# LOAD-BALANCER TASKS

| **Category** | **Details** |
| --- | --- |
| **Title** | Classic Load Balancer, Application Load Balancer (HTTP & HTTPS), Network Load Balancer, ACM SSL Certificate, Route 53 Domain Mapping, ALB Logging to S3 |
| **Objective** | To deploy multiple EC2 instances across different Availability Zones and configure Classic Load Balancer, Application Load Balancer, Network Load Balancer, HTTPS using ACM Certificate, map custom domain via Route 53, and enable ALB access logs, connection logs, and health logs to S3 for monitoring and security. |
| **Prerequisites** | • AWS Account |
| • VPC with 2 public subnets |  |
| • EC2 key pair |  |
| • Default security group with HTTP (80) & HTTPS (443) allowed |  |
| • Apache/httpd installed via user data |  |
| • Route 53 hosted zone for mujju.store |  |
| • ACM certificate requested and validated via CNAME |  |
| • S3 bucket for ALB logs with correct bucket policy |  |
| **Errors Faced** | • EC2 health check failed for Classic/ALB/NLB |
| • HTTPS listener could not be created due to missing certificate |  |
| • ACM certificate stuck in “Pending Validation” |  |
| • Domain mujju.store not resolving after ALB creation |  |
| • S3 bucket showing Access Denied for log delivery |  |
| • Load balancer logs not appearing immediately |  |
| **How I Fixed the Errors** | •Corrected health check path to **/index.html** and restarted Apache |
| • Added ACM validation CNAME in Route 53 → certificate **Issued** |  |
| • Selected ACM certificate while creating HTTPS listener |  |
| • Created A-Record (Alias) in Route 53 pointing mujju.store → ALB DNS |  |
| • Added required S3 bucket policy for logdelivery.elasticloadbalancing.amazonaws.com |  |
| • Enabled ALB logs again and verified log files in S3 after a few minutes |  |
| **Conclusion** | All EC2 instances, load balancers (Classic, ALB, NLB), HTTPS configuration with ACM, domain routing, and S3 logging were successfully implemented. Traffic is now load-balanced across instances, domain resolves securely via HTTPS, and ALB logs are stored in S3 for monitoring. The environment is fully functional, scalable, secure, and production-ready. |

1. Configure Classic Load balancer.

**1. Launched EC2 Instance – Classic-EC2-1**

1. Opened **Launch Instance** wizard.
2. Selected an **Amazon Linux 2023 AMI**.
3. Chose **t2.micro** instance type.
4. Selected an existing **Key Pair: n.virginia**.
5. Configured **Network Settings**:
   * VPC: vpc-0f2fed5265d575471
   * Subnet: PUBLIC-SUBNET-1 (us-east-1a)
   * Auto-assign Public IP: **Enabled**
   * Security Group: **default**
6. Added **User Data script**:

#!/bin/bash

sudo yum update -y

sudo yum install -y httpd

systemctl start httpd

systemctl enable httpd

echo "HELLO THIS IS MY CLASSIC LOAD BALANCER EC2-1" > /var/www/html/index.html

Launched the EC2 instance.

Verified by hitting its public IP → Displayed message:  
**“HELLO THIS IS MY CLASSIC LOAD BALANCER EC2-1”**

**2. Launched Second EC2 Instance – Classic-EC2-2**

1. Repeated launch process for a second EC2.
2. Chose:
   * Subnet: PUBLIC-SUBNET-02 (us-east-1c)
   * Same VPC.
   * Same Security Group (default).
3. Updated user data to show a different message:
4. echo "HELLO THIS IS MY CLASSIC LOAD BALANCER EC2-2" > /var/www/html/index.html
5. Launched the instance.
6. Verified its public IP → Message displayed:  
   **“HELLO THIS IS MY CLASSIC LOAD BALANCER EC2-2”**

**3. Created a Classic Load Balancer**

1. Opened **Create Classic Load Balancer** wizard.
2. Entered Load Balancer Name: **my-classic-lb**
3. Scheme: **Internet Facing**
4. Selected same VPC:
   * vpc-0f2fed5265d575471
5. Selected two subnets:
   * us-east-1a (PUBLIC-SUBNET)
   * us-east-1c (PUBLIC-SUBNET-02)
6. Assigned **default security group**.
7. Configured Listener:
   * Load Balancer Protocol: **HTTP : 80**
   * Instance Protocol: **HTTP : 80**
8. Enabled:
   * **Cross-zone load balancing**
   * **Connection draining (300 seconds)**

