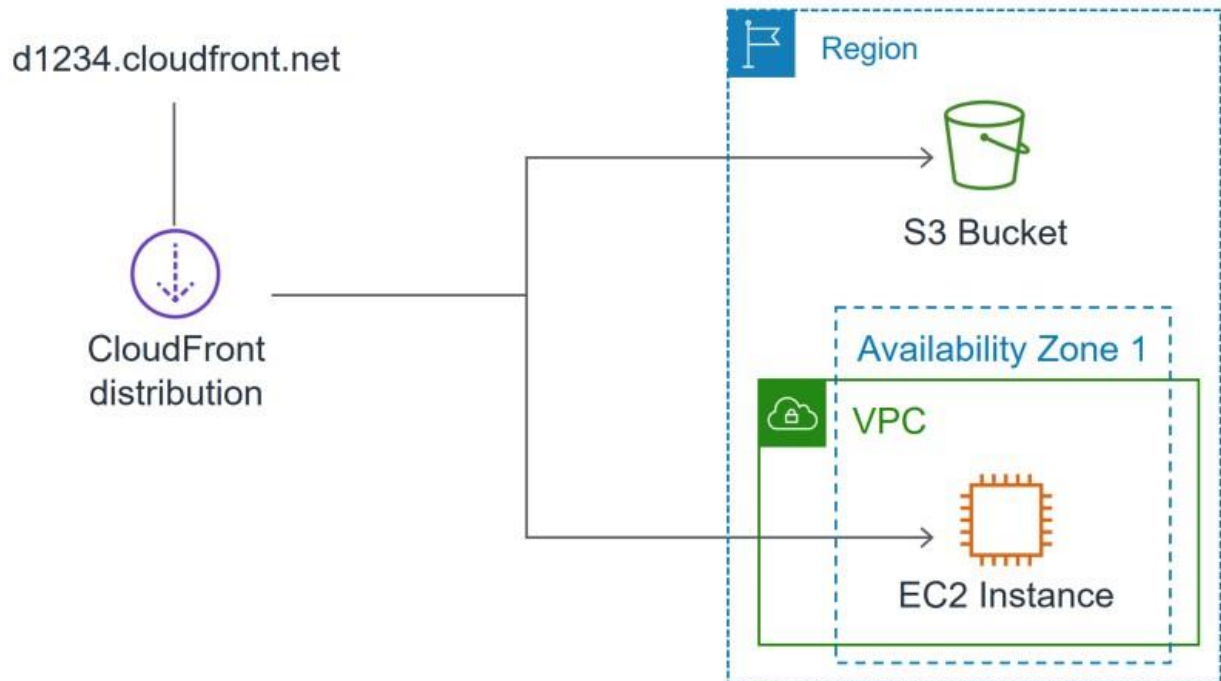


Accelerate Content using Edge Computing

In this experiment, you will learn how to set up a CloudFront Edge Computing distribution to front a simple web application with static and dynamic content hosted on Amazon S3 and on an Amazon EC2 instance respectively, as per the below diagram. You will learn how to test it, and check what are the special headers sent by CloudFront. Finally, you will invalidate the Edge Cached content and configure graceful Edge Computing Content failures using custom error pages.



Create Edge Origins

In this section, you will create both S3 and EC2 Edge Cache Origins using a provided CloudFormation template.

Go to CloudFormation console in North Virginia **us-east-1**.

