

Session 14 HA II

Lab1: SnapGallery – Highly Available Web App with Shared Storage and S3 Error Page

Scenario Summary

The DevOps Team has been tasked to provide infrastructure support for one of our clients running a product, **SnapGallery**, which is preparing for a major product launch. The CTO requires a resilient AWS infrastructure to host their image-sharing web application with high availability, shared storage, and graceful failure handling.

Your task is to **build a production-ready web application infrastructure** on AWS that meets the following requirements:

- Two EC2 instances (in different Availability Zones) will host the Apache-based web application and serve image content.
- Both EC2 instances must mount a **shared Amazon EFS file system** to store user-uploaded content, ensuring consistency and availability.
- An Application Load Balancer (ALB) must distribute incoming traffic evenly across the two EC2 instances.
- In the event **that both EC2 instances become unavailable**, the ALB should automatically **redirect users to a static error page hosted on Amazon S3**.

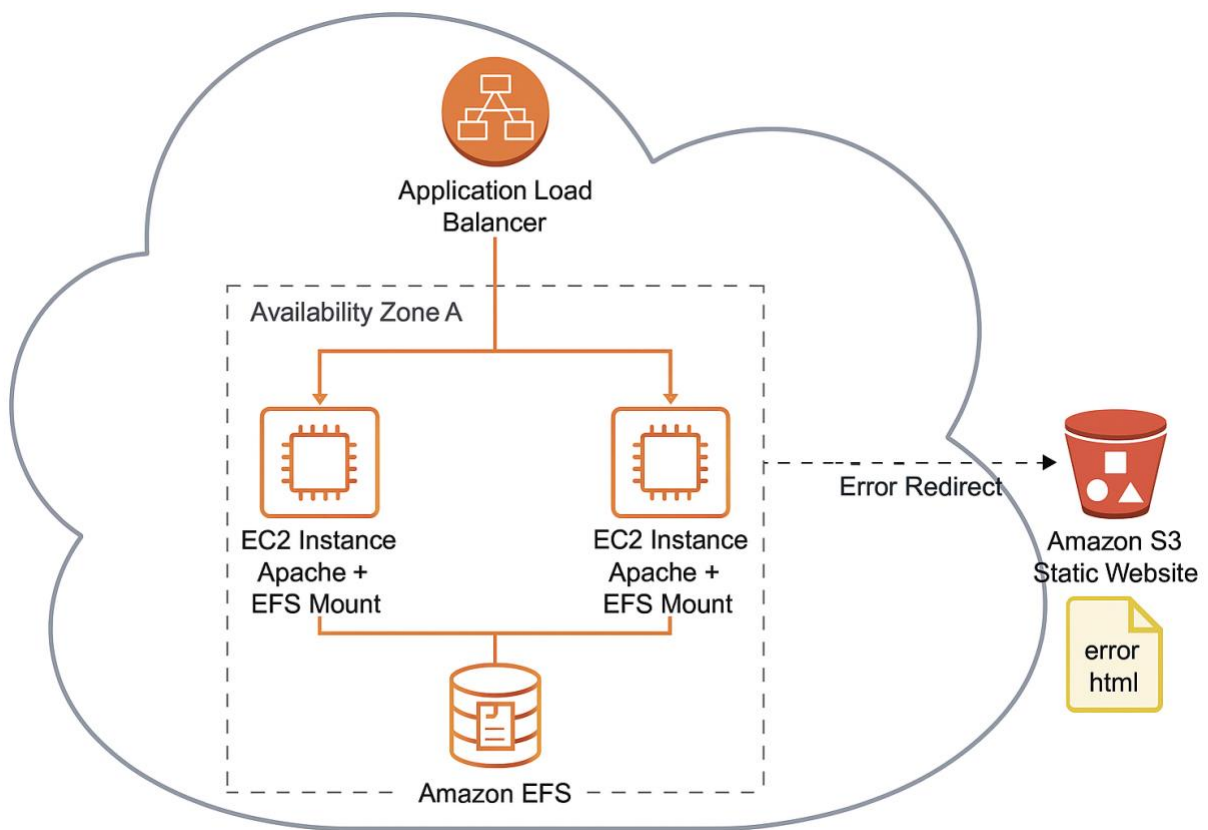
Your solution must be robust, scalable, and capable of handling production traffic, even in the face of instance failures or zone outages.

