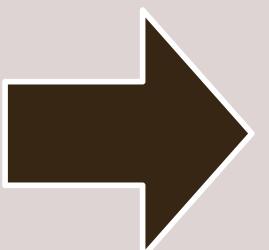


# E-Commerce CLOUD SERVICES



MEMBERS:  
CHIA YUE SHENG  
HII ZI WEI  
LEE HIEN LEONG  
TEH BEE LING  
MEGAN LIEW WAN LING

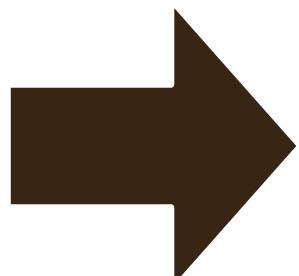


# INTRODUCTION



## Why E-Commerce?

- Digital shopping is booming
- Demand for fast, secure, scalable platforms.



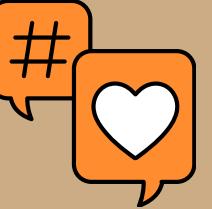
**GIFTOS**

HOME   SHOP   WHY US   TESTIMONIAL   CONTACT US   Login    

# Welcome To Our Gift Shop

Sequi perspiciatis nulla reiciendis, rem, tenetur impedit, eveniet non necessitatibus error distinctio mollitia suscipit. Nostrum fugit doloribus consequatur distinctio esse, possimus maiores aliquid repellat beatae cum perspicere enim accusamus perferendis.

# METHODOLOGY ALLIGN WITH PROBLEM STATEMENTS



**GIFTOS**

Latin English

HOME SHOP WHY US TESTIMONIAL CONTACT US Login

Welcome To Our Gift Shop

Sequi persipciatis nulla reiciendis, rem, tenetur impedit, eveniet non necessitatibus error distinctio mollitia suscipit. Nostrum fugit doloribus consequatur distinctio esse, possimus maiores aliquid repellat beatae cum, persipciatis enim, accusantium preferendis.

CONTACT US

Upload succeeded  
For more information, see the Files and folders table.

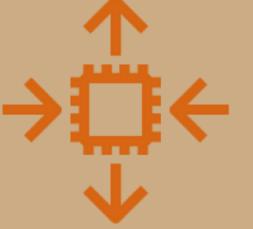
Files and folders (20 total, 1.5 MB)

Name	Folder	Type	Size	Status	Error
agency-img.jpg	images/	image/jpeg	158.0 KB	Succeeded	-
favicon.png	images/	image/png	27.4 KB	Succeeded	-
gifts.svg	images/	image/svg+xml	3.4 KB	Succeeded	-
high-quality.svg	images/	image/svg+xml	1.6 KB	Succeeded	-
line.png	images/	image/png	301.0 B	Succeeded	-
logo.png	images/	image/png	2.2 KB	Succeeded	-
p1.png	images/	image/png	71.7 KB	Succeeded	-
p2.png	images/	image/png	79.5 KB	Succeeded	-
p3.png	images/	image/png	114.7 KB	Succeeded	-

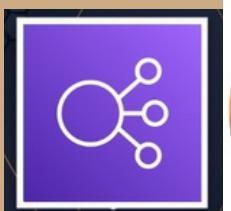
## HOST WEBSITE



EC2



Auto Scaling



ELB



ALB



Cloudwatch

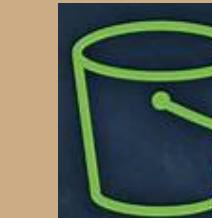
## PERFORMANCE



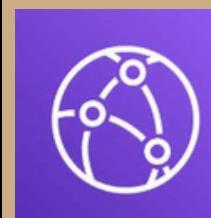
S3



S3 Glacier



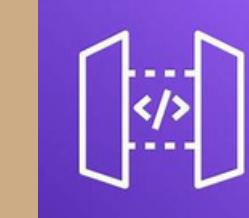
S3 Intelligent-Tiering



CloudFront

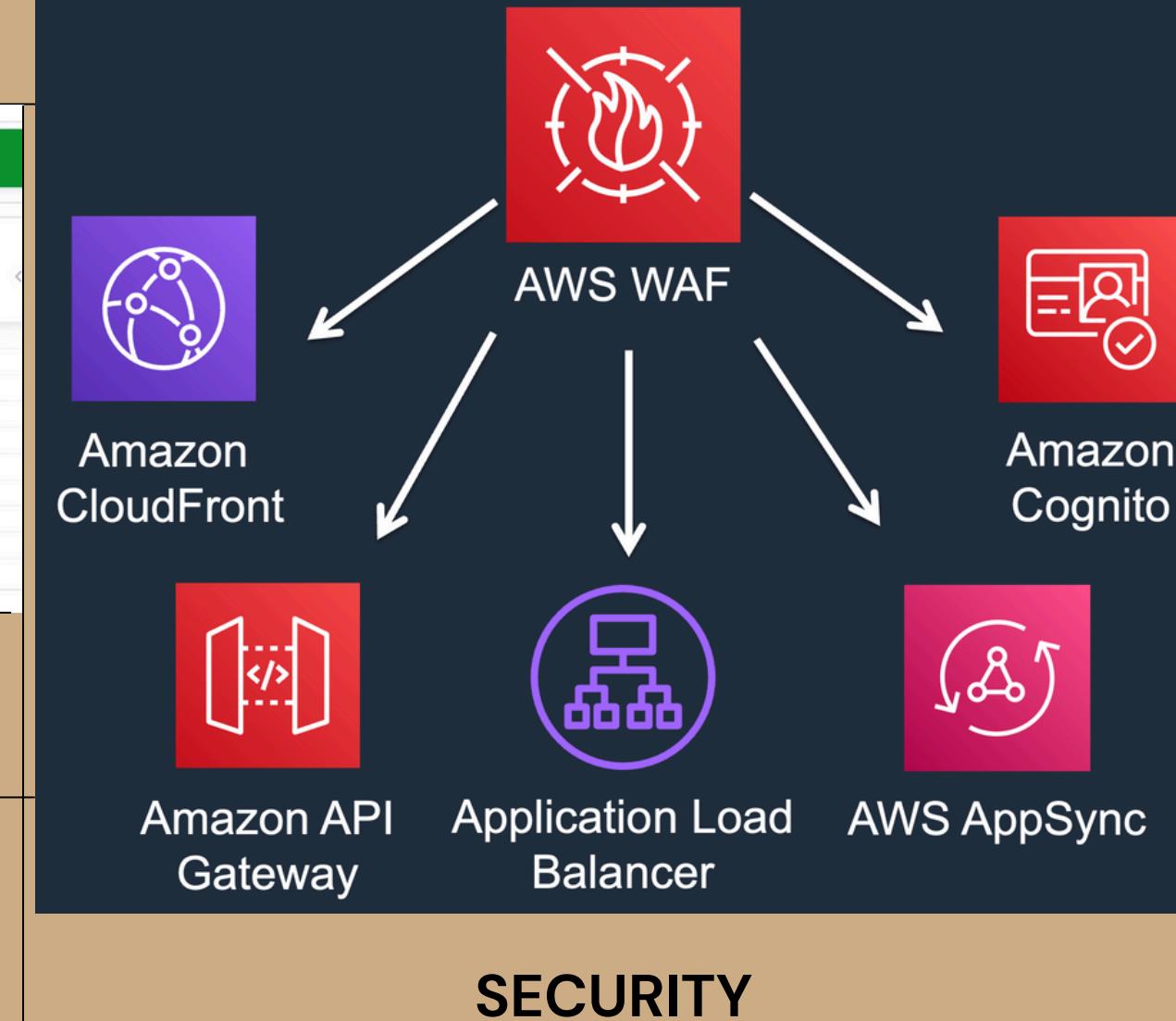


Lambda



API Gateway

## AWS WAF - Layer 7 Protection



WAF



Shield

# Architecture Diagram

HII ZI WEI 2204587

## User use different devices

- user interacts with website via HTTPS over the Internet using a domain. The browser/app first resolves the domain.

## User routed to closest AWS Region

## CloudFront acts as the entry point to check:

- static content
- dynamic content / API call

## Web Application Firewall (WAF) Filters Traffic

## Check Cache

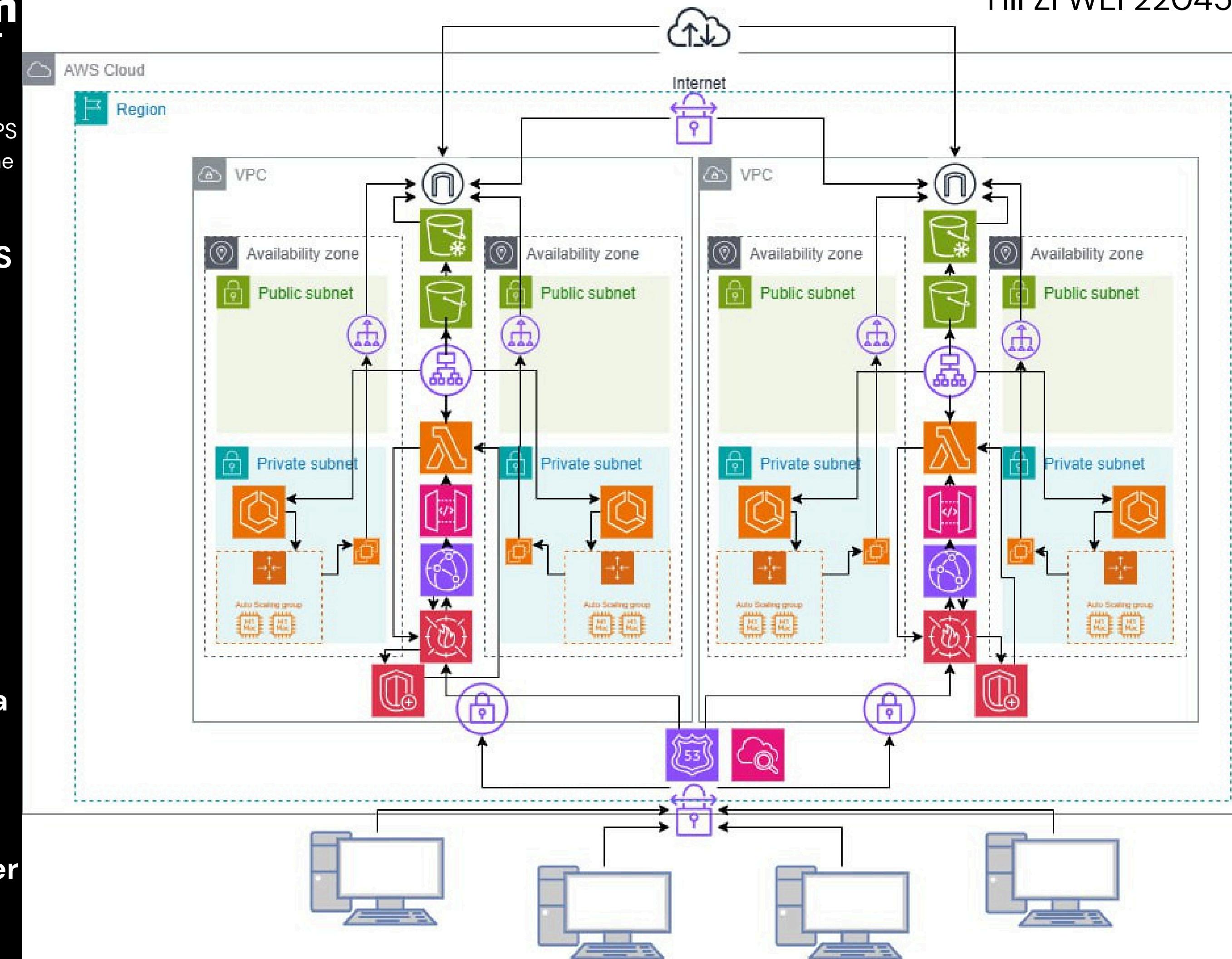
- yes - serve from edge location
- no - forward from origin (S3/ALB/API)

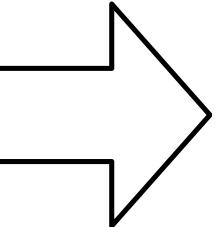
## Request Enters the VPC via Load Balancer

## EC2 in Private Subnet

## Processes Request

## Response Sent Back to User





# PROBLEM STATEMENTS

## Problem Statement 1:



95%



### 👎 Previously – Static Website

limits our ability to deliver personalized user experiences, real-time updates, and data-driven features



### Roadmap toward a fully functional platform in the coming



### EC2 Hosting:

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with 'EC2' selected and a list of options: Dashboard, EC2 Global View, Events, Instances (with 'Instances' expanded), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main area is titled 'Instances (1/1) Info' and shows a single instance named 'web-example' with ID 'i-077596dc7fe521d5f'. The instance is listed as 'Running' with an 't2.micro' instance type and '2/2 checks passed' status. Below the instance table, there's a section for 'i-077596dc7fe521d5f (web-example)' with tabs for 'Details', 'Status and alarms', 'Monitoring', 'Security', 'Networking', 'Storage', and 'Tags'. At the bottom, there's a 'Instance summary' section.