You can use this URL

<https://console.aws.amazon.com/cloudformation/home?region=us-east-1#>

Or use the service search as we've shown in earlier experiments with **CloudFormation**

Stacks (10)



Delete

Update

Stack actions ▼

Create stack ▲

Q Filter by stack name

With new resources (standard)

With existing resources (import resources)

Stack name

Status

Created time ▼

Description

1. Select **Create stack with new resources** ('Create stack' > 'With new resources (standard)').

Create stack

Prerequisite - Prepare template

Prepare template

Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready☐ Use a sample template☐ Create template in Designer

Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL☒ Upload a template file

Upload a template file

Choose file

EdgeCacheOrigin.yaml

JSON or YAML formatted file

S3 URL: <https://s3-external-1.amazonaws.com/cf-templates-1pljupgill286-us-east-1/202129224z-EdgeCacheOrigin.yaml>

[View in Designer](#)

Cancel

Next

2. Select **Template is ready** and **Upload a template file** options, then choose and upload the following template file: **EdgeCacheOrigin.yaml**, which is available in the Course Github experiments folder
3. Select **Next**

Specify stack details

Stack name

Stack name

edge-cache-origin-student04

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

No parameters

There are no parameters defined in your template

Cancel

Previous

Next

4. Enter a name for your stack as **edge-cache-origin-studentXX** and select **Next**

Configure stack options

Tags

You can specify tags (key-value pairs) to apply to resources in your stack. You can add up to 50 unique tags for each stack. [Learn more](#)

project

edge-cache-origin

Remove

Add tag

5. Enter **project** for our tag key name and **edge-cache-origin** for our tag key value, leave the rest of the configurations as default and select **Next**
6. Click on the **Estimate cost** link to view the Pricing Calculator in a new browser tab

SIMPLE MONTHLY CALCULATOR

Need Help? [Watch the Videos](#) or [Read How AWS Pricing W](#)

Simple Monthly Calculator deprecation update: We appreciate your continuous feedback regarding the [AWS Pricing Calculator](#). The Simple Monthly Calculator's features requested from our customers are available in the AWS Pricing Calculator. We will continue to add services to the AWS Pricing Calculator. If you have any feedback, contact us by using the [Feedback](#) link in the AWS Pricing Calculator.

☐ **FREE TIER:** New Customers get free usage tier for first 12 months


Services

Estimate of your Monthly Bill (\$ 8.50)

Choose region:

US East (N. Virginia)



Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

 Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances.

[Clear Form](#)

Newer versions of the EC2 calculators are available: [Amazon EC2](#), [EC2 Dedicated Host](#), [Elastic Graphics](#), [Elastic IP](#)

Compute: Amazon EC2 Instances:

	Description	Instances	Usage	Type	Billing Option	Monthly Cost
	Instance	1	24 Hours/Day	Linux on t2.micro	On-Demand (No Cor)	\$ 8.50
	Add New Row					

7. Return to the CloudFormation browser tab and select **Create stack**.

CloudFormation > Stacks > edge-cache-origin-student04

Stacks (6)

Filter by stack name

Active View nested

1

edge-cache-origin-student04
2021-10-19 07:35:17 UTC-0500
CREATE_IN_PROGRESS

EdgeloTBootcamp
2021-10-16 20:39:33 UTC-0500
CREATE_COMPLETE

edge-cache-origin-student04

Delete

Stack info Events Resources Outputs Parameters Template Change

Events (1)

Search events

Timestamp	Logical ID	Status
2021-10-19 07:35:17 UTC-0500	edge-cache-origin-student04	CREATE_IN_PROGRESS

8. Wait for the stack status to show **CREATE_COMPLETE**, and remember you have to click the refresh button to update status.

CloudFrontLab

Delete Update Stack actions

Stack info Events Resources Outputs Parameters Template Change sets

Overview

Refresh

Stack ID	Description
arn:aws:cloudformation:us-east-1:376444050498:stack/CloudFrontLab/7da35100-a9e9-11e9-8a38-0abb1914a70	Nodejs webserver on EC2 & S3 bucket
Status	Status reason
CREATE_COMPLETE	-
Root stack	Parent stack
-	-

9. Viewing the Events Tab we note details of our CloudFormation stack including the web server instance, S3 bucket for object storage (web content) and security group information. This information can be useful for debugging in case of any errors.