Resource Specifications

- A Virtual Private Cloud (VPC) with at least two public subnets located in different Availability Zones.
- Two Amazon EC2 instances running Amazon Linux 23 or Ubuntu 20.04, each with Apache installed and configured to serve web content.
- A shared Amazon EFS file system with mount targets in both Availability Zones and mounted to both EC2 instances.
- Security groups allowing HTTP access on port 80 from the Application Load Balancer, NFS access for EFS mounting, and SSH access from your IP.
- An Application Load Balancer with a target group containing both EC2 instances and configured health checks.

- A static S3 bucket with public access enabled for static website hosting and containing a custom error.html page.
- A listener rule in the ALB to redirect traffic to the S3 static website when the target group is unavailable.
- (Optional) An IAM role attached to EC2 instances for SSM access and limited S3 permissions, if needed for future automation.

Architecture Diagram



Section 1: Create Network Infrastructure

✅ Objective:

Set up a custom VPC with two public subnets across multiple AZs, internet access, and routing.

Steps:

1. **Create a VPC**
 - Name: snapgallery-vpc
 - CIDR: 10.0.0.0/16
 - Enable DNS hostnames
 2. **Create Subnets**
 - Public-Subnet-1 (AZ-a): 10.0.1.0/24
 - Public-Subnet-2 (AZ-b): 10.0.2.0/24
 3. **Create and Attach an Internet Gateway**
 - Name: snapgallery-igw
 - Attach it to snapgallery-vpc
 4. **Route Table**
 - Name: public-rt
 - Route: 0.0.0.0/0 → Internet Gateway
 - Associate with both public subnets
 5. **Enable Auto-assign Public IP**
 - For both subnets: Actions → Modify auto-assign IP → Enable
-

Section 2: Create Security Groups

Objective:

Allow traffic to/from EC2, EFS, and ALB securely.

Steps:

1. **Create Security Group: `sg-web-efs`**
 - Inbound rules:
 - HTTP (80) from 0.0.0.0/0
 - SSH (22) from your IP
 - NFS (2049) from self (sg-web-efs)
 - Outbound: All traffic
 2. **Create Security Group: `sg-alb`**
 - Inbound rules:
 - HTTP (80) from 0.0.0.0/0
 - Outbound: All traffic
-

Section 3: Create EFS File System

Objective:

Provision a shared file system accessible by both EC2 instances.

Steps:

1. Go to **EFS** → **Create File System**
 - o Name: snapgallery-efs
 - o VPC: snapgallery-vpc
2. **Add Mount Targets**
 - o AZ-a → Subnet-1 → Use sg-web-efs
 - o AZ-b → Subnet-2 → Use sg-web-efs

Network access

Network

Virtual Private Cloud (VPC) [Learn more](#)
Choose the VPC where you want EC2 instances to connect to your file system.

vpc-09edc9094e365dd56
test-vpc

Mount targets

A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. [Learn more](#)

Availability zone	Subnet ID	IP address type	IPv4 address	IPv6 address	Security groups	
us-east-1a	subnet-071db38...	IPv4 only	Optional	-	Choose security g... sg-02bfca9340e1825c b pub-private-sec-g	Remove
us-east-1b	subnet-0eb84c8...	IPv4 only	Optional	-	Choose security g... sg-02bfca9340e1825c b pub-private-sec-g	Remove
us-east-1c	subnet-0dd1a3e...	IPv4 only	Optional	-	Choose security g... sg-02bfca9340e1825c b pub-private-sec-g	Remove
us-east-1d	subnet-0aaebcc6...	IPv4 only	Optional	-	Choose security g... sg-02bfca9340e1825c b pub-private-sec-g	Remove

[Add mount target](#)

Ensure you select the correct security group that has ports http Port80, ssh Port22, and NFS Port2049 open

3. Leave performance and throughput settings as default.
4. Click **Create File System** and **note the EFS ID**.

Section 4: Launch EC2 Instances

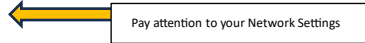
✓ Objective:

Launch **two** instances in different AZs, install Apache, and mount EFS.

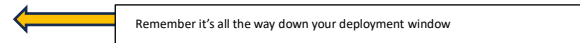
Step 1:

Repeat these for **two instances**, adjusting AZ and subnet.