**Instance Type:** t2.micro (cost-effective)

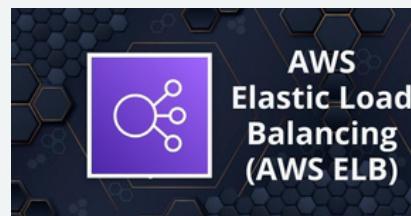
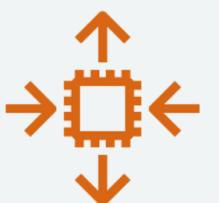
**Web Server:** Apache installed on the EC2 instance to serve web pages.

**Storage:** Default instance storage for website files.

**Network:** Public IP + Security Group (opened Port 80 for HTTP traffic).

## Problem Statement 2:

### ♥ HIGH TRAFFIC & SCALABILITY ISSUES.



TEH BEE LING 2204237

Sales/Promotional Campaigns → Slow Loading Time

Step 1  
 Choose launch template

Step 2  
 Choose instance launch options

Step 3 - optional  
 Integrate with other services

Step 4 - optional  
 Configure group size and scaling

Step 5 - optional  
 Add notifications

Step 6 - optional  
 Add tags

Step 7  
 Review

**Launch template** Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

**Launch template**  
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

MyTemplate

Create a launch template Info

**Version**  
Default (1)

Create a launch template version Info

**Auto Scaling group**

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

**Attach to an existing load balancer**  
Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups  
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

**Existing load balancer target groups**  
Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

first-tg | HTTP Application Load Balancer: alb

**Desired capacity**  
Specify your group size.  
1

**Scaling** Info  
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

**Scaling limits**  
Set limits on how much your desired capacity can be increased or decreased.

**Min desired capacity** 1 Equal or less than desired capacity

**Max desired capacity** 2 Equal or greater than desired capacity

**Scaling group**  
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies  
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy  
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

**Scaling policy name** Target Tracking Policy

**Metric type** Info  
Monitored metric that determines if resource utilization is too low or high. Using EC2 metrics, consider enabling detailed monitoring for better scaling performance.

Average CPU utilization

**Target value** 80

**Instance warmup** Info  
300 seconds

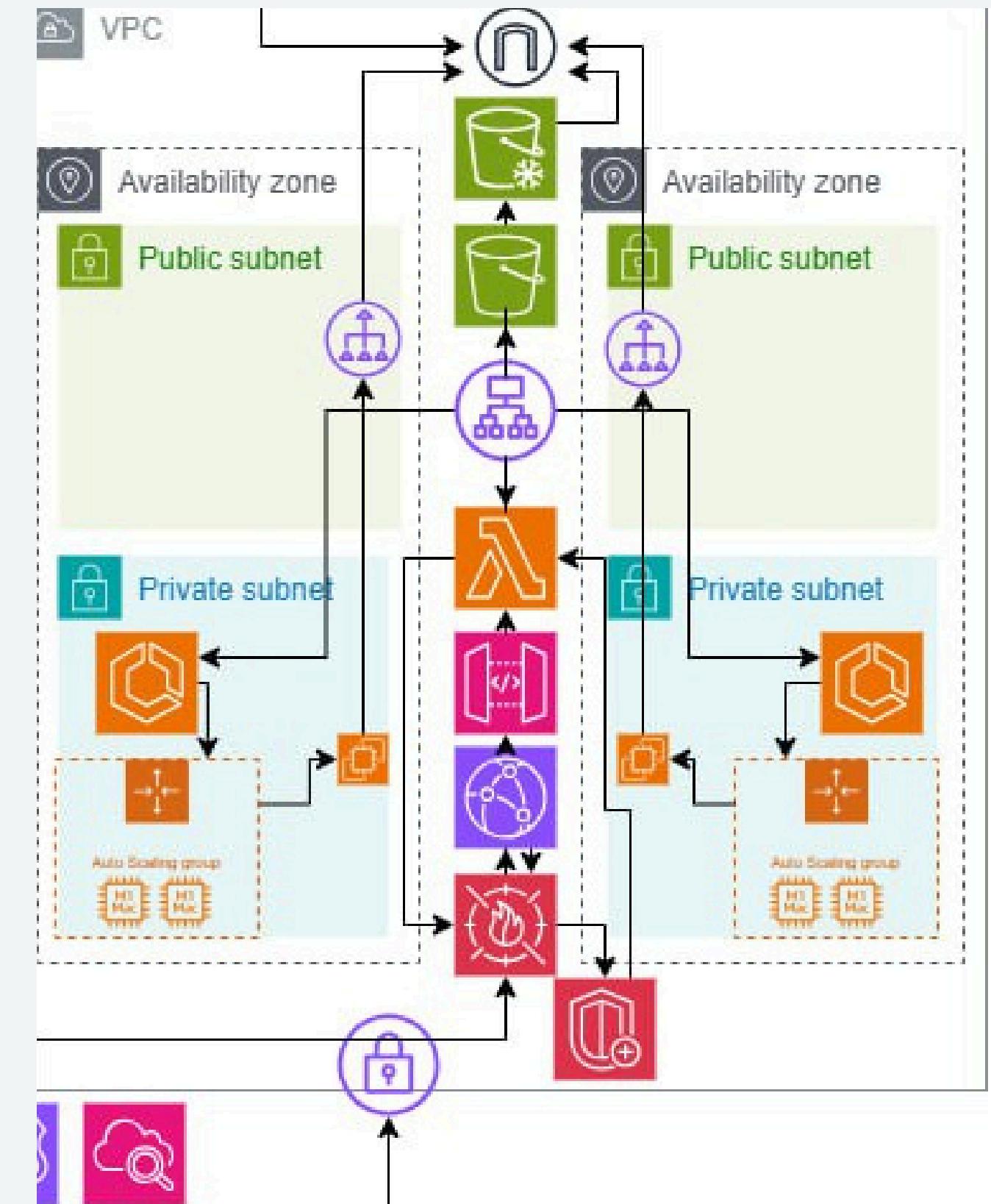
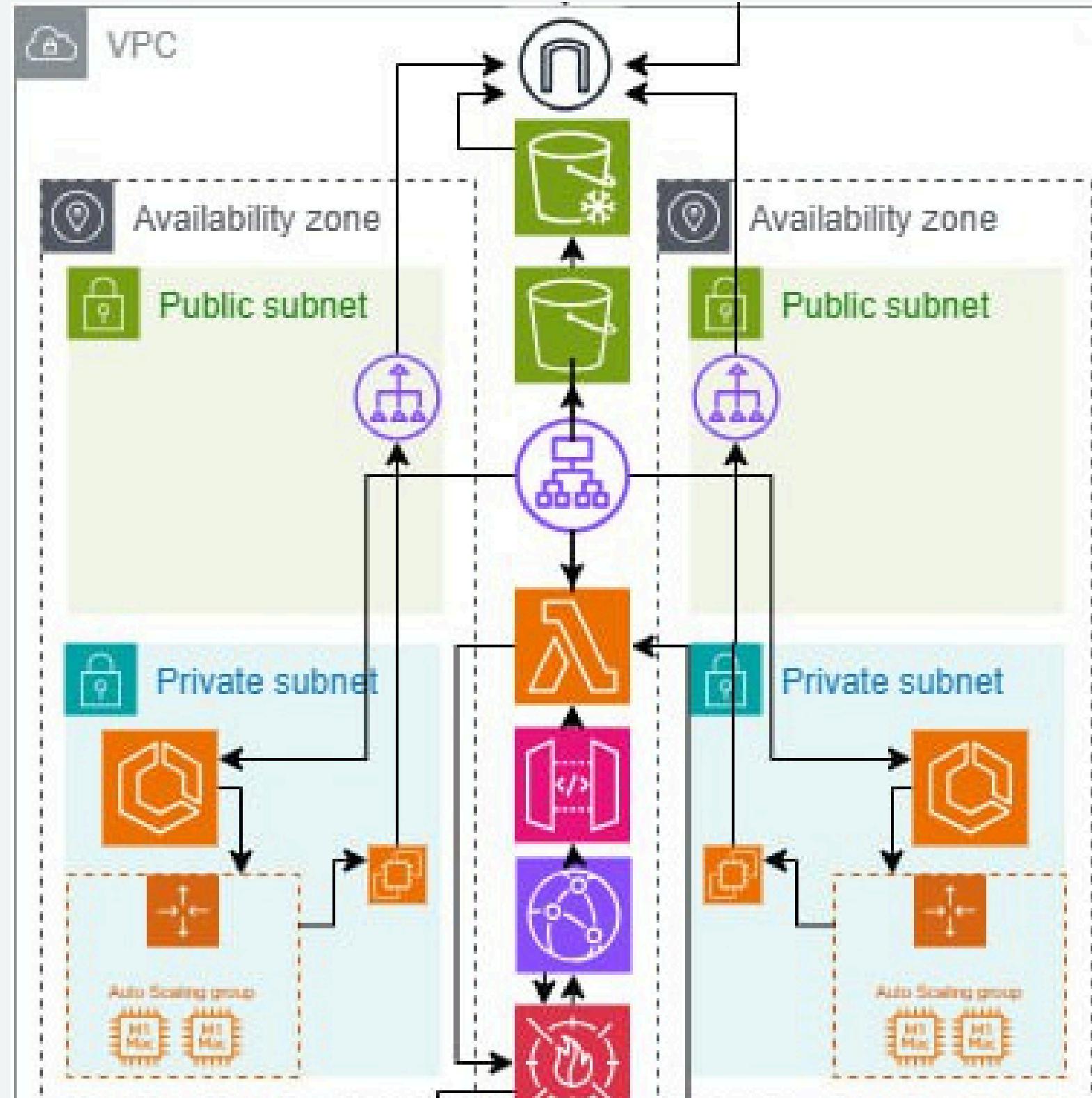
## Problem Statement 7:

### ♥ NO ALERTING OR VISIBILITY INTO SERVER ISSUES.



# Architecture Diagram

TEH BEE LING 2204237



# Problem Statement 3:

WEBSITE LOADS SLOWLY DUE TO LARGE STATIC FILES.



HII ZI WEI 2204587

The screenshot shows two main sections. On the left, the 'Create bucket' wizard is displayed with the following details:

- General configuration**:
  - AWS Region**: US East (N. Virginia) us-east-1
  - Bucket type**: General purpose (selected)
  - Bucket name**: my-static-assets-bucket
- Bucket name validation**: Bucket names must be 3 to 63 characters and unique within the global name space.