**4. Registered EC2 Instances with Classic Load Balancer**

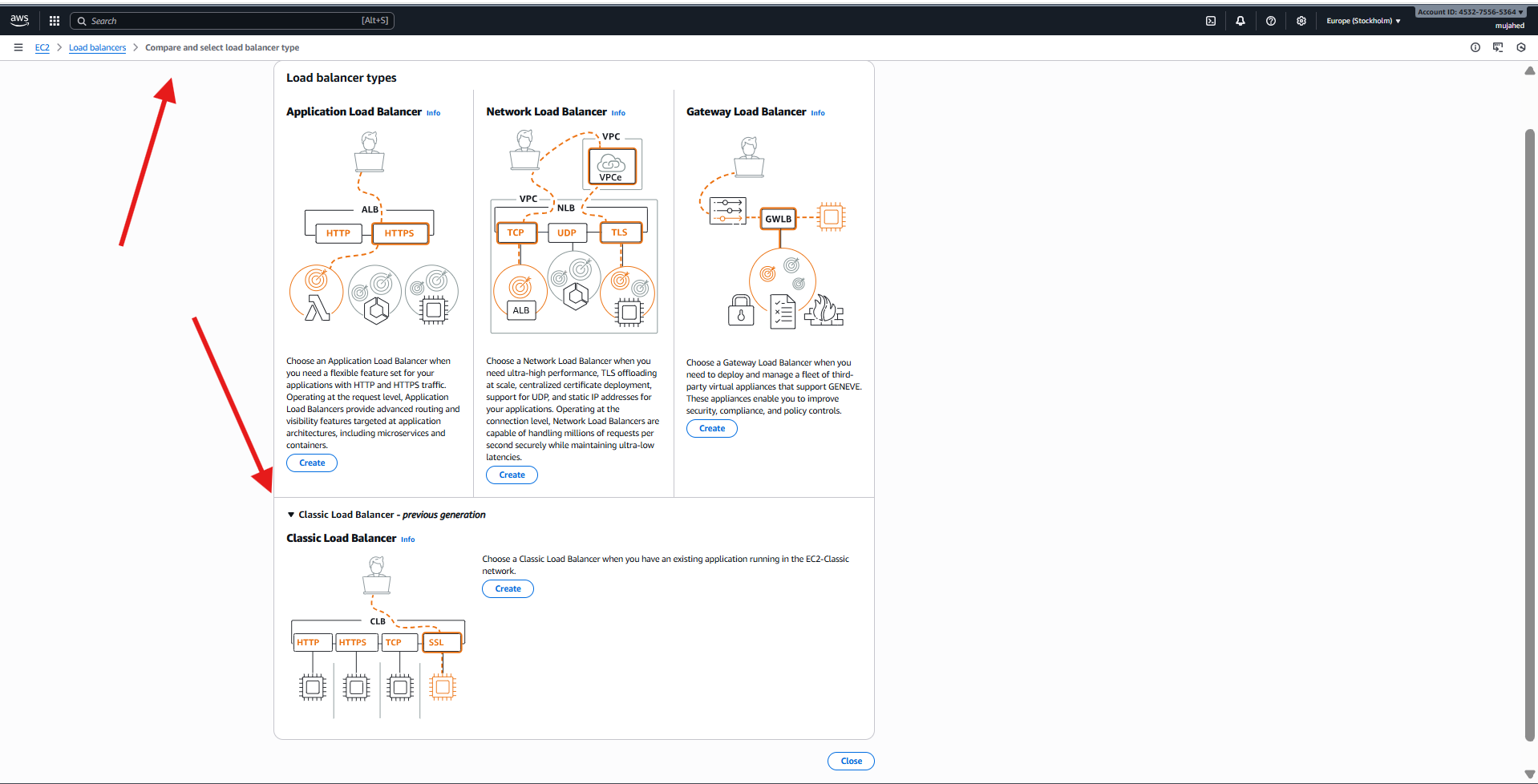
1. In the “Add Instances” section:
   * Selected both:
     + i-0cS9000a5ua0a69b (CLASSIC-EC2-1)
     + i-0aB64b07B854091c2 (CLASSIC-EC2-2)
2. Clicked **Confirm**.