- **AMI:** Amazon Linux 2023
- **Instance type:** t2.micro
- **Network:** snapgallery-vpc
- **Subnet:**
 - **Instance 1** → Subnet-1
 - **Instance 2** → Subnet-2
- **Auto-assign public IP:** Enabled
- **Security group:** sg-web-efs



User Data Script (update fs-xxxxxx with your EFS ID):



```
#!/bin/bash
yum install -y amazon-efs-utils httpd
mkdir -p /mnt/photos
mount -t efs <fs-xxxxxx>:/ /mnt/photos
ln -s /mnt/photos /var/www/html/photos
echo "<h1>Welcome to SnapGallery!</h1>" > /var/www/html/index.html
systemctl start httpd
systemctl enable httpd
```

- **Tag:**
 - **Name** = snapgallery-web-1
 - **Name** = snapgallery-web-2
- Repeat Step1 for second Instance

Breakpoint:

- Use **Instance Connect** to verify:

```
curl localhost
ls /mnt/photos
Output should look like this:
```

```
[ec2-user@ip-10-0-94-154 ~]$ curl localhost
ls /mnt/photos
<h1>Welcome to SnapGallery!</h1>
[ec2-user@ip-10-0-94-154 ~]$
```

Section 5: Configure Application Load Balancer

✓ Objective:

Distribute traffic to EC2 instances and monitor health.

Steps:

1. Go to **EC2** → **Load Balancers** → **Create Load Balancer** → **Application Load Balancer**

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

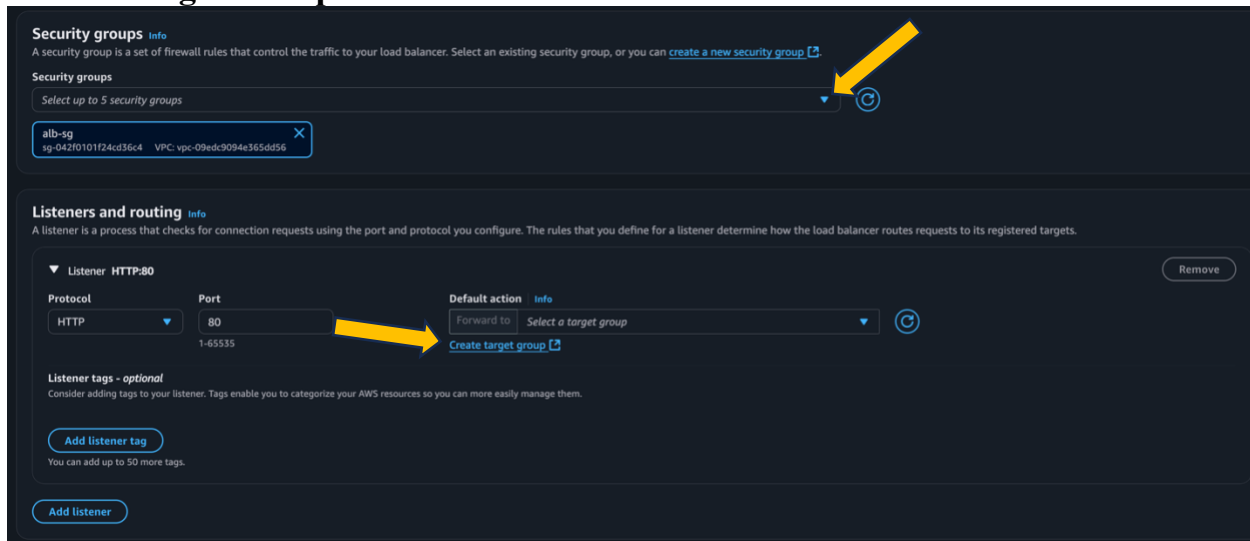
Load balancer types	Application Load Balancer Info	Network Load Balancer Info	Gateway Load Balancer Info
	<p>Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.</p> <p>Create</p>	<p>Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.</p> <p>Create</p>	<p>Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.</p> <p>Create</p>

► [Classic Load Balancer - previous generation](#)

[Close](#)

2. Name: snapgallery-alb
3. Scheme: Internet-facing
4. Listener: HTTP (port 80)
5. VPC: snapgallery-vpc
6. Subnets: Select both public subnets
7. Security group: sg-alb