edge-cache-origin-student04

Delete

Update

Stack actions ▼

Create stack ▼

Stack info

Events

Resources

Outputs

Parameters

Template

Change sets

Events (11)

Q

Search events

Timestamp	Logical ID	Status	Status reason
2021-10-19 07:36:06 UTC-0500	edge-cache-origin-student04	✔ CREATE_COMPLETE	-
2021-10-19 07:36:04 UTC-0500	Instance	✔ CREATE_COMPLETE	-
2021-10-19 07:35:43 UTC-0500	S3Bucket	✔ CREATE_COMPLETE	-
2021-10-19 07:35:32 UTC-0500	Instance	ⓘ CREATE_IN_PROGRESS	Resource creation Initiated

10. View the Outputs Tab and note the DNS name of your EC2 web server as well as the S3 bucket name. During the stack launch process, you can check the progress via Events Tab and get more details for debugging in case of any errors.

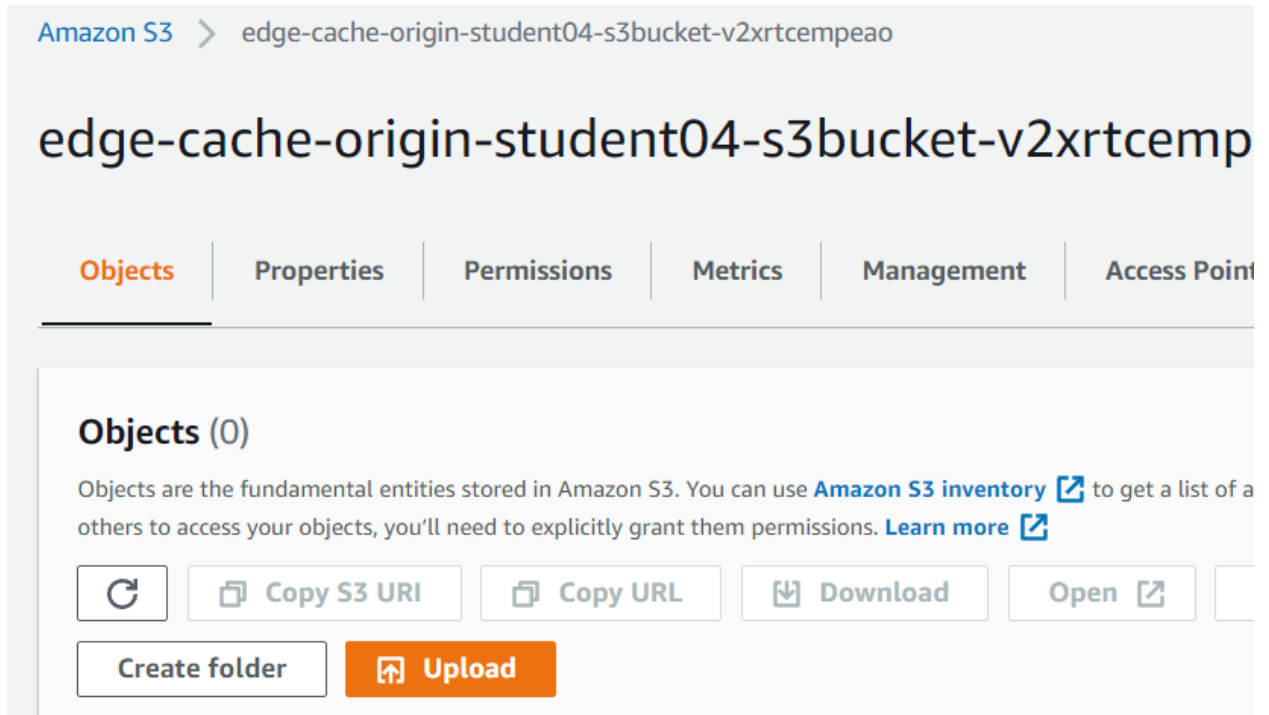
edge-cache-origin-student04					Delete	Update	Stack acti
Stack info	Events	Resources	Outputs	Parameters	Template	Change sets	
Outputs (2)							
<input type="text" value="Search outputs"/>							
Key ▲	Value ▼	Description					
BucketName	edge-cache-origin-student04-s3bucket-v2xrtcempeao	Name of the S3 Edge Cache Origin					
EC2DNSName	ec2-54-91-104-204.compute-1.amazonaws.com	The domain name of the EC2 Edge Cache Origin					