**5. Verified Health Checks**

1. Went to **Target Instances** tab.
2. Observed both EC2 instances showing:
   * **Status: In-service**
   * Meaning both passed the health checks.

**6. Final Testing**

1. Copied the **DNS name** of the Classic Load Balancer:
2. my-classic-lb-1604166499.us-east-1.elb.amazonaws.com
3. Loaded it multiple times.
4. Confirmed **round-robin traffic distribution**:
   * Sometimes showing: *HELLO THIS IS MY CLASSIC LOAD BALANCER EC2-1*
   * Sometimes showing: *HELLO THIS IS MY CLASSIC LOAD BALANCER EC2-2*



A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Configure Application Load balancer.

**1. Created EC2 Instances for ALB**

launched **two EC2 instances**

**EC2 Instance 1 – my-alb-ec2-1**

* Installed Apache (httpd) via user data.
* Added HTML message:
* HELLO THIS IS MY APP-LOAD-BALANCER-EC2-1

**EC2 Instance 2 – my-alb-ec2-2**

* Installed Apache (httpd) via user data.
* Added HTML message:
* HELLO THIS IS MY APP-LOAD-BALANCER-EC2-2

Both instances were running inside:

* **VPC:** vpc-0f2fed5265d575471
* **Subnets:**
  + us-east-1a
  + us-east-1c
* **Security Group:** default
* Port 80 open for HTTP traffic.

**2. Created a Target Group**

created a new target group for ALB.

**Target Group Settings**

| **Setting** | **Value** |
| --- | --- |
| **Target type** | Instances |
| **Target group name** | my-alb-target-grp |
| **Protocol** | HTTP |
| **Port** | 80 |
| **IP address type** | IPv4 |
| **VPC** | vpc-0f2fed5265d575471 |

**Health Check Settings**

| **Setting** | **Value** |
| --- | --- |
| **Protocol** | HTTP |
| **Health check path** | /index.html |

**3. Registered EC2 Instances to the Target Group**

added the two EC2 instances:

| **Instance ID** | **Name** | **Port** | **Status** |
| --- | --- | --- | --- |
| i-027859… | my-alb-ec2-2 | 80 | Registered |
| i-0b7dea… | my-alb-ec2-1 | 80 | Registered |

Both were added under **pending** and later became **Healthy**.

**4. Created an Application Load Balancer**

continued to create the ALB:

**ALB Details**

| **Property** | **Value** |
| --- | --- |
| **Name** | my-app-lb |
| **Type** | Application Load Balancer |
| **Scheme** | Internet-facing |
| **Security group** | default |
| **Listeners** | HTTP:80 |
| **VPC** | vpc-0f2fed5265d575471 |
| **Subnets** | us-east-1a, us-east-1c |

**Routing Rule**

Default rule:

IF no other rule → Forward to target group my-alb-target-grp

**5. Verified Load Balancer Resource Map**

viewed the ALB resource map showing:

* **Listener** → HTTP:80
* **Rule** → Forward to target group
* **Target Group** → my-alb-target-grp
* **Targets** → Two healthy EC2 instances

Both targets displayed **Healthy** status.

**6. Tested the Application Load Balancer**

hit both EC2 public IPs manually first:

**EC2-1 Output**

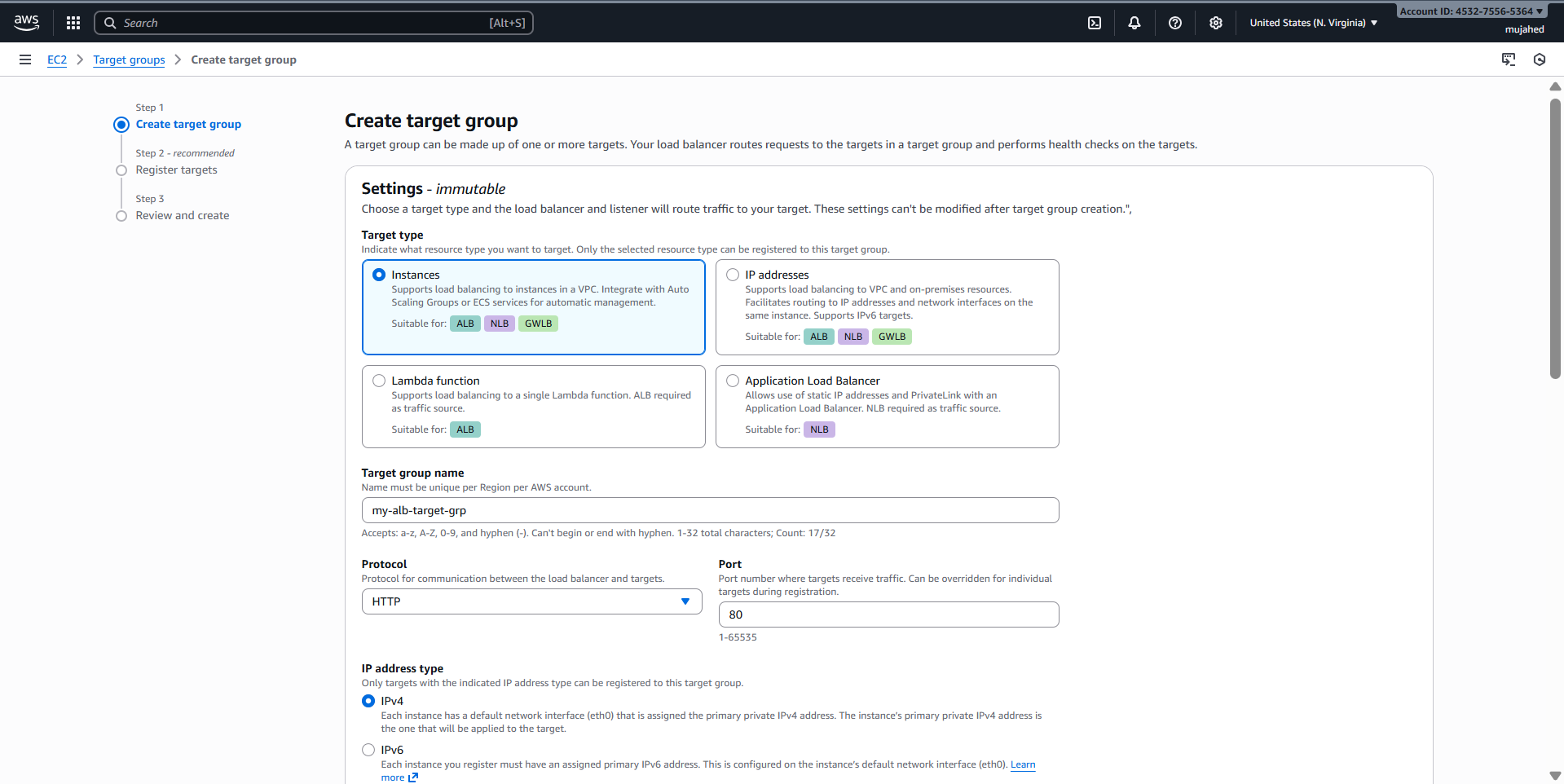
HELLO THIS IS MY APP-LOAD-BALANCER-EC2-1