On the right, the 'my-static-assets-bucket' objects list is shown, featuring:

- Objects (0)**: A table with columns for Name, Type, Last modified, and Size. It displays a message: "No objects. You don't have any objects in this bucket."
- Actions buttons: Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, Upload.

## Edit Block public access (bucket settings)

### Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

#### Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

#### Block public access to buckets and objects granted through new access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

#### Block public access to buckets and objects granted through any access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

#### Block public access to buckets and objects granted through new public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

#### Block public and cross-account access to buckets and objects through any public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Cancel

Save changes

Created bucket and Changed Permission.

## Edit bucket policy

### Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Buckets

#### Bucket ARN

arn:aws:s3:::my-static-assets-bucket

### Policy

```
1 {  
2   "Version": "2012-10-17",  
3   "Statement": [  
4     {  
5       "Sid": "PublicReadGetObject",  
6       "Effect": "Allow",  
7       "Principal": "*",  
8       "Action": "s3:GetObject",  
9       "Resource": "arn:aws:s3:::my-static-assets-bucket/*"  
10    }  
11  ]  
12 }
```

# Connect With EC2 instances → Linux → Offload the static content in EC2.

```
Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

Last login: Wed Apr 30 17:08:58 2025 from 18.206.107.27
[ec2-user@ip-172-31-85-59 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-85-59 ~]$ ls
temp
[ec2-user@ip-172-31-85-59 ~]$ ^C
[ec2-user@ip-172-31-85-59 ~]$ ^[[200~ls /var/www/html
-bash: $'\E[200~ls': command not found
[ec2-user@ip-172-31-85-59 ~]$ ~ls /var/www/html
-bash: ~ls: command not found
[ec2-user@ip-172-31-85-59 ~]$ ls /var/www/html
contact.html css fonts images index.html js shop.html testimonial.html why.html
[ec2-user@ip-172-31-85-59 ~]$ cd /var/www/html
[ec2-user@ip-172-31-85-59 html]$ ls
contact.html css fonts images index.html js shop.html testimonial.html why.html
[ec2-user@ip-172-31-85-59 html]$ sudo sed -i 's|src="images/|src="https://my-static-assets-bucket.s3.amazonaws.com/ig' *.html
[ec2-user@ip-172-31-85-59 html]$ █ I
```



`sudo sed -i 's|src="images/|src="https://mt-static-assets-bucket.s3.amazonaws.com/ig' *.html`

- to offload the assets from EC2 to S3, by integrating S3 Assets in EC2 Website
- searches for any src="images/ paths in the HTML and replaces them with the full S3 bucket URL so that image references point to your static assets in S3



# Create CloudFront Distribution

CloudFront > Distributions > Create

**Amazon S3**

- giftos.s3bucket.s3.amazonaws.com
- my-static-assets-bucket.s3.amazonaws.com

**Elastic Load Balancer**

- my-alb
- alb

**API Gateway**

No origins available.

**Mediastore container**

No origins available.

**Mediapackage container**

Choose origin

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or custom origin.

**EPRNF51V3FQ9I Standard**

**General** **Security** **Origins**

## Details

**Distribution domain name**

d3jh5pg84cn11g.cloudfront.net



**Objects (20)**

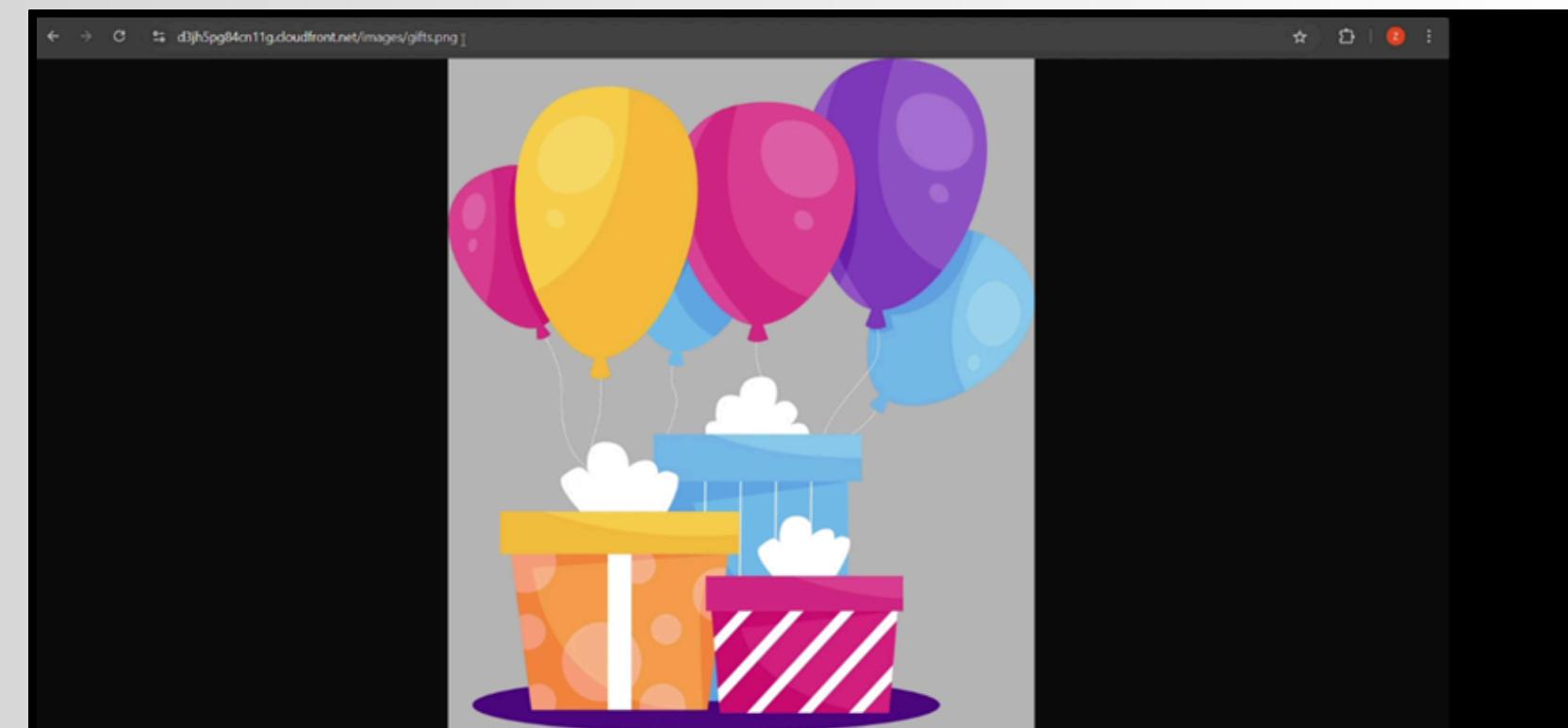
Copy S3 URI  Copy URL  Download  Open  Delete  Actions  Create folder  Upload

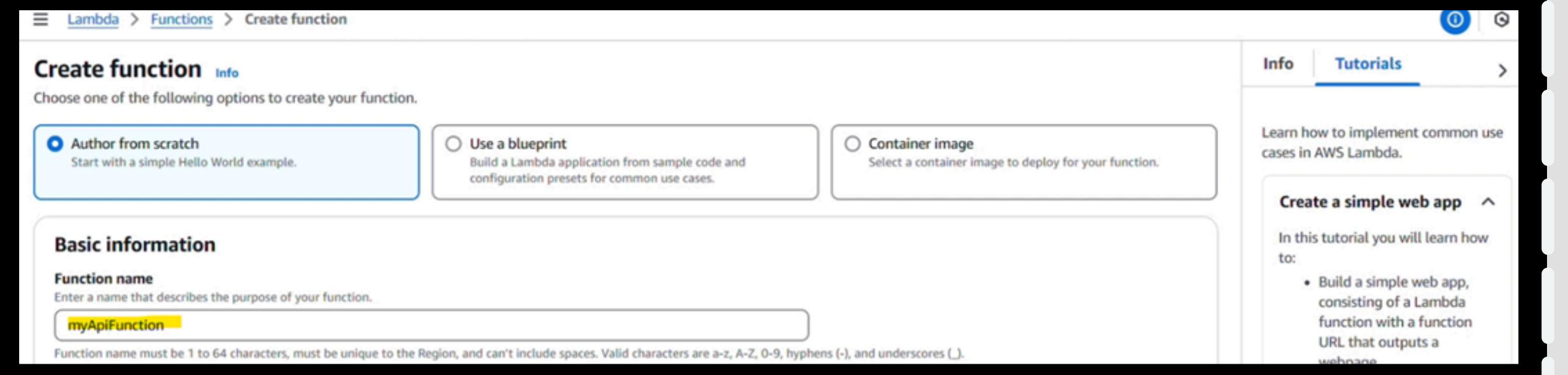
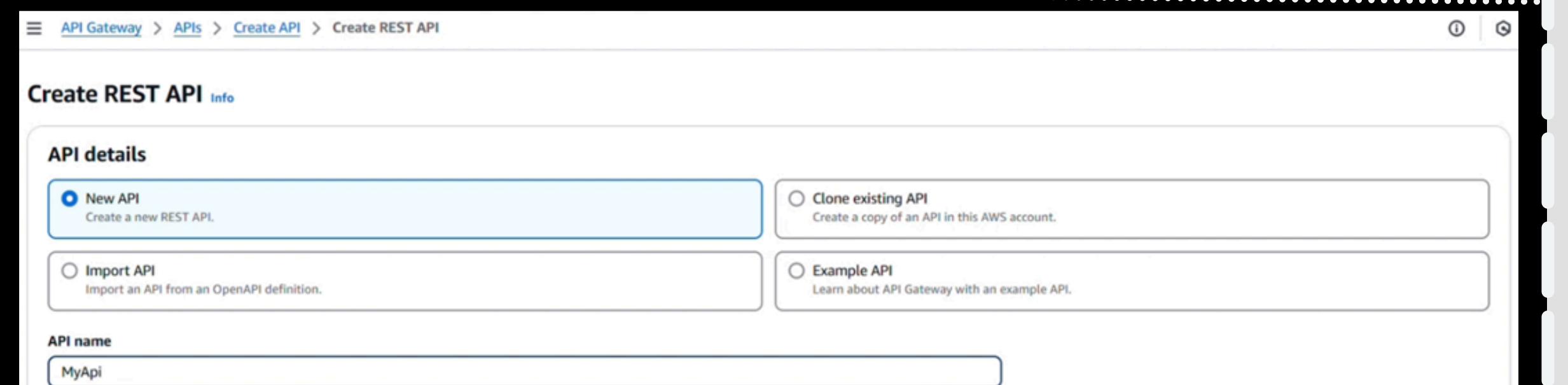
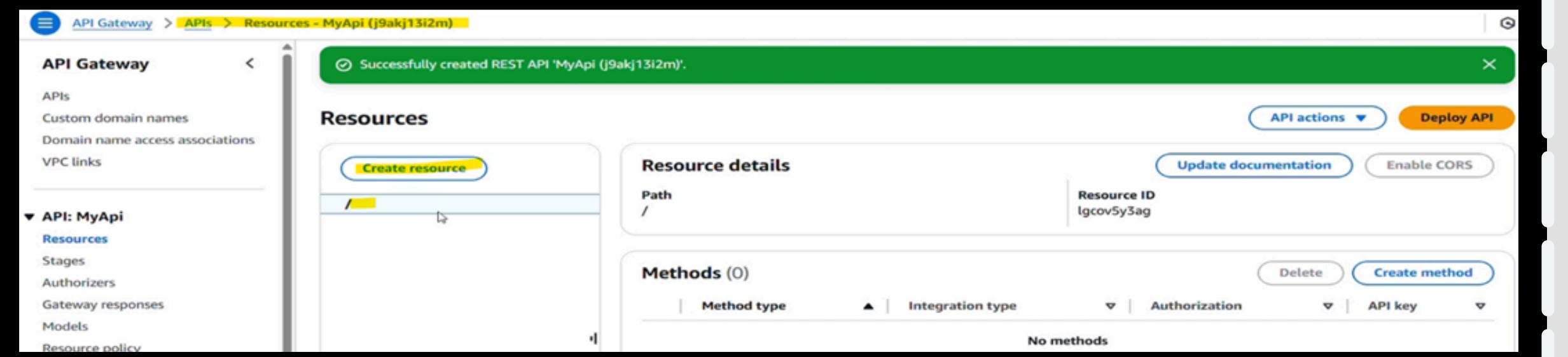
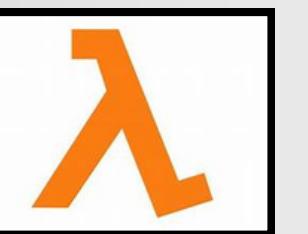
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">favicon.png</a>	png	May 1, 2025, 01:25:10 (UTC+08:00)	27.4 KB	Standard
<input type="checkbox"/>	<a href="#">free.svg</a>	svg	May 1, 2025, 01:25:19 (UTC+08:00)	3.4 KB	Standard
<input type="checkbox"/>	<a href="#">gifts.png</a>	png	May 1, 2025, 01:25:20 (UTC+08:00)	107.5 KB	Standard