Create Edge Origin Content

1. Create an **index.html** file on your computer using a text editor of your choice and paste the following HTML content in it. This HTML calls the dynamic content using an iframe tag. In fact, when a user makes a request for index.html, the browser sends a subsequent request to /api. Alternatively the Edge Origin content **index.html** is also available in the experiments folder of the **course GitHub repository**.

```
<!DOCTYPE html>
<html lang="en">
  <body>
    <table border="1" width="100%">
      <thead>
        <tr><td><h1>Edge Computing Experiment</h1></td></tr>
      </thead>
      <tfoot>
        <tr><td>Edge Services - Origin Content</td></tr>
      </tfoot>
      <tbody>
        <tr><td>Response sent by Edge API</td></tr>
      </tbody>
      <tbody>
        <tr><td> <iframe src='/api' style="width:100%;
height:100%;"></iframe></td></tr>
      </tbody>
    </table>
  </body>
</html>
```

2. Note the S3 Edge Content Origin Bucket name in the Outputs tab for CloudFormation stack we created. It should be similar to **edge-cache-origin-student04-s3bucket-v2xrtcempeao**
3. Browse to the S3 console through the Service Search dialog or via the following deep link <https://us-east-1.console.aws.amazon.com/s3>








4. Select **Upload**
5. Select **Add files**
6. Browse to the **index.html** that you created or downloaded for our Edge Origin Content and select that file for upload.
7. Leave the remaining options for permissions, location, and properties as default and select **Upload**


Edge Object Storage Security

When you try to request `index.html` using the S3 provided Object URL, the access will be denied since it is not configured as public object.


Objects (1)

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

  Copy S3 URI  Copy URL  Download  Open

Create folder  Upload





 Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	 index.html	html	October 19, 2021, 07:55:40 (UTC-05:00)

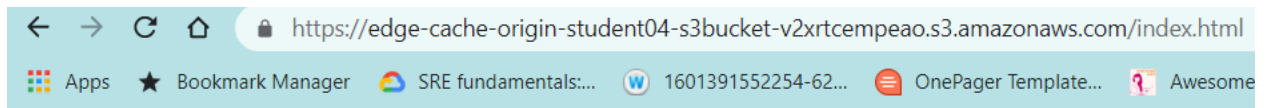
1. Select the **index.html** hyperlinked name to view the file information

Properties Permissions Versions

Object overview

Owner	student4	S3 URI	 s3://edge-cache-origin-student04-s3bucket-v2xrtcmpeao/index.html
AWS Region	US East (N. Virginia) us-east-1	Amazon Resource Name (ARN)	 arn:aws:s3:::edge-cache-origin-student04-s3bucket-v2xrtcmpeao/index.html
Last modified	October 19, 2021, 07:55:40 (UTC-05:00)	Entity tag (Etag)	 8e8e708db9b37c0d292ae5195e3deb5a
Size	482.0 B	Object URL	 https://edge-cache-origin-student04-s3bucket-v2xrtcmpeao.s3.amazonaws.com/index.html
Type	html		

2. Select the **Object URL** from the Object overview in Properties tab to open the index.html that we created and uploaded in a web browser