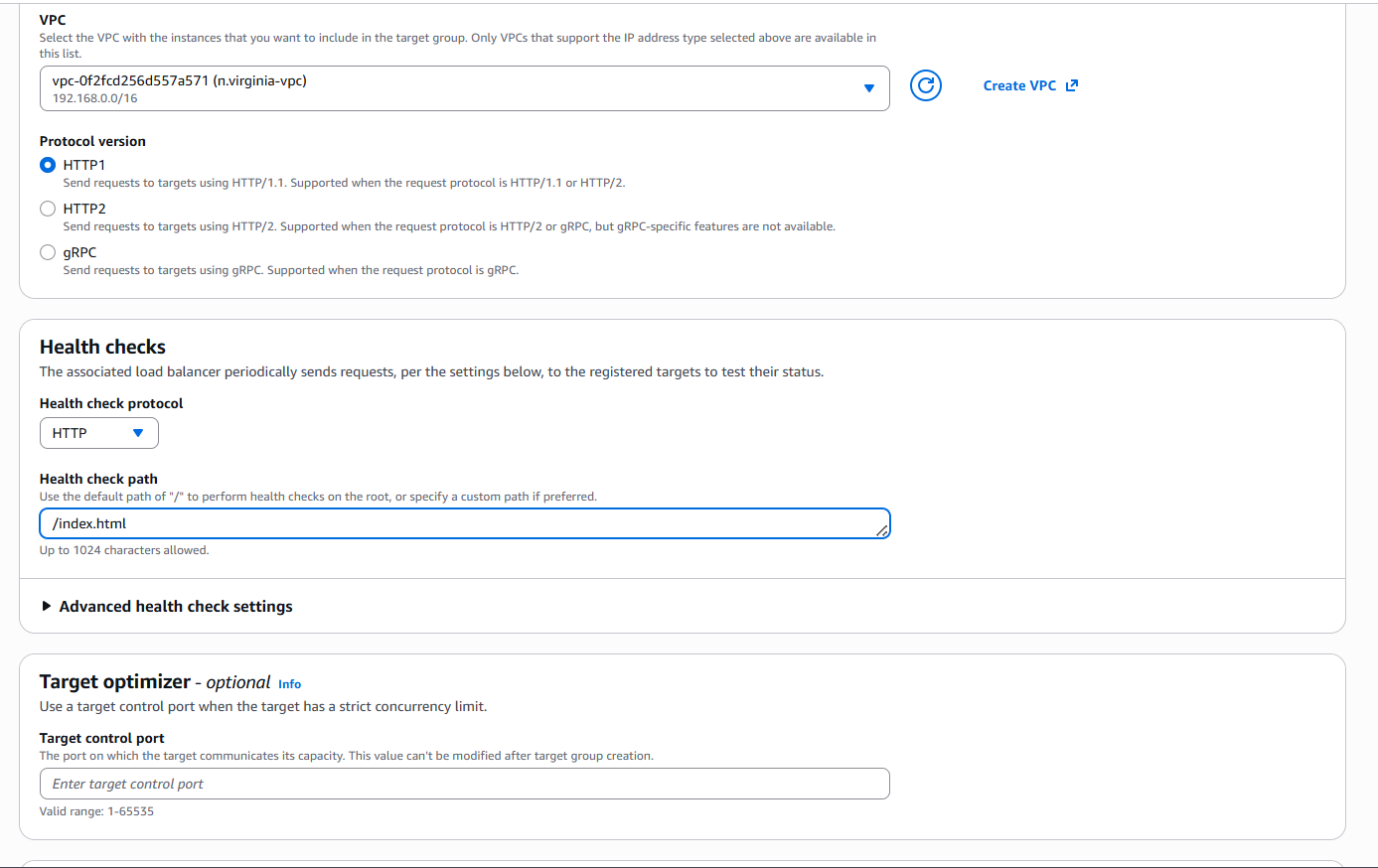
**EC2-2 Output**

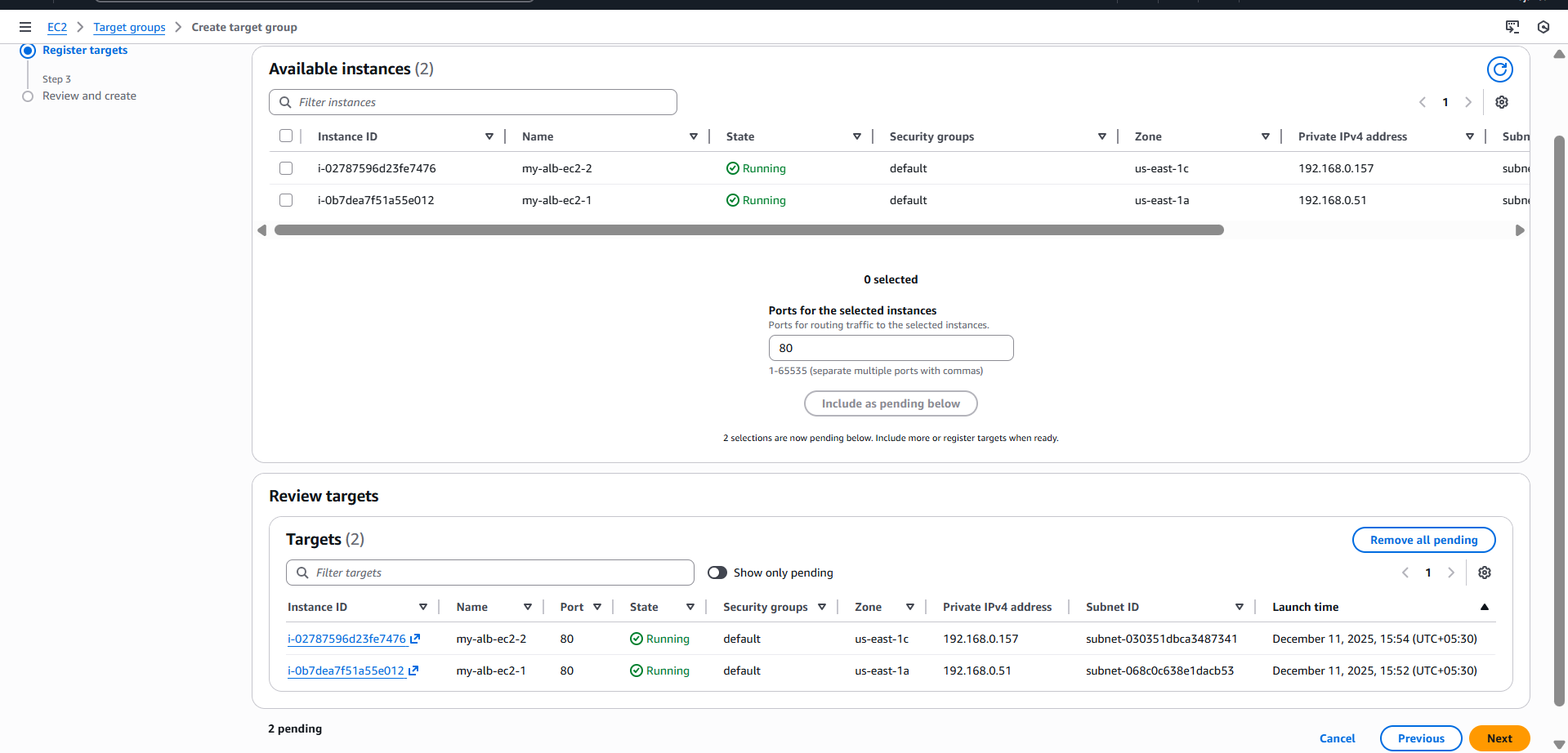
HELLO THIS IS MY APP-LOAD-BALANCER-EC2-2

Then accessed the ALB's DNS/public IP (e.g., 98.92.199.108):

* On refresh, ALB alternated between the two EC2 outputs.
* This confirmed **round-robin load balancing**.







A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a message

AI-generated content may be incorrect.

A white rectangular box with text

AI-generated content may be incorrect.

1. Configure Network Load balancer.

**1. Created Two EC2 Instances for NLB**

launched **two EC2 instances** that will serve traffic behind the Network Load Balancer.

**EC2 Instance 1 (my-nlb-ec2-1)**

* Installed Apache (httpd).
* Added webpage:
* HELLO THIS IS MY NETWORK-LOAD-BALANCER-EC2-1

**EC2 Instance 2 (my-nlb-ec2-2)**

* Installed Apache (httpd).
* Added webpage:
* HELLO THIS IS MY NETWORK-LOAD-BALANCER-EC2-2

Both instances:

* Are inside **VPC:** vpc-0f2fed5265d575471
* Subnets: **us-east-1a** and **us-east-1c**
* Security Group: **default**
* Port 80 open.

**2. Created Target Group for the Network Load Balancer**

created a new target group for the NLB.