[d3jh5pg84cn11g.cloudfront.net/images/gifts.png](#)



# CREATE FUNCTION



# CREATE REST API & RESOURCES

### Create resource

**Resource details**

**Proxy resource** [Info](#)  
Proxy resources handle requests to all sub-resources. To create a proxy resource use a path parameter that ends with a plus sign, for example {proxy+}.

**Resource path**  **Resource name**

**CORS (Cross Origin Resource Sharing)** [Info](#)  
Create an OPTIONS method that allows all origins, all methods, and several common headers.

[Cancel](#) [Create resource](#)

API Gateway > APIs > Resources - MyApi (j9akj13i2m) > Create method

**Successfully created resource '/products'**

### Create method

**Method details**

**Method type**

**Integration type**

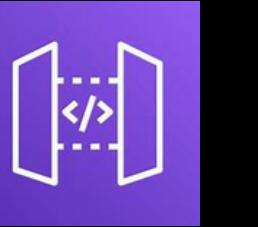
- Lambda function**  
Integrate your API with a Lambda function. 
- HTTP**  
Integrate with an existing HTTP endpoint. 
- Mock**  
Generate a response based on API Gateway mappings and transformations. 

- AWS service**  
Integrate with an AWS Service. 
- VPC link**  
Integrate with a resource that isn't accessible over the public internet. 

**Lambda function**  
Provide the Lambda function name or alias. You can also provide an ARN from another account.

**Grant API Gateway permission to invoke your Lambda function**  
When you save your changes, API Gateway updates your Lambda function's resource-based policy to allow this API to invoke it.

**Integration timeout** [Info](#)  
By default, you can enter an integration timeout of 50 - 29,000 milliseconds. You can use Service Quotas to raise the integration timeout to greater than 29,000 ms



### Deploy API

Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. [Learn more](#)

**Stage**

**Stage name**

**Note** A new stage will be created with the default settings. Edit your stage settings on the Stage page.

**Deployment description**

[Cancel](#) [Deploy](#)

## Stages

The screenshot shows the 'Stages' section of the AWS API Gateway console. A new stage, 'MyApiStage', has been created. The 'Stage details' section displays the following information:

- Stage name:** MyApiStage
- Rate Info:** 10000
- Cache cluster Info:** Inactive
- Burst Info:** 5000
- Default method-level caching:** Inactive

An 'Invoke URL' is provided: <https://j9akj13i2m.execute-api.us-east-1.amazonaws.com/MyApiStage>.

Stage Created.

Triggers correct.

The screenshot shows the 'Configuration' tab for a Lambda function named 'myApiFunction'. On the left, a sidebar lists various configuration sections: General configuration, Triggers, Permissions, Destinations, Function URL, Environment variables, Tags, VPC, RDS databases, and Monitoring and operations tools. The 'Triggers' section is currently selected.

Under the 'Triggers' section, there is one entry:

- Trigger:** API Gateway: MyApi
- API endpoint:** <https://j9akj13i2m.execute-api.us-east-1.amazonaws.com/MyApiStage/products>
- Details:**
  - API type: REST
  - Authorization: NONE
  - isComplexStatement: No
  - Method: GET
  - Resource path: /products
  - Service principal: apigateway.amazonaws.com
  - Stage: MyApiStage
  - Statement ID: e38f296a-be9f-591e-8244-3f02ac4a2495

# Test to retrieve gifts.png

Code | **Test** | Monitor | Configuration | Aliases | Versions

Code source [Info](#)

Upload from ▾

myApiFunction

EXPLORER

MYAPIFUNCTION

index.mjs

DEPLOY

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

index.mjs

```
JS index.mjs X
JS index.mjs > [e] handler
1 import { S3Client, HeadObjectCommand } from '@aws-sdk/client-s3';
2
3 const s3 = new S3Client({
4   region: 'us-east-1',
5   logger: console
6 });
7
8 export const handler = async () => {
9   const BUCKET = 'my-static-assets-bucket';
10  const KEY = 'images/gifts.png';
11
12  try {
13    console.log(`Attempting to access s3://${BUCKET}/${KEY}`);
14
15    const response = await s3.send(new HeadObjectCommand({
16      Bucket: BUCKET,
17      Key: KEY
18    }));
19  }
```

Execution Results

✓ Executing function: succeeded ([Logs](#))

▼ Details

```
{  
  "status": "SUCCESS",  
  "metadata": {  
    "lastModified": "2025-04-30T17:25:20.000Z",  
    "size": 110099,  
    "contentType": "image/png"  
  }  
}
```

**Summary****Code SHA-256**  
xu4aRwvQv7GiB3iZ6+IAcbq0iZuaolU2YqMCkArD0AM=**Function version**  
[\\$LATEST](#)**Duration**  
585.81 ms**Resources configured**  
128 MB**Execution time**

1 second ago

**Request ID**

90b2128b-db29-4c7a-a8dc-1accf6eb2c9f

**Billed duration**

586 ms

**Max memory used**

101 MB

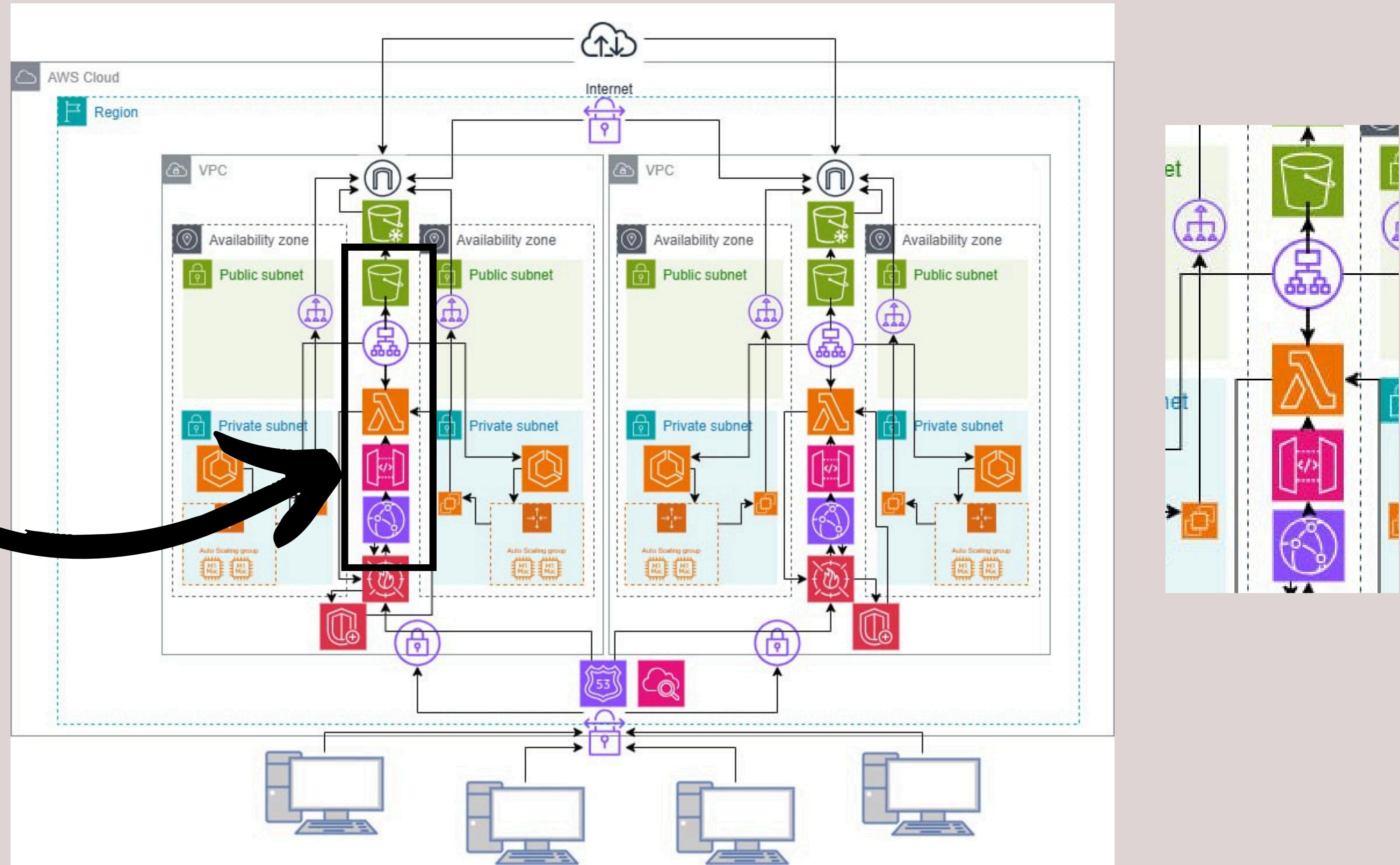
**Log output**The area below shows the last 4 KB of the execution log. [Click here](#) to view the corresponding CloudWatch log group.

```
START RequestId: 90b2128b-db29-4c7a-a8dc-1accf6eb2c9f Version: $LATEST  
2025-05-01T10:54:41.765Z 90b2128b-db29-4c7a-a8dc-1accf6eb2c9f INFO Attempting to access s3://my-static-assets-bucket/images/gifts.png  
2025-05-01T10:54:42.307Z 90b2128b-db29-4c7a-a8dc-1accf6eb2c9f INFO {  
  clientName: 'S3Client',  
  commandName: 'HeadObjectCommand',  
  input: { Bucket: 'my-static-assets-bucket', Key: 'images/gifts.png' },  
  output: {  
    AcceptRanges: 'bytes',  
    LastModified: 2025-04-30T17:25:20.000Z,  
    ContentLength: 110099,
```

# Test Successful.

# Architecture Diagram

HII ZI WEI 2204587



# Problem Statement 5:

♥ COSTLY STORAGE FOR INFREQUENTLY ACCESSED MEDIA.



Amazon S3 > Buckets > my-static-assets-bucket > Lifecycle configuration > Create lifecycle rule

**Lifecycle rule configuration**

**Lifecycle rule name**  
MoveToGlacier  
Up to 255 characters

**Choose a rule scope**  
 Limit the scope of this rule using one or more filters  
 Apply to all objects in the bucket

**Filter type**  
You can filter objects by prefix, object tags, object size, or whatever combination suits your usecase.