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>C5SD55NSGH3Y39WT</RequestId>
  <HostId>A6nwj9aC9/ftDwv/ARpHSWIVFIA5QHwa0Mk5n+ZlgrahqH1SK70gX0Yy+J0bVSpKnuiIfnKKIbM=</HostId>
</Error>
```

3. Notice that we are presented with an error page not the index.html Edge Content that we created. We'll correct that security issue later in this experiment

Edge Computing Serverless Lambda

The CloudFormation template has deployed a Node.Js based application that listens to HTTP requests on port 80 of the EC2 instance. Upon receiving a request, the application will send back a JSON response that includes the headers received in the request. It will also inspect the query string info and return some data from the webserver based on the query string value. The application code is below for your reference:

```
const express = require('express')
const app = express()

app.get('/api', function (req, res) {
  console.log(JSON.stringify(req.headers))
  if (req.query.info) {
    require('child_process').exec('cat ' + req.query.info,
      function (err, data) {
        res.send(new Date().toISOString() + '\n' +
          JSON.stringify(req.headers) + '\n'+data)
      });
  } else {
    res.send(new Date().toISOString() + '\n' +
      JSON.stringify(req.headers))
  }
});

app.listen(8080, function () {
  console.log('api is up!')
})
```

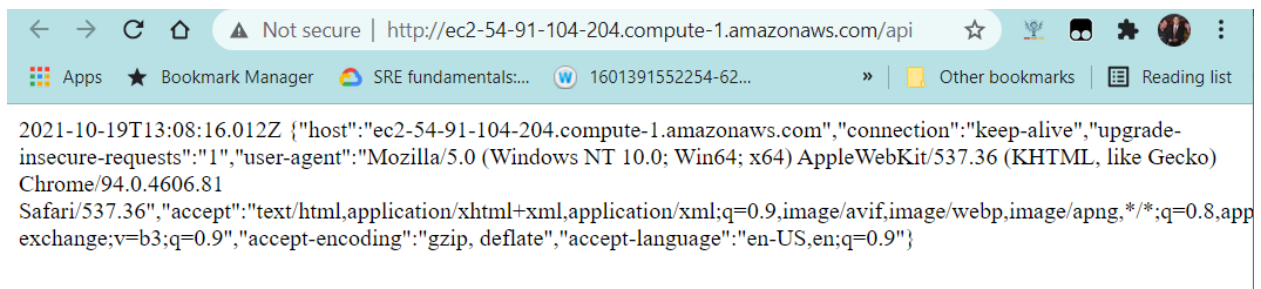
Make sure the application work by entering the following in your browser: <http://EC2-DNS-name/api>. The EC2DNSName is listed in the outputs for your CloudFormation

stack (below the name of the S3 bucket we just uploaded our index.html into). It should look similar to `ec2-54-91-104-204.compute-1.amazonaws.com`

The URL should be similar to

<http://ec2-54-91-104-204.compute-1.amazonaws.com/api>

You should see a response like the example below.



Create Edge Computing Distribution

1. Browse to the CloudFront console via the Service Search dialog or use the following service deep link <https://us-east-1.console.aws.amazon.com/cloudfront>

Get started with CloudFront

Enable accelerated, reliable and secure content delivery for Amazon S3 buckets, Application Load Balancers, Amazon API Gateway APIs, and more in 5 minutes or less.

[Create a CloudFront distribution](#)

2. Select **Create a CloudFront distribution**.

Select a delivery method for your content.

Web

Create a web distribution if you want to:

- Speed up distribution of static and dynamic content, for example, .html, .css, .php, and graphics files.
- Distribute media files using HTTP or HTTPS.
- Add, update, or delete objects, and submit data from web forms.
- Use live streaming to stream an event in real time.

You store your files in an origin - either an Amazon S3 bucket or a web server. After you create the distribution, you can add more origins to the distribution.

[Get Started](#)

3. Configure the default origin to the previously created S3 bucket and grant CloudFront the permissions to the bucket using Origin Access Identity settings:

- Restrict Bucket Access: Yes
- Origin Access Identity: Create a new Identity
- Grant Permissions on Bucket: Yes, Update Bucket policy
- Origin Domain Name: Select the S3 bucket, similar to **edge-cache-origin-student04-s3bucket-v2xrtcempeao**, that we created in our CloudFormation by clicking in the input box and it should be automatically visible in the list provided. This will automatically populate the Name field, as well. Leave the remaining defaults in the Origin section for the new Edge Distribution

Create distribution

Origin

Origin domain
Choose an AWS origin, or enter your origin's domain name.

Q edge-cache-origin-student04-s3bucket-v2xrtcempeao.s3.us-east-1.amazonaws.com X

Origin path - optional [Info](#)
Enter a URL path to append to the origin domain name for origin requests.

Enter the origin path

Name
Enter a name for this origin.

edge-cache-origin-student04-s3bucket-v2xrtcempeao.s3.us-east-1.amazonaws.com