Create Target Group



Security groups [info](#)
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups
Select up to 5 security groups

alb-sg
sg-042f0101f24cd36c4 VPC: vpc-09edc9094e365dd56

Listeners and routing [info](#)
A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener HTTP:80 [Remove](#)

Protocol HTTP **Port** 80 **Default action** Forward to [Select a target group](#) [Create target group](#)

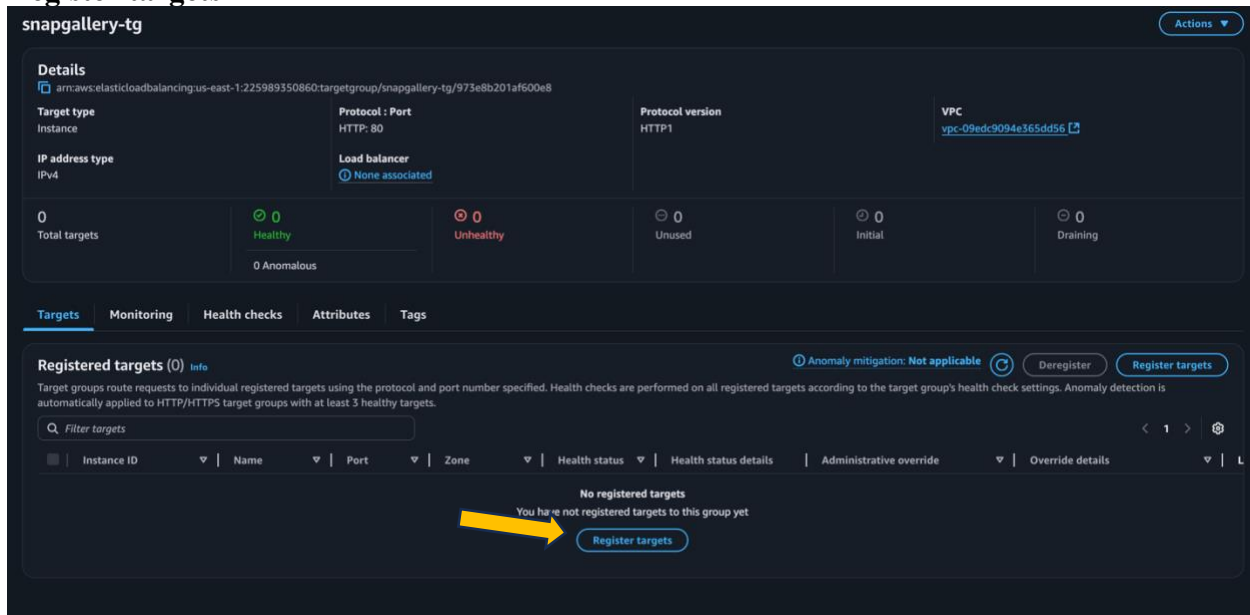
Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

[Add listener tag](#)
You can add up to 50 more tags.

[Add listener](#)

- Target type: instance
- Name: snapgallery-tg
- Protocol: HTTP
- Port: 80
- Health check path: /index.html

Register targets: Select both EC2 instances



snapgallery-tg [Actions](#)

Details
arn:aws:elasticloadbalancing:us-east-1:225989350860:targetgroup/snapgallery-tg/973e8b201af600e8

Target type Instance **Protocol : Port** HTTP: 80 **Protocol version** HTTP1 **VPC** vpc-09edc9094e365dd56

IP address type IPv4 **Load balancer** [None associated](#)

0 Total targets **0** Healthy **0** Unhealthy **0** Unused **0** Initial **0** Draining

0 Anomalous

Targets **Monitoring** **Health checks** **Attributes** **Tags**

Registered targets (0) [info](#) [Anomaly mitigation: Not applicable](#) [Deregister](#) [Register targets](#)

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

No registered targets
You have not registered targets to this group yet

[Register targets](#)

Register targets
Select instances, specify ports, and add the instances to the list of pending targets. Repeat to add additional combinations of instances and ports to the list of pending targets. Once you are satisfied with your selections, click Register pending targets.