**Prefix**  
Add filter to limit the scope of this rule to a single prefix.  
old-assets/

Don't include the bucket name in the prefix. Using certain characters in key names can cause problems with some applications and protocols. [Learn more](#)

**Lifecycle rule actions**  
Choose the actions you want this rule to perform.

Transition current versions of objects between storage classes  
This action will move current versions.  
 Transition noncurrent versions of objects between storage classes  
This action will move noncurrent versions.  
 Expire current versions of objects  
 Permanently delete noncurrent versions of objects  
 Delete expired object delete markers or incomplete multipart uploads  
These actions are not supported when filtering by object tags or object size.

**Transitions are charged per request**  
For a lifecycle transition action, each request corresponds to an object transition. For details on lifecycle transition pricing, see requests pricing info on the requests pricing info on the Storage & requests tab of the [Amazon S3 pricing page](#).  
 I acknowledge that this lifecycle rule will incur a transition cost per request

**By default, objects less than 128KB will not transition across any storage class**  
We don't recommend transitioning objects less than 128 KB because the transition costs can outweigh the storage savings. If your use case requires transitioning objects less than 128 KB, specify a minimum object size filter for each applicable lifecycle rule with a transition action.

**Choose storage class transitions**  
Glacier Instant Retrieval  
**Days after objects become noncurrent**  
90  
**Number of newer versions to retain - Optional**  
Number of versions  
Remove  
Can be 1 to 100 versions. All other noncurrent versions will be moved.  
**Add transition**

**Review transition and expiration actions**

**Current version actions**  
Day 0  
No actions defined.

**Noncurrent versions actions**  
Day 0  

- Objects become noncurrent

↓

Day 90  

- 0 newest noncurrent versions are retained
- All other noncurrent versions move to Glacier Instant Retrieval

## Lifecycle Rule 1: Move to Glacier

- Non-current version
- old-assets
- 90 days

Amazon S3 > Buckets > my-static-assets-bucket > Lifecycle configuration > Create lifecycle rule

### Create lifecycle rule Info

**Lifecycle rule configuration**

**Lifecycle rule name**  
MoveToIntelligentTiering  
Up to 255 characters

**Choose a rule scope**  
 Limit the scope of this rule using one or more filters  
 Apply to all objects in the bucket

**Lifecycle rule actions**  
Choose the actions you want this rule to perform.

Transition current versions of objects between storage classes  
This action will move current versions.

Transition noncurrent versions of objects between storage classes  
This action will move noncurrent versions.

Expire current versions of objects

Permanently delete noncurrent versions of objects

Delete expired object delete markers or incomplete multipart uploads  
These actions are not supported when filtering by object tags or object size.

**Choose storage class transitions**  
Intelligent-Tiering

**Days after object creation**  
30

**Review transition and expiration actions**

**Current version actions**

Day 0  
• Objects uploaded  
↓  
Day 30  
• Objects move to Intelligent-Tiering

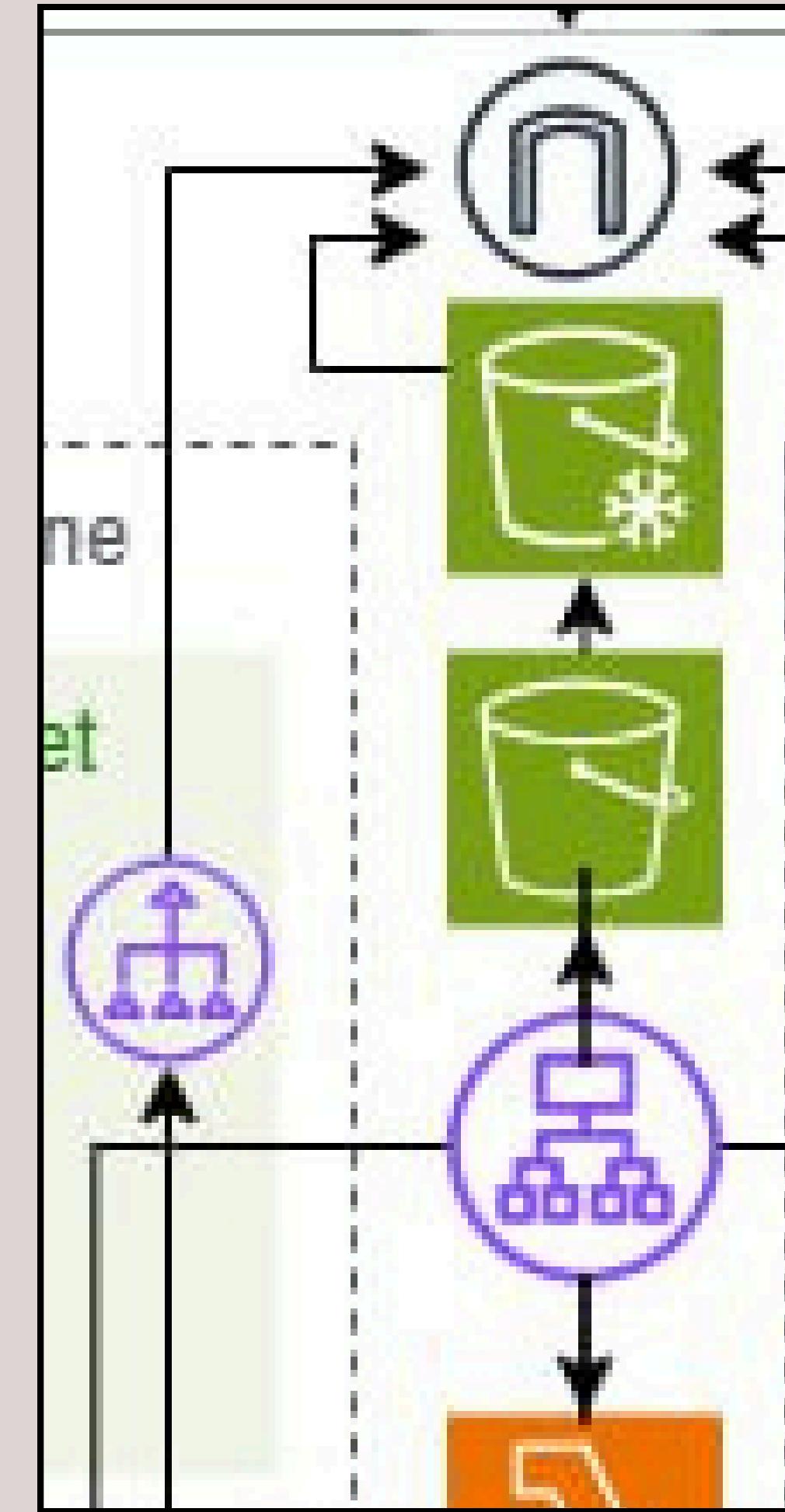
**Noncurrent versions actions**

Day 0  
No actions defined.

## Lifecycle Rule 2: Move to Intelligent-Tiering

- Current version
- all objects
- 30 days

# Architecture Diagram



## Problem Statement 4: Dynamic Content Performance Bottleneck

**Website struggles to  
handle dynamic content efficiently**

**All dynamic requests are processed by a  
single EC2 instance during:**

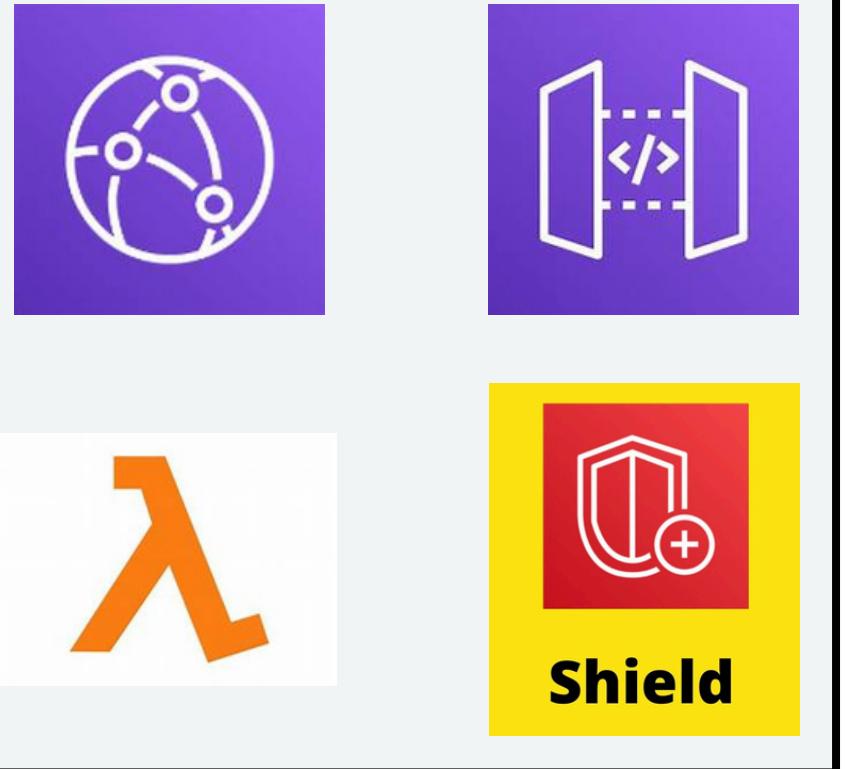
- Order placements (buyers)
- Product updates (sellers)
- Authentication and notifications  
(developers)

**Results in:**

- Slow response time
- User frustration
- Poor scalability under load

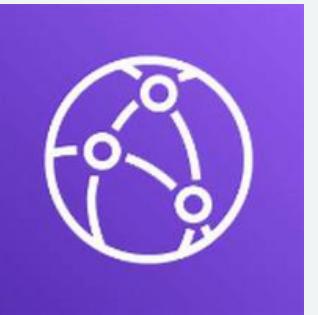
**Solution?**

**Offload dynamic tasks**



# Solution 4: Serverless Dynamic Handling with AWS

## Offload dynamic tasks using Lambda, API Gateway, and CloudFront



### CloudFront:

[CloudFront > Distributions > Create](#)

Choose the type of distribution that best fits your needs

Single website or app  
Choose if you have a single app or website

**Origin**

**Origin domain**  
Choose an AWS origin, or enter your origin's domain name. [Learn more](#)

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin ID.