**Target Group Settings**

| **Setting** | **Value** |
| --- | --- |
| **Target type** | Instances |
| **Name** | MY-NLB-TGP |
| **Protocol** | HTTP |
| **Port** | 80 |
| **IP version** | IPv4 |
| **VPC** | vpc-0f2fed5265d575471 |

**3. Configured Health Checks**

| **Setting** | **Value** |
| --- | --- |
| **Health Check Protocol** | HTTP |
| **Health Check Path** | /index.html |

This ensures the NLB checks the actual web page served.

**4. Registered EC2 Instances to Target Group**

added the two EC2s:

| **Instance ID** | **Name** | **Port** | **State** |
| --- | --- | --- | --- |
| i-05d3ba… | my-nlb-ec2-2 | 80 | Running |
| i-045368… | my-nlb-ec2-1 | 80 | Running |

Both were added and moved to **Healthy** later.

**5. Created the Network Load Balancer**

configured the NLB with the following settings:

**Basic Configuration**

| **Property** | **Value** |
| --- | --- |
| **Load Balancer Name** | MY-NLB |
| **Type** | Network Load Balancer |
| **Scheme** | Internet-facing |
| **IP address type** | IPv4 |

**6. Network Mapping**

selected VPC and subnets:

* **VPC:** vpc-0f2fed5265d575471
* **Subnets:**
  + us-east-1a (PUBLIC-SUBNET)
  + us-east-1c (PUBLIC-SUBNET-02)

**7. Selected Security Group**

NLB does not use SG for Layer 4, but console still shows default SG.

* Security Group: **default**

**8. Listener & Routing Configuration**

User configured:

**Listener**

| **Property** | **Value** |
| --- | --- |
| **Protocol** | HTTP |
| **Port** | 80 |

**Default Action**

Routing action:

Forward to target group → MY-NLB-TGP

Target group selected with:

* Weight: 1
* Percent: 100%

**9. NLB Resource Map Verification**

checked the resource map:

* Listener → HTTP:80
* Routing Rule → Forward to MY-NLB-TGP
* Target Group → MY-NLB-TGP
* Targets → 2 Healthy instances

Both targets show:  
 Healthy  
 Port 80

**10. Tested NLB DNS Name**

accessed:

my-nlb-726353798.us-east-1.elb.amazonaws.com

Refreshes alternated between outputs:

**EC2-1 Response**

HELLO THIS IS MY NETWORK-LOAD-BALANCER-EC2-1

**EC2-2 Response**

HELLO THIS IS MY NETWORK-LOAD-BALANCER-EC2-2

This confirmed **load balancing is working correctly**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a number

AI-generated content may be incorrect.

1. Attach SSL for application load balancer.

**1. Requested SSL Certificate in AWS Certificate Manager (ACM)**

**AWS Certificate Manager (ACM)**

**Requested a public certificate for the domain:**

mujju.store

**Validation Method Used:**

**DNS Validation (CNAME Records)**

**Certificate Status:**

**ISSUED** (Meaning validation succeeded)

This was confirmed in ACM dashboard.

**2. Route 53 Hosted Zone Verification**

opened Route 53 → Hosted Zone for **mujju.store** and verified:

**A, NS, SOA records exist**

**CNAME record for ACM validation exists**

\_39724c...mujju.store

→

\_d5f33c0...validation.aws.

Status: **SUCCESS**

This means the DNS validation for SSL certificate completed correctly.

**3. Creating an HTTPS Application Load Balancer (ALB)**

began creating a **new ALB** to support HTTPS traffic.

**Basic Configuration**

| **Setting** | **Value** |
| --- | --- |
| Load balancer name | **my-alb** |
| Scheme | Internet-facing |
| IP type | IPv4 |
| VPC | vpc-0f2fed5265d575471 |

**4. Network Mapping**

selected **two Availability Zones** and corresponding subnets:

* us-east-1a → PUBLIC-SUBNET
* us-east-1c → PUBLIC-SUBNET-02

This ensures high availability.

**5. Security Group Assignment**

selected:

**default security group**

**6. Listener Configuration**

created an HTTPS listener:

**Listener: HTTPS : 443**

| **Setting** | **Value** |
| --- | --- |
| Protocol | HTTPS |
| Port | 443 |
| Default Action | Forward to target group |
| Target Group | **my-alb-tgp** (already created earlier) |

**7. Secure Listener Settings (SSL/TLS Settings)**

This is the key part.

configured:

**Security Policy**

Selected recommended modern TLS policy:

ELBSecurityPolicy-TLS13-1-2-Res-Req-2025-09 (recommended)

**Certificate Source**

From ACM

**Selected ACM Certificate**

mujju.store

This certificate was the one that earlier became **ISSUED**.