S3 bucket access [Info](#)
Use a CloudFront origin access identity (OAI) to access the S3 bucket.

☒ Don't use OAI (bucket must allow public access)

☐ Yes use OAI (bucket can restrict access to only CloudFront)

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

Add header

Enable Origin Shield [Info](#)
Origin Shield is an additional caching layer that can help reduce the load on your origin and help protect its availability.

☒ No

☐ Yes

► Additional settings

4. Configure the Default Cache Behavior as follows:

Default cache behavior

Path pattern [Info](#)

Default (*)

Compress objects automatically [Info](#)

☐ No

☒ Yes

Viewer

Viewer protocol policy

☐ HTTP and HTTPS

☒ Redirect HTTP to HTTPS

☐ HTTPS only

Allowed HTTP methods

☒ GET, HEAD

☐ GET, HEAD, OPTIONS

☐ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.

☒ No

☐ Yes

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

☐ Cache policy and origin request policy (recommended)

☒ Legacy cache settings

Headers

Choose which headers to include in the cache key.

None

Query strings

Choose which query strings to include in the cache key.

None

Cookies

Choose which cookies to include in the cache key.

None

Object caching

☐ Use origin cache headers

☒ Customize

Minimum TTL

Minimum time to live in seconds.

0

Maximum TTL

Maximum time to live in seconds.

86400

Default TTL

Default time to live in seconds.

86400

- Viewer Protocol Policy: Redirect HTTP to HTTPS
- Cache and origin request settings: Use legacy cache settings
- Object Caching: Customize
- Minimum TTL: 86400

Settings

Price class [Info](#)

Choose the price class associated with the maximum price that you want to pay.

☐ Use all edge locations (best performance)
☒ Use only North America and Europe
☐ Use North America, Europe, Asia, Middle East, and Africa

AWS WAF web ACL - optional

Choose the web ACL in AWS WAF to associate with this distribution.

Choose web ACL

Alternate domain name (CNAME) - optional

Add the custom domain names that you use in URLs for the files served by this distribution.

Add item

To add a list of alternative domain names, use the bulk editor.

Custom SSL certificate - optional

Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

Choose certificate

Request certificate

Supported HTTP versions

Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.

☒ HTTP/2

Default root object - optional

The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.

index.html

Standard logging

Get logs of viewer requests delivered to an Amazon S3 bucket.

☒ Off
☐ On

IPv6

☐ Off
☒ On

Description - optional

Cancel

Create distribution

- In the Distributions Settings section, configure **Default root object** to **index.html** and **Price class** to **Use only North America and Europe**, leave the rest to defaults.

Default Root Object

index.html



In this lab, you will be using a domain name provided by CloudFront, however, if you want to use your own domain name, you can configure it with Alternate Domain Names (CNAMEs) section.

- Select **Create distribution**. CloudFront will start creating the Edge Computing Cache distribution and normally it takes 5 to 10 minutes to fully propagate. The status of the distribution will be In Progress. To check the status, you can click on the Distribution menu on left pane.

CloudFront Distributions

Create Distribution		Distribution Settings	Delete	Enable	Disable		
Viewing: Any Delivery Method		Any State					<< < Viewing
Delivery Method	ID	Domain Name	Comment	Origin	CNAMEs	Status	State
Web	E1LSB008BU16W1	d1wj00cdxoism.cloudfront.net	-	cloudfrontlab-s3bucket-17xkmsdvqboj9.s3.amazonaws.com	-	In Progress	Enabled

Add Edge Cache API Origin

1. In the distributions console, click on your distribution ID, then select the Origins Tab to view the Origins and Origin Groups view.

CloudFront > Distributions > E19Y8F73FLETMC