**Protocol** [Info](#)  
 HTTP only  
 HTTPS only  
 Match viewer

**HTTP port**  
Enter your origin's HTTP port. The default is port 80.  
80

**EC2 instance public IPv4**

**Name**  
Enter a name for this origin.  
**EC2-Origin**

**Add custom header - optional**  
CloudFront includes this header in all requests that it sends to your origin.

[Add header](#)

**Enable Origin Shield**  
Origin shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.

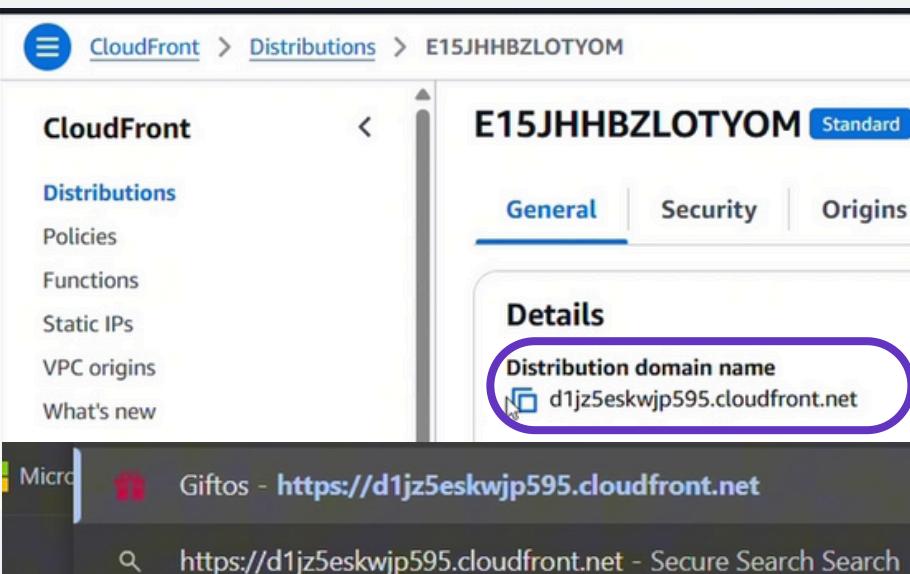
No  
 Yes

**Origin Shield region**  
Choose origin shield region.  
**US East (N. Virginia) us-east-1**

**Enable Origin Shield**

**Testing**

{



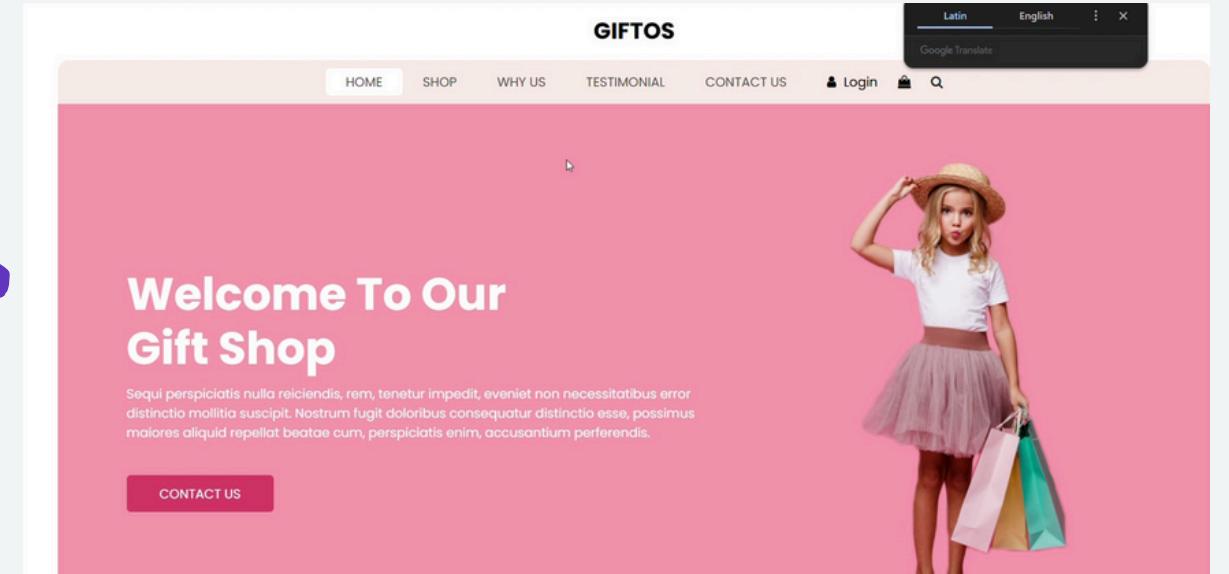
[CloudFront > Distributions > E15JHHBZLOTYOM](#)

**E15JHHBZLOTYOM** Standard

**General** **Security** **Origins**

**Details**

Distribution domain name  
<https://d1jz5eskwp595.cloudfront.net>



**GIFTOS**

HOME SHOP WHY US TESTIMONIAL CONTACT US Login

Welcome To Our Gift Shop

CONTACT US

# Solution 4: Serverless Dynamic Handling with AWS

## Offload dynamic tasks using Lambda, API Gateway, and CloudFront



### Lambda + API Gateway:

Lambda > Functions > Create function

Choose one of the following options to create your function:

- Author from scratch  
Start with a simple Hello World example.
- Use a blueprint  
Build a Lambda application from sample code and configuration presets for common use cases.
- Use a template  
Create a Lambda function based on a template provided by a partner.

**Basic information**

**Function name**  
Enter a name that describes the purpose of your function.  
**EC2-Proxy-Lambda**

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (\_).

**Runtime** | **Info**  
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.  
**Node.js 22.x**

```
[ec2-user@ip-172-31-85-59 ~]$ # Install Node.js and npm
sudo yum install -y nodejs npm

# Create the layer
mkdir -p axios-layer/nodejs
cd axios-layer/nodejs
npm init -y --silent
npm install axios

# Zip the layer
cd ..
zip -r axios-layer.zip nodejs

# Upload directly to Lambda Layers
aws lambda publish-layer-version \
--layer-name axios-layer \
--zip-file fileb:/axios-layer.zip \
--compatible-runtimes nodejs18.x nodejs20.x \
--region us-east-1
```

**i-0869726b2c261080e (web-example)**

PublicIPs: 3.86.243.76 PrivateIPs: 172.31.85.59

Create a Lambda Function

Install Node.js to the EC2

# Solution 4: Serverless Dynamic Handling with AWS

## Offload dynamic tasks using Lambda, API Gateway, and CloudFront

### Lambda + API Gateway:

```
ec2-user@ip-172-31-85-59 axios-layer]$ aws lambda publish-layer-version --layer-name axios-layer --zip-file fileb://axios-layer.zip --compatible-runtimes nodejs8.x --region us-east-1
"Content": {
  "Location": "https://prod-04-2014-layers.s3.us-east-1.amazonaws.com/snapshots/718213540041/axios-layer-974150ea-a5ab-41d1-b2d5-41577c7f79f8?versionId=0L36IwypGVRkx8VETNPalQx0.08yyS8-x-Amz-Security-Token=IQoJb3JpZ2luX2VjECEaCXvzLWVhc3QtMSJHMEUCID7wFsf9Iu7wfrAqjVJFPOoLv%2BBTMf7cOBW4qC8Di4FdAiEA1K51p2CFB36UVds%2FEflbTn0k9GkLoI19A2qh5ONM%2F0quAUUv%2F%2F%2F%2F%2F%2F%2FARAAGw3NDk2Nzg5MDI4Mzk1DpYFRqh5aP722JjSqMBdeVicm2rr0ORGzE4FDvUvQJK3Akalm7B3vJXYWUzWNnfzPuukiNZ9xOGYvkocWp4ZstTEM%29IotVuA%2B815OUS98p51Wgjou2nbMwlHEIR2F2FnhAhY1%2FG3mhBCC7q2mc5McjfxiF32vtzb4UsqbenReoPVWGel4Z2XNZqNKKD1H3pwy28RVG0tioGgwGMqECN6N31v4w2aE8z4eOH2%2BW1v77HdVrARY%2BZr0g8DULw6K7w40rx91QXPQOk5x76HP7PmgBfqnh9pMw0RLXkdO2aHx4eXFeKB7%2FxTxCckjzdxpF7%2BRdXY8P2KtHG5s%2Bw5zW1EJavukyWmtWH6f7WxtxtguXWG%2FmCP0hZXQasN0jDsyCPKJ2Noc5LDaJfw89czzbigWI6JKfoafkERRB%2BfGi%2F2kn6Ib4S5lgJ79pQT%2BacKjBzvBxQz8DpnDjrjLK3tmjNPfOxxR2%2BERkUqjAcYQJCrJfuOBKOrjbgwG6zcTgzRWqAPdoQ0mlJ91vnbpJd62dAtGYYwC2Eoonxx1vnTJrP%2FiQpvpt4Ft3QY60ngsH2KJciUfc82FRXqg1lvjz7TLSD82BGl3ZSzDhpKEEjTAlpqahFlUDPyxGCYL4zePjQnt8XR8hEQig6EBQQjL76GYmAGM47nUTX%2FBunWN4RAyrk2KUB7JJxS36UMEggw6DLR2mkWkn8pCETgtN7qko%2BYIV%2FaZhtLezKqDyZms90AO1syIIh6MoBzlW0qr06xfqvvee7YooCEYG8nX%2Fve2VBMCeBOKSzzD6Cpo9YQD11GSvz2ipQbx6B%2BNxFo8KRHAV3eyLI1kYQ6JmYldyt3OfogB6u11i01eFswuDKYBWYkZyXI9KnQjBspkw7uzMwAY6sQG351itHrw96M2X1qEEEJsGVJZb0PsNSYGMiS13qjh7QihXoU4oih9r6TVODgSu%2BBk%2Fjlyw%2FBEUsVNzXMn508Dc5Tvj733pTT6c1%2Fd94o8La2BaGp0B%2F3YQABr3QqEsux8c1LSHcwmOPWizE9jew%2FZXBmvVLW6pgmhNkAU99kHgTpWTrb%2BuLIooysODjk5XSL%2FN5tfUkLarqIDrw7hgQDb1GscXE%3D6X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20250501T090206&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credential=ASIA25DCYHY35TW7N23X%2F20250501%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=a41806573fc8eed8b024a55eaef298632d92433d723efd02c4e238a217a0f277",
  "CodeSha256": "EkKrtVcLVZJ5soKS1K7Ko0eFlUQOJArdltpJu6nZjbc=",
  "CodeSize": 853187
},
"LayerArn": "arn:aws:lambda:us-east-1:718213540041:layer:axios-layer",
"LayerVersionArn": "arn:aws:lambda:us-east-1:718213540041:layer:axios-layer:1",
"Description": "",
"CreatedDate": "2025-05-01T09:02:10.427+0000",
"Version": 1,
"CompatibleRuntimes": [
  "nodejs18.x"
]
}
```

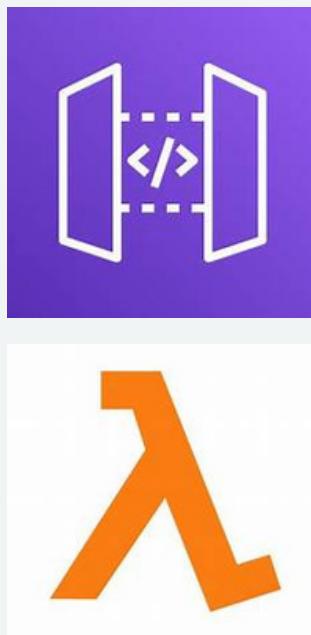
Modules of Axios

Add to layer of Lambda

The screenshot shows the AWS Lambda console interface. On the left, a modal window titled 'Add layer' is displayed. It contains 'Function runtime settings' (Runtime: Node.js 22.x, Architecture: x86\_64) and a 'Choose a layer' section. In the 'Custom layers' tab, 'axios-layer' is selected from a dropdown menu. At the bottom of the modal, there is a red circle around the 'Add' button. On the right, the main Lambda function page is visible, showing the 'EC2-Proxy-Lambda' function with its code properties, runtime settings (Runtime: Node.js 22.x, Handler: index.handler), and layers section.

# Solution 4: Serverless Dynamic Handling with AWS

## Offload dynamic tasks using Lambda, API Gateway, and CloudFront



### Lambda + API Gateway:

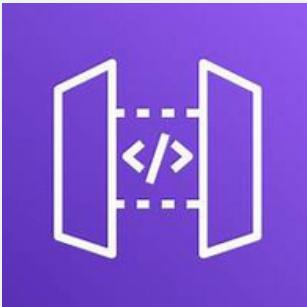
Screenshot of the AWS API Gateway 'Create REST API' page. It shows three options: 'New API' (selected), 'Import API', and 'Example API'. The 'API name' field contains 'EC2-Proxy-API'. The background features a yellow and green abstract pattern.

Screenshot of the AWS API Gateway 'Resources - EC2-Proxy-API' page. A green success message says 'Successfully created resource '/reviews''. The left sidebar shows the API structure: 'API: EC2-Proxy-API' with 'Resources', 'Stages', 'Authorizers', etc. The main panel shows a tree view of resources: '/' (POST /notify), '/auth' (POST), '/notify' (POST), '/reviews' (GET). On the right, the '/notify - POST - Method execution' details are shown, including the ARN: arn:aws:execute-api:us-east-1:718213540041:ne9uriic7k/\*/POST/notify, Resource ID 3kt637, and a flow diagram showing the request path from Client to Lambda integration.