Available instances (2/2)

Filter instances

<input checked="" type="checkbox"/>	Instance ID	Name	State	Security groups	Zone	Private IPv4 address	Subnet ID	Launch time
<input checked="" type="checkbox"/>	i-0222ac23618b4830a	Inst-2	Running	pub-private-sec-g	us-east-1d	10.0.110.12	subnet-0aaebcc6140101447	June 14, 2025, 1
<input checked="" type="checkbox"/>	i-0fca113ecbb6c1371	Inst-1	Running	pub-private-sec-g	us-east-1c	10.0.94.154	subnet-0dd1a3e6a25ffaab4	June 14, 2025, 1

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.

80
1-45535 (separate multiple ports with comma)

[Include as pending below](#)

[Review targets](#)

Review and register pending targets

Breakpoint:

”
A target group in AWS defines the EC2 instances that the Application Load Balancer routes traffic to and monitors their health; it ensures traffic is only sent to healthy instances and enables failover or redirection (e.g., to an S3 error page) when all targets become unhealthy.

- ALB → Target Groups → Check health = healthy
- ⇒ finish creating your ALB , we'll come back to it after s3

Section 6: S3 Static Site for Error Page

Objective:

Host a fallback error page in S3 to be used by ALB if all EC2s fail.

Steps:

1. Go to S3 → Create Bucket: `snappgallery-error-page`
 - Uncheck "Block all public access"

- Enable static website hosting(*Go into the created **bucket**>properties>enable **Static web hosting***)
- Index document: `error.html`

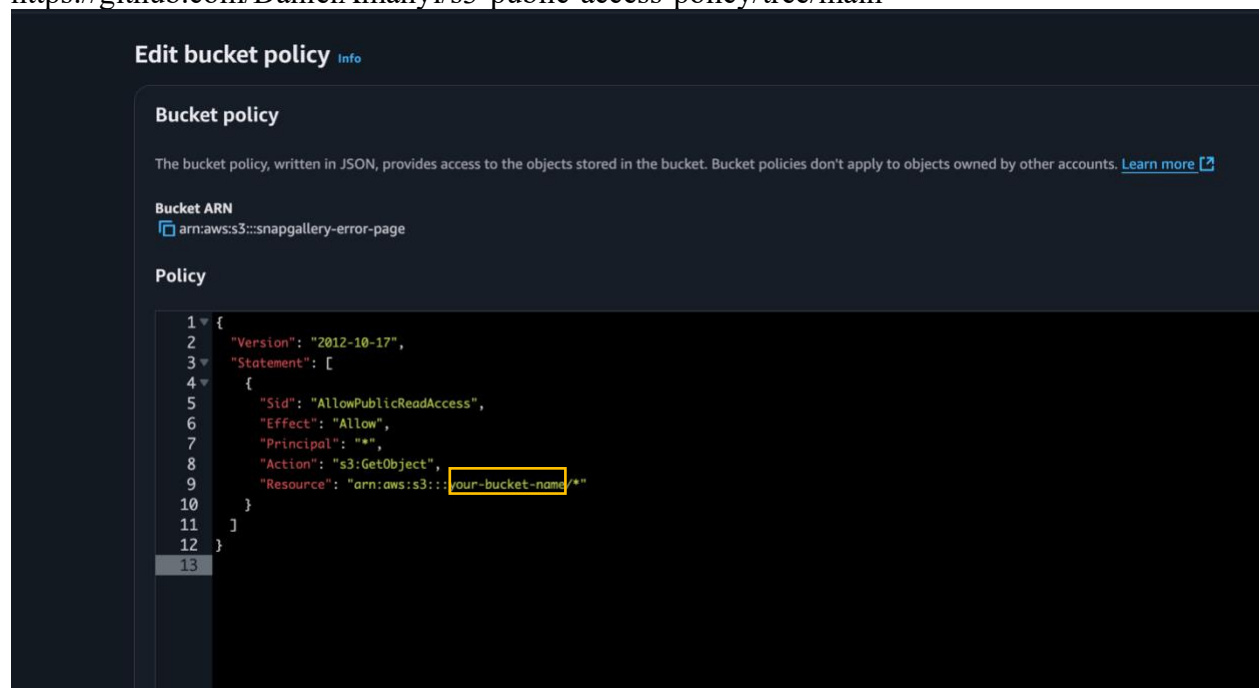
2. Upload `error.html`:

You can create a `.html` error page, or download one from <https://github.com/rapidruby/simple-error-pages?tab=readme-ov-file>