**8. Final Review & Creation**

After configuring:

* Listener = HTTPS
* Routing = Forward to target group
* SSL certificate = mujju.store

clicked **Create load balancer**.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Map Application load balancer to R53.

**1. Opened Route 53 → Hosted Zone for mujju.store**

navigated to:

Route 53 → Hosted Zones → mujju.store

This is where DNS records for your domain are managed.

**2. Started Creating a New DNS Record**

Clicked:

Create record

This opened the "Quick create record" page.

**3. Created an A-Record for Domain (Root Domain Mapping)**

**Record Details Configured:**

| **Field** | **Value** |
| --- | --- |
| **Record name** | (blank) — meaning this record is for the root domain → mujju.store |
| **Record type** | A – Routes traffic to an IPv4 address and some AWS resources |
| **Alias** | Enabled (Yes) |

**4. Selected Alias Target → Application Load Balancer**

chose:

Alias to Application and Classic Load Balancer

Region: US East (N. Virginia)

Then selected HTTPS ALB:

dualstack.my-alb-10635285706.us-east-1.elb.amazonaws.com

This is the DNS name of your Application Load Balancer.

**5. Evaluated Target Health**

set:

Evaluate target health → Yes

This ensures Route 53 stops routing traffic if all ALB targets become unhealthy.

**6. Finalized the Record**

Clicked:

Create records

This created the DNS entry.

**7. Verified the DNS Entry**

In the Route 53 Hosted Zone view, you confirmed:

**A Record Created Successfully**

| **Column** | **Value** |
| --- | --- |
| Record name | mujju.store |
| Type | A |
| Routing policy | Simple |
| Alias | Yes |
| Value | dualstack.my-alb-10635285706.us-east-1.elb.amazonaws.com |

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Push the application load balancer logs to S3

**1. Created an S3 Bucket for ALB Logs**

opened **Amazon S3 → Create Bucket** and created:

alb-logs-bucket-s3-a

Bucket created successfully  
 Region: US East (N. Virginia)  
 No public access enabled (correct for logging buckets)

**2. Opened the Application Load Balancer (my-alb)**

navigated to:

EC2 → Load Balancers → my-alb

Then selected the:

**Attributes tab**

And clicked:

Edit

**3. Enabled ALB Logging Features**

Inside **Edit load balancer attributes**, you enabled:

**Access logs**

(This captures all HTTP/HTTPS requests that reach your ALB)

**Connection logs**

(This captures TCP connection details)

**Health Check logs**

(This logs all health check requests sent by ALB)

For all three, you selected the same S3 bucket:

s3://alb-logs-bucket-s3-a

Then clicked:

Save changes

This enabled the ALB to write logs to S3.

**4. Edited S3 Bucket Policy for Log Delivery**

To allow ALB to deliver logs, you opened the bucket:

alb-logs-bucket-s3-a

And edited the bucket policy to:

{

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

"Statement": [

{

"Sid": "AWSALBLogsDelivery",

"Effect": "Allow",

"Principal": {

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

},

"Action": [

"s3:PutObject",

"s3:PutObjectAcl"

],

"Resource": "arn:aws:s3:::alb-logs-bucket-s3-a/\*"

}

]

}

A success message confirmed:

“Successfully edited bucket policy.”

This policy is mandatory — without it, ALB logs cannot be delivered.

**5. Verified Logs Delivered to S3 Bucket**

opened the bucket folder structure:

alb-logs-bucket-s3-a → AWSLogs/ → 453275565364/

saw 3 log objects:

**elasticloadbalancing/ (main ALB access logs directory)**

**ELBAccessLogTestFile (test file created by AWS)**

**ELBHealthCheckLogTestFile (test file for health check logs)**

The existence of these files confirms:

**ALB logging is working successfully**

S3 is receiving log files from Load Balancer.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A white rectangular frame with text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.