# Solution 4: Serverless Dynamic Handling with AWS

## Offload dynamic tasks using Lambda, API Gateway, and CloudFront

### Lambda + API Gateway:



**Deploy API**

Create or select a stage where your API will be deployed. You can use the deployment history to revert or change the active deployment for a stage. [Learn more](#)

**Stage**

\*New stage\*

**Stage name**

prod

A new stage will be created with the default settings. Edit your stage settings on the [Stage page](#).

**Deployment description**

**Cancel Deploy**

**API Gateway > APIs > EC2-Proxy-API (ne9uriic7k) > Stages**

**API Gateway**

APIs  
Custom domain names  
Domain name access associations  
VPC links

**Stages**

Successfully created deployment for EC2-Proxy-API. This deployment is active for prod.

**Notifications**

**Stage actions** **Create stage**

**prod**

**Stage details**

Stage name: prod  
Rate Info: 10000  
Cache cluster Info: Inactive  
Burst Info: 5000  
Default method-level caching: Inactive

**Web ACL**  
-

**Client certificate**  
-

**Lambda > Functions > EC2-Proxy-Lambda**

**Configuration**

**General configuration**

**Triggers (3)**

**Find triggers**

**Trigger**

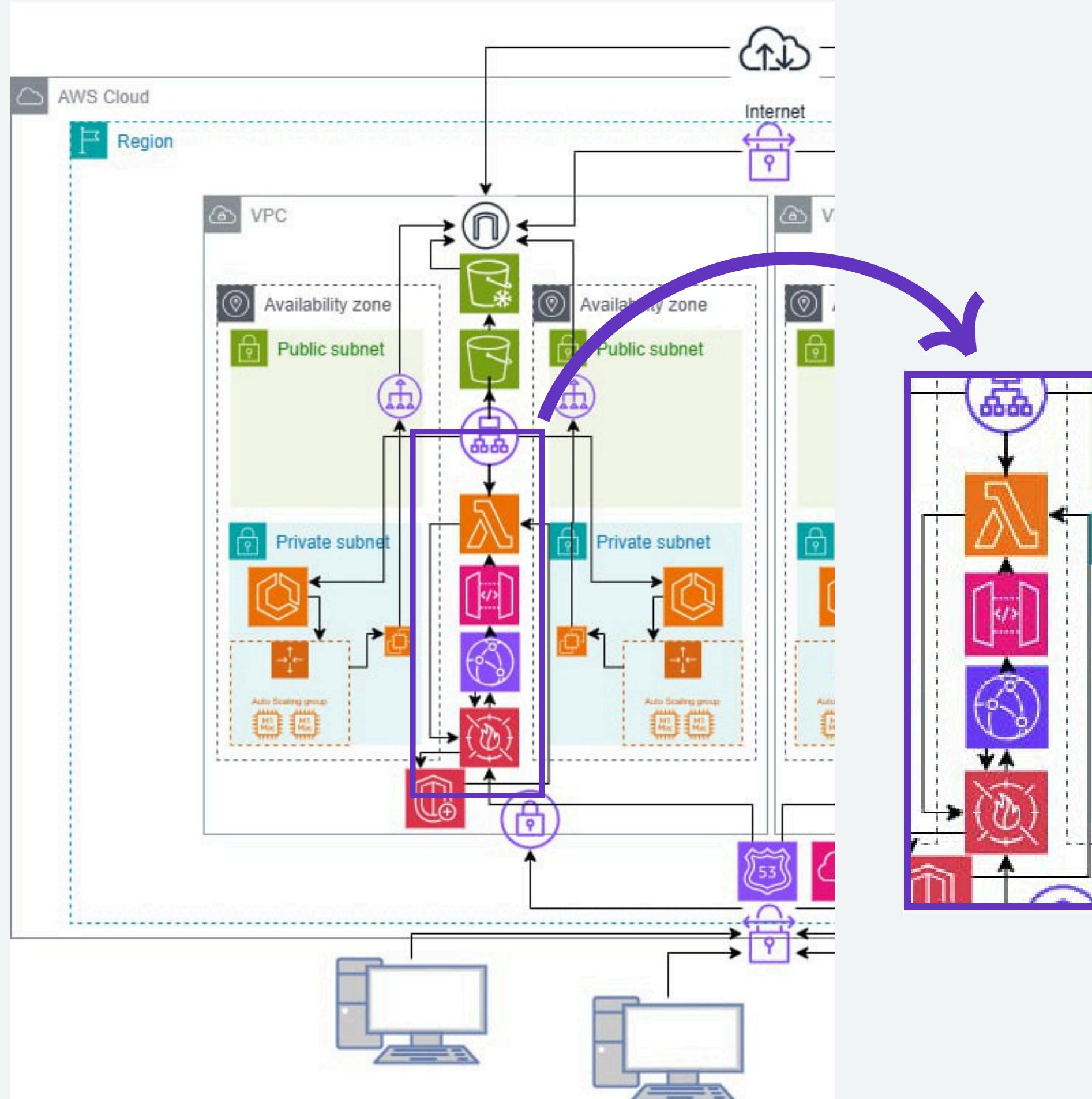
- API Gateway: EC2-Proxy-API  
arn:aws:execute-api:us-east-1:718213540041:ne9uriic7k/\*/POST/auth  
API endpoint: <https://ne9uriic7k.execute-api.us-east-1.amazonaws.com/prod/auth>
- API Gateway: EC2-Proxy-API  
arn:aws:execute-api:us-east-1:718213540041:ne9uriic7k/\*/GET/reviews  
API endpoint: <https://ne9uriic7k.execute-api.us-east-1.amazonaws.com/prod/reviews>
- API Gateway: EC2-Proxy-API  
arn:aws:execute-api:us-east-1:718213540041:ne9uriic7k/\*/POST/notify  
API endpoint: <https://ne9uriic7k.execute-api.us-east-1.amazonaws.com/prod/notify>

Deploy into prod Stage

Triggers Created

# Architecture Diagram

CHIA YUE SHENG 2204673



## When a user requests dynamic content

**Content is cached:** CloudFront instantly serves the response without contacting the backend

**Content is not cached:** CloudFront forwards the request to the origin

**Lightweight dynamic content:** (Login, Notif, Review)  
Handled by Lambda functions via API Gateway.

**Heavy dynamic content:**  
Remains on EC2 instances behind an Application Load Balancer (ALB).

Problem Statement 6:

♥ SECURITY AGAINST BOTS & DDOS ATTACKS IS VULNERABLE.

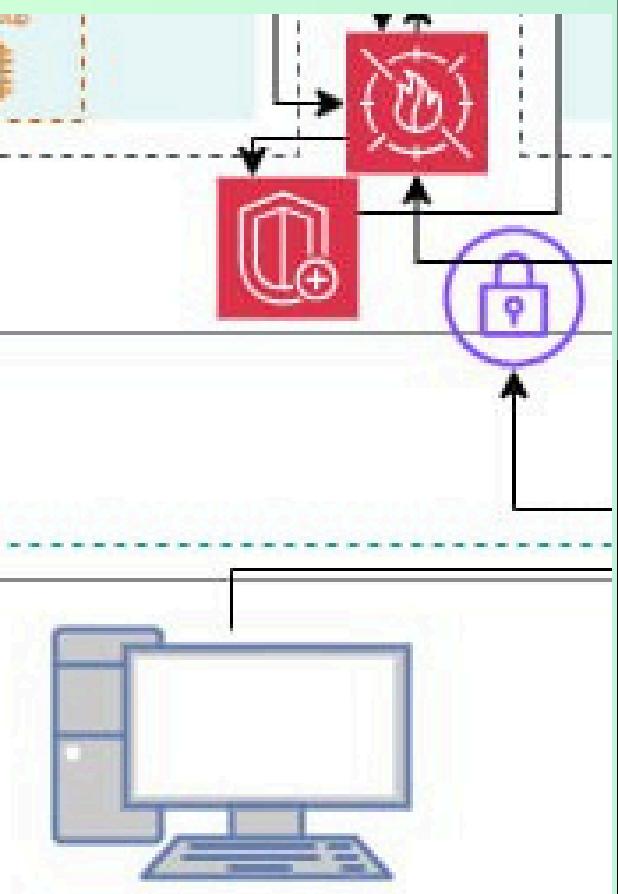


WAF



Shield

Architecture diagram



## AWS WAF configuration steps



First, tap on the WAF & Shield and Web ACLs. Then, proceed the steps from the picture at the left side. Finally, tap create ACLs.

Step 1

Describe web ACL and  
associate it to AWS resources

Step 2

**Add rules and rule groups**

Step 3

**Set rule priority**

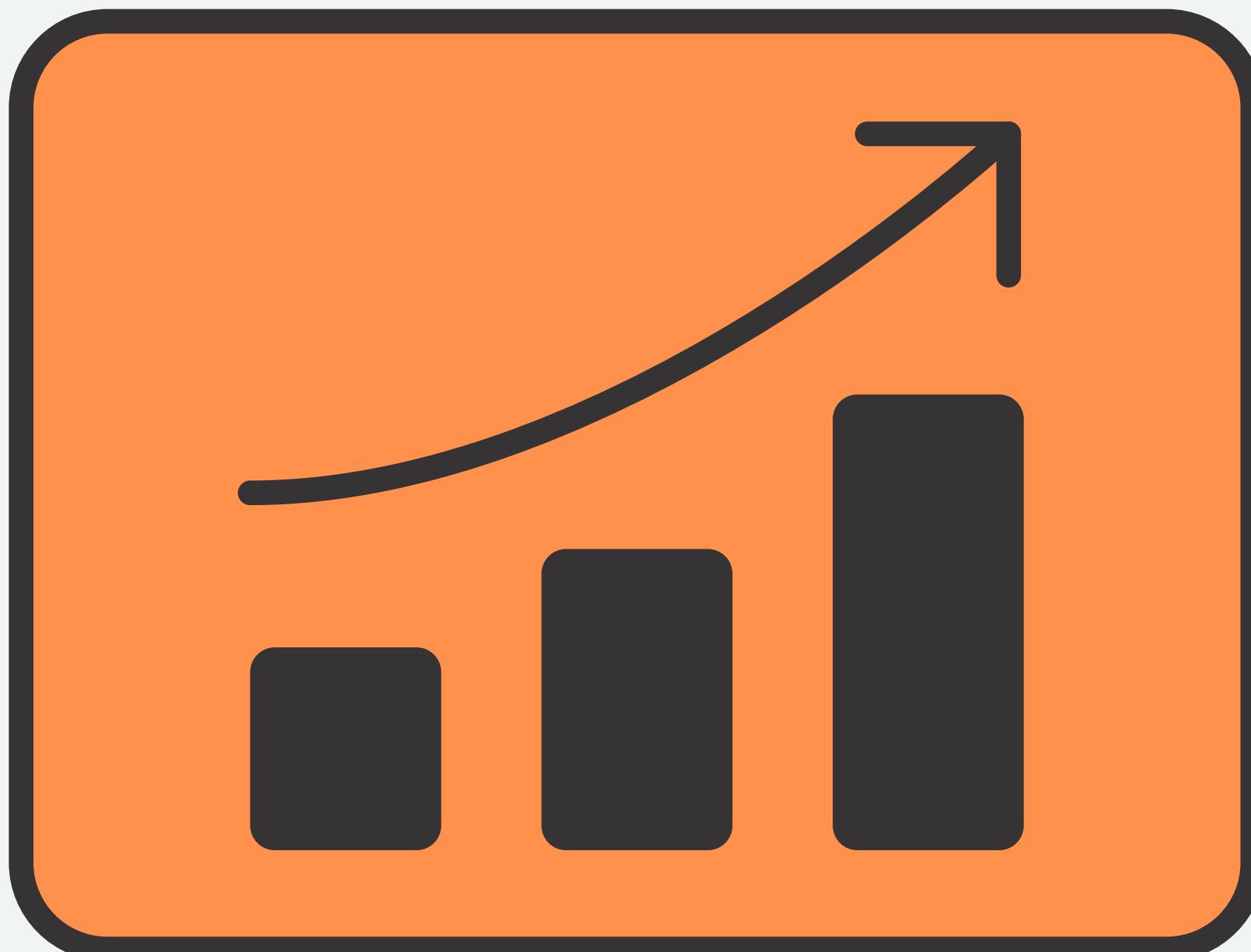
Step 4

**Configure metrics**

Step 5

**Review and create web ACL**

# AWS and other cloud comparison

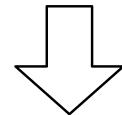


Feature	AWS	Alibaba Cloud
Global Availability	Very High	Strong in Asia only
Service Maturity	Most mature	Improving
Documentation and Support	Excellent	Limited for global users
Compliance and Certifications	Global compliance	China/Asia focused
Developer Ecosystem	Huge and active	Smaller outside China
AI/ML Services	Industry-leading	Good, but less adopted
Pricing	Competitive but complex	Often cheaper in Asia