- ## 3. Set public read access:
- Use object permissions or bucket policy

(Remember to access through your permissions Tab)

<https://github.com/DanielAmanyi/s3-public-access-policy/tree/main>



4. Copy the S3 website URL.

Amazon S3 > Buckets > snapgallery-error-page

snapgallery-error-page Info

Objects | Metadata | **Properties** | Permissions | Metrics

Bucket overview

AWS Region
US East (N. Virginia) us-east-1

Object Lock
Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. Object Lock works only in versioned buckets.
Object Lock
Disabled

Requester pays
When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. [Learn more](#)
Requester pays
Disabled

Static website hosting
Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)
<http://snapgallery-error-page.s3-website-us-east-1.amazonaws.com>

Go back to your ALB TAB

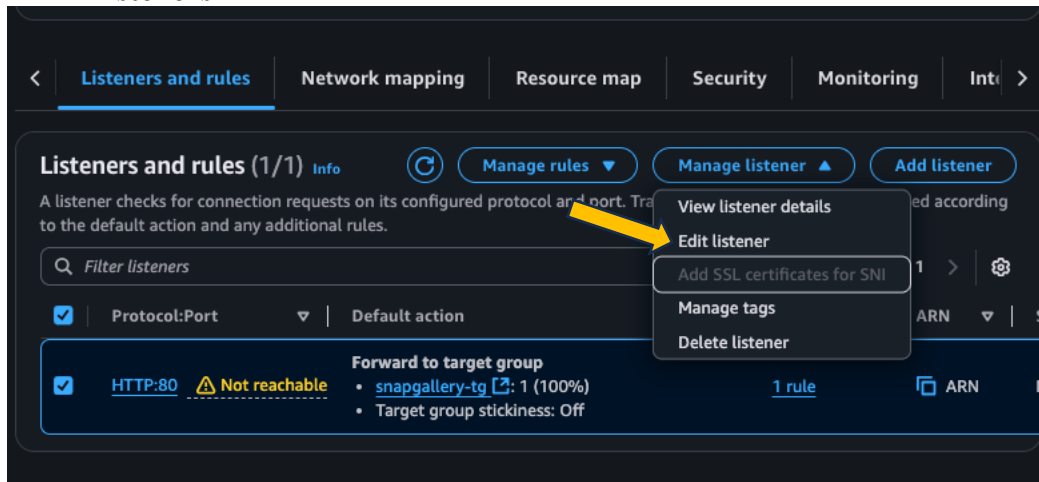
Section 7: Add ALB Error Response Redirect

✓ Objective:

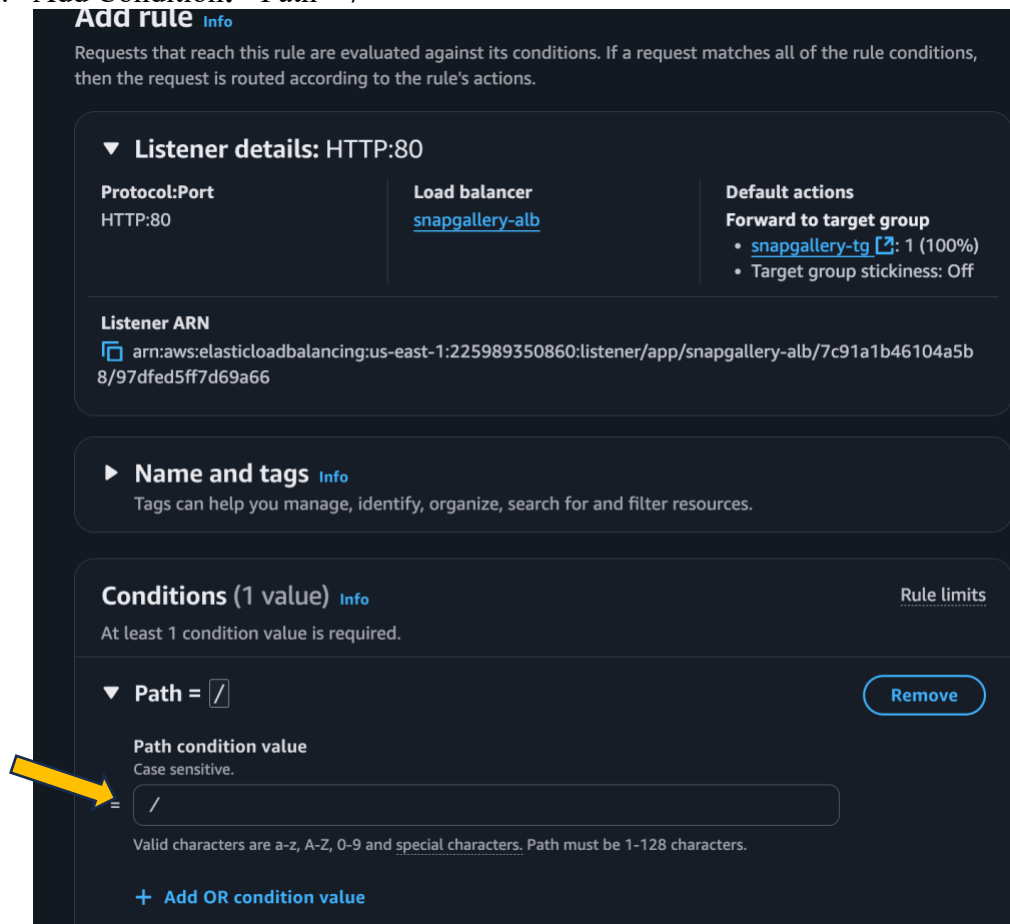
Make the ALB redirect to S3 error page when EC2s are unhealthy.

