click **Save**

**🔸 Step 4: Wait for Logs to Appear**

* Logs are written in folders like:

php-template

CopyEdit

AWSLogs/<account-id>/elasticloadbalancing/<region>/<date>/

* It may take a few minutes to hours before logs appear

✅ Done! Your ALB logs will now be pushed to the specified S3 bucket.