E19Y8F73FLETMC

General | **Origins** | Behaviors | Error pages | Geographic restrictions | Invalidations | Tags

Origins

Edit Delete **Create origin**

Filter origins by property or value

< 1 > ⚙

Origin name
edge-cache-origin-student04-s3bucket-v2xrtcmepao.s3.us-east-1.amazonaws.com

Origin groups

Edit Delete **Create origin group**

Filter origin groups by property or value

< 1 > ⚙

Origin group name

2. Enter the EC2DNSName from our CloudFormation Outputs that will be similar to ec2-54-91-104-204.compute-1.amazonaws.com. Increase the keep alive timeout to 60 seconds. Please note that although we want to serve content on HTTPS to users, we want to keep HTTP connection the origin to reduce the TLS overhead on the origin. This is configured by setting the Origin Protocol Policy to HTTP

CloudFront > Distributions > E19Y8F73FLETMC > Create origin

Create origin

Settings

Origin domain
Choose an AWS origin, or enter your origin's domain name.

Protocol [Info](#)

☒ HTTP only
☐ HTTPS only
☐ Match viewer

3. Select **Create origin** button.
- 4.
5. Select the Behaviours Tab

General

Origins and Origin Groups

Behaviors

Error Pages

Restrictions

Invalidations

Tags

CloudFront compares a request for an object with the path patterns in your cache behaviors based on the order of the cache behaviors in your distribution. Arrange cache behaviors in the order in which you want CloudFront to evaluate them.

Create Behavior

Edit

Delete

Change Precedence:

Move Up

Move Down

Save

	Precedence ▾	Path Pattern	Origin or Origin Group	Viewer Protocol Policy
<input type="checkbox"/>	0	Default (*)	S3-cloudfrontlab-s3bucket	Redirect HTTP to HTTPS

6. Select **Create behavior**

Create behavior

Settings

Path pattern [Info](#)

Origin and origin groups

Compress objects automatically [Info](#)

- ☐ No
☒ Yes

Viewer

Viewer protocol policy

- ☐ HTTP and HTTPS
☒ Redirect HTTP to HTTPS
☐ HTTPS only

Allowed HTTP methods

- ☒ GET, HEAD
☐ GET, HEAD, OPTIONS
☐ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

7. Update

- Path pattern - /api
- Compress objects – Yes
- Viewer – Redirect HTTP to HTTPS

8. Update the Cache Key and Origin Request Settings as noted below selecting **Legacy cache settings** option

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

- ☐ Cache policy and origin request policy (recommended)
- ☒ Legacy cache settings

Headers

Choose which headers to include in the cache key.

All



You are choosing to forward all headers to the origin, which means CloudFront doesn't cache objects with this cache behavior. It sends every request to the origin.

Query strings

Choose which query strings to include in the cache key.

All

Cookies

Choose which cookies to include in the cache key.

All

Object caching

- Headers – All
- Query Strings – All
- Cookies – All

9. Select **Create behavior**

Test The Edge Application On CloudFront


1. Note that when we uploaded our index.html we never made it public so we received an error on viewing. We'll verify that is still occurring. View our CloudFront Edge Cache Distribution

CloudFront > Distributions > E19Y8F73FLETMC

E19Y8F73FLETMC

General | Origins | Behaviors | Error pages | Geographic restrictions

Details

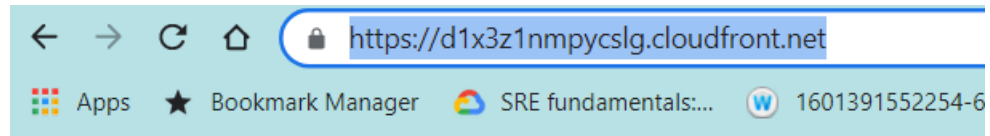
Distribution domain name
 d1x3z1nmpycslg.cloudfront.net

Settings

Description
-

Price class
Use only North America and Europe

2. Check the **Distribution domain name** that CloudFront has associated to your distribution in the General tab. It should be similar to <https://d1x3z1nmpycslg.cloudfront.net/> as seen in the view.
3. View the CloudFront distribution we've created in a browser. CloudFront distributions can be ready to be used locally even if the status is still in progress, since the status will change to deployed when the propagation has reached all 220+ edge locations.



This XML file does not appear to have any style information associated with it