Steps:

1. EC2 → Load Balancers → Select snapgallery-alb
2. Go to **Listeners** → View/edit rules



3. Add Condition: >Path> /



- On HTTP 503, redirect to external URL

Default action [Info](#)
The default action is used if no other rules apply. Choose the default action for traffic on this listener.

Routing action

☐ Forward to target groups ☒ **Redirect to URL** ☐ Return fixed response

Redirect to URL [Info](#)
Redirect requests from one URL to another. You cannot redirect HTTPS to HTTP. To avoid a redirect loop, you must modify at least one of the following components: protocol, port, hostname or path. Components that you do not modify retain their original values.

☐ URI parts ☒ **Full URL**

Full URL [Info](#)
Enter the redirect URL.

protocol://hostname:port/path?query

Status code

▶ **Server-side tasks and status**
After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

[Cancel](#) [Save changes](#)

- Enter your S3 site URL (e.g., [http://snapgallery-error-page.s3-website-
<region>.amazonaws.com/error.html](http://snapgallery-error-page.s3-website-<region>.amazonaws.com/error.html))
- Priority>1

Section 8: Test the Setup

Go Back to ALB

✓ Objective:

The screenshot shows the AWS Management Console for an Elastic Load Balancing (ALB) instance named 'snapgallery-alb'. The console displays various details in a grid layout:

- Details**
 - Load balancer type:** Application
 - Status:** Active (with a green checkmark icon)
 - Scheme:** Internet-facing
 - Hosted zone:** Z35SXDOTRQ7X7K
 - VPC:** vpc-09edc9094e365dd56
 - Availability Zones:** subnet-0dd1a3e6a25ffaab4 (us-east-1c (use1-az1)), subnet-0aaebcc6140101447 (us-east-1d (use1-az2))
 - Load balancer IP address type:** IPv4
 - Date created:** June 14, 2025, 18:17 (UTC+01:00)
- Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:225989350860:loadbalancer/app/snapgallery-alb/7c91a1b46104a5b8
- DNS name Info:** snapgallery-alb-1545199321.us-east-1.elb.amazonaws.com (A Record)

A yellow arrow points from the 'Hosted zone' field to the 'DNS name Info' field.

Confirm content is served, EFS is shared, and error handling works.

1. Visit the **ALB DNS name** in your browser
 - Should load Welcome to SnapGallery!
2. On EC2 instance:

```
echo "New photo uploaded" > /mnt/photos/test.txt
```

3. On the other EC2:

```
cat /mnt/photos/test.txt
```

4. **Simulate failure:**
 - Stop both EC2s
 - Revisit ALB URL → You should be redirected to your S3 error page

Project Objective

The objective of this project is to reinforce your hands-on capabilities in the following key areas:

- Cloud Networking
- Virtual Machine Configuration
- Network File Storage
- Data Persistence
- Object Storage
- Distributed Systems
- High Availability

- Microservices