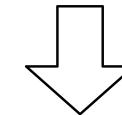
# Timeline

6 MONTHS  
Q1



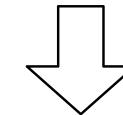
Backend  
with  
Route 53

6 MONTHS  
Q2



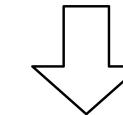
Database

6 MONTHS  
Q3



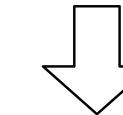
API, LAMBDA

6 MONTHS  
Q4



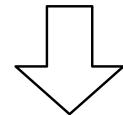
REAL TIME  
CHATBOT

6 MONTHS  
Q5



BACK UP

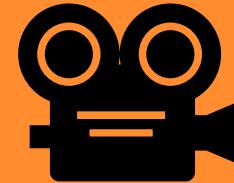
1 YEAR



Advanced  
enhancements

# FUTURE WORK

## BACKEND & DATABASE IMPLEMENTATION



- Serverless architecture (build modular and event-driven API endpoints)
- AWS Lambda (handle backend logic execution)
- AWS AppSync (facilitate real-time communication)
- Amazon RDS with PostgreSQL (ACID compliance and relational integrity)

## ADVANCED ENHANCEMENTS

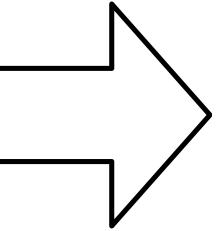


- Lambda@Edge (run code closer to users)
- Amazon ElastiCache (Redis) - (low-latency caching)
- Aurora Global Database (faster read access)
- Amazon Lex (virtual assistant)
- Amazon Personalize (build personalized systems)
- Amazon Kinesis (up-to-date)

---

# **CONCLUSION**

---



# PROBLEM STATEMENTS

## Problem Statement 1:

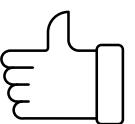


95%



👎 Previously – Static Website

limits our ability to deliver personalized user experiences, real-time updates, and data-driven features



Roadmap toward a fully functional platform in the coming





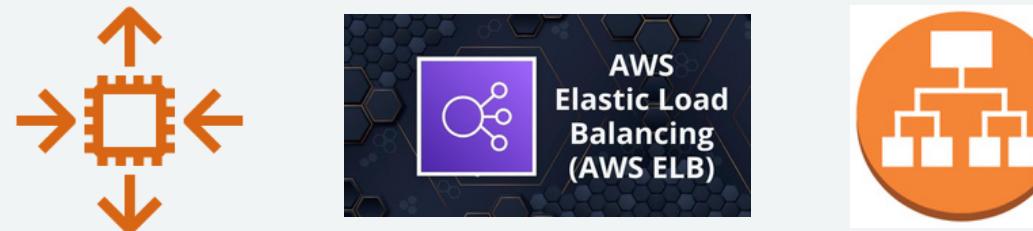
# AWS SERVICES' Solutions



## Problem Statement 2:

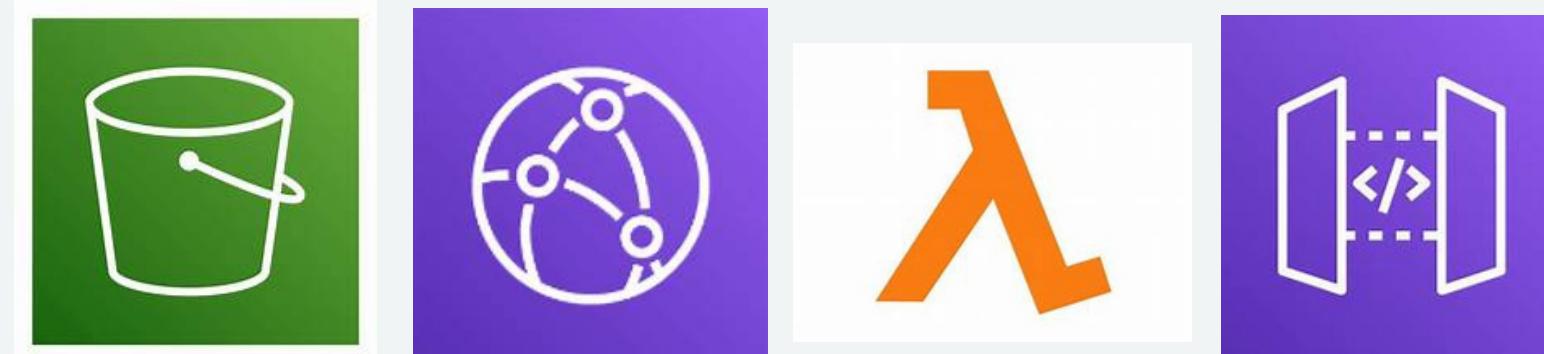
- ♥ HIGH TRAFFIC & SCALABILITY ISSUES.

Sales/Promotional Campaigns → Slow Loading Time



## Problem Statement 3:

- ♥ WEBSITE LOADS SLOWLY DUE TO LARGE STATIC FILES.



## Problem Statement 4:

- ♥ increased load on the EC2 server causes delays in executing tasks such as processing orders, updating stock levels, and managing user sessions



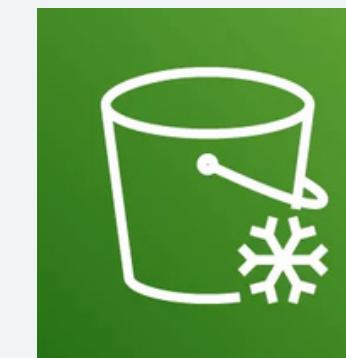


# AWS SERVICES' Solutions



Problem Statement 5:

- ♥ COSTLY STORAGE FOR INFREQUENTLY ACCESSED MEDIA.



Problem Statement 6:

- ♥ SECURITY AGAINST BOTS & DDOS ATTACKS IS VULNERABLE.



WAF



Shield

Problem Statement 7:

- ♥ NO ALERTING OR VISIBILITY INTO SERVER ISSUES.



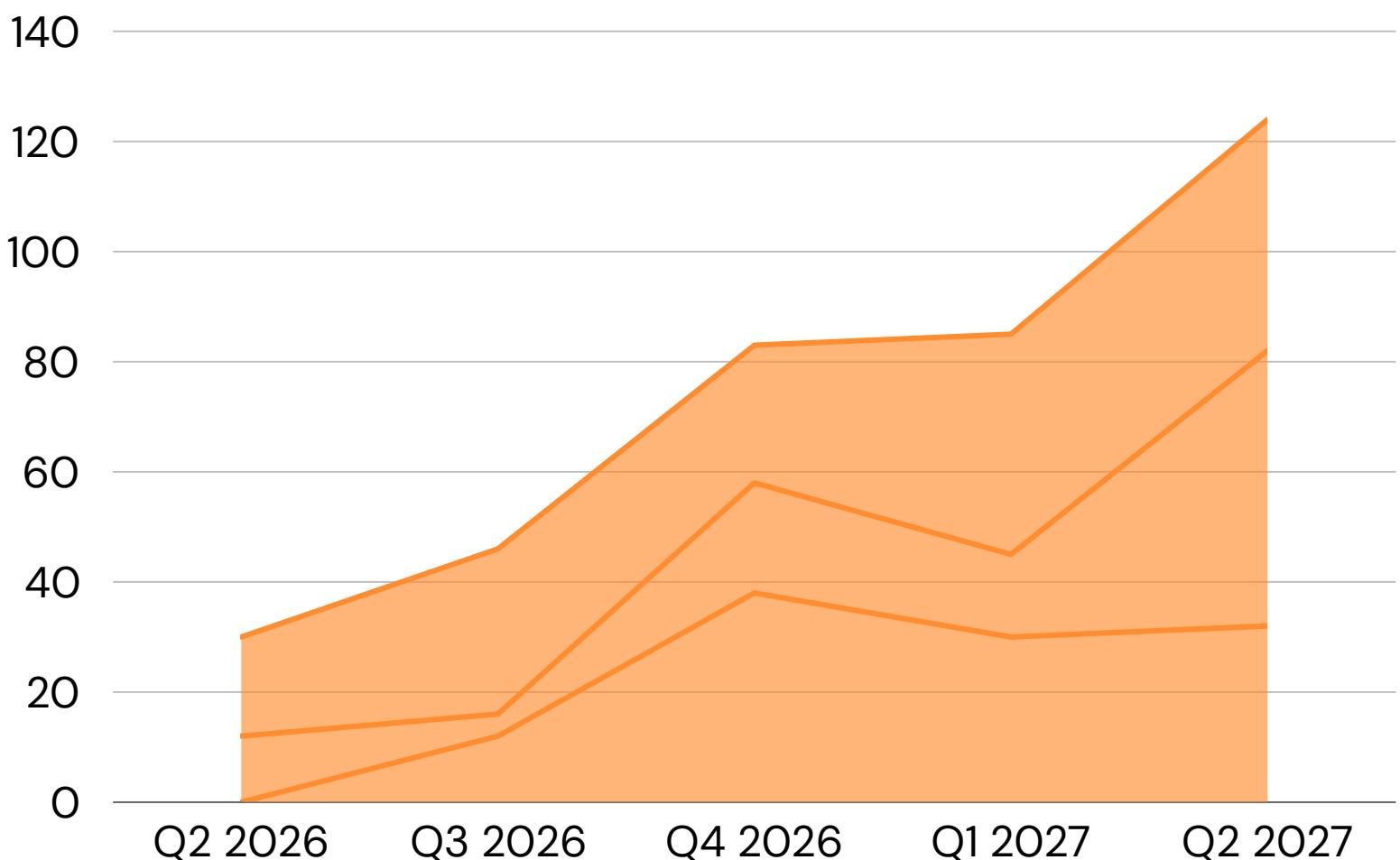
# CONCLUSION

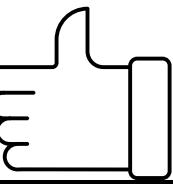


## PREDICTED GROWTH FOR GIFTOS

- Successfully solved all 7 problems with scalable & secure AWS architecture
- Hosted on EC2 with Auto Scaling & Load Balancer for high availability
- Include transitioned from static site to dynamic architecture
- CloudWatch enables real-time monitoring & resource optimization
- S3 + CloudFront ensure fast global delivery of static content
- Integrated Lambda + API Gateway for dynamic backend (e.g., auth, notifications)
- Future-ready: database, real-time analytics, and security (WAF, Shield, S3 Glacier)

**Ready for future growth and advanced features**





# THANK YOU



AWS

YOUR BEST CHOICE