```
<?xml version="1.0" encoding="UTF-8" ?>
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>E7MR6F0CKRE0N29D</RequestId>
  <HostId>22p6y42CpsQk2CqaCgq7hXkQ59i/j1LZwReKhaesAngkxU4tITYWftdgi</HostId>
</Error>
```

4. Browse to the S3 Console via the Service Search dialog or the following deep link. <https://console.aws.amazon.com/s3>

Amazon S3 > edge-cache-origin-student04-s3bucket-v2xrtcempeao

edge-cache-origin-student04-s3bucket-v2xrtcempeao [Info](#)

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1)

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.

Copy S3 URI Copy URL Download Open Delete


Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	index.html	html	October 19, 2021, 07:55:40 (UTC-05:00)

5. Click on the [index.html](#) hyperlink to view the properties of the file we've uploaded.

index.html [Info](#)

 Copy S3 URI

 Download

Open [↗](#)

Object actions ▼

Properties


Permissions

Versions

Access control list (ACL)

Grant basic read/write permissions to AWS accounts. [Learn more](#) [↗](#)





Edit

Grantee	Object	Object ACL
Object owner (your AWS account)	Read	Read, Write
Canonical ID:  5016ed133ed623f822f165bd457e068142876662		

6. Select the Permissions tab and select **Edit**

Access control list (ACL)

Grant basic read/write permissions to AWS accounts. [Learn more](#) [↗](#)

Grantee	Objects	Object ACL
Object owner (your AWS account) Canonical ID:  5016ed133e d623f822f165bd457e0681428 76662ae649a1d4ec38741d181 542e	<input checked="" type="checkbox"/> Read	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Everyone (public access) Group:  http://acs.amazon aws.com/groups/global/AllUser s	<input checked="" type="checkbox"/>  Read	<input checked="" type="checkbox"/>  Read <input type="checkbox"/> Write

7. Check the box for **Read** on the Objects and Object ACL



When you grant access to the Everyone or Authenticated users group grantees, anyone in the world can access this object.

[Learn more](#)

☒ I understand the effects of these changes on this object.

Access for other AWS accounts

No other AWS accounts associated with the resource.

[Add grantee](#)

Specified objects

Name	Type	Last modified	Size
index.html	html	October 19, 2021, 07:55:40 (UTC-05:00)	482.0 B

Cancel

Save changes

8. Check the box for **I understand the effects of these changes ...** and select **Save changes**
9. Check the **Distribution domain name** that CloudFront has associated to your distribution in the General tab. It should be similar to <https://d1x3z1nmpycslg.cloudfront.net/> as seen in the view.

←

→

↻

🏠

🔒 https://d1x3z1nmpycslg.cloudfront.net/index.html

📱 Apps

★ Bookmark Manager

🌐 SRE fundamentals:...

🔍 1601391552254-62...

📄 OnePage

Edge Computing Experiment

Response sent by API

2021-10-19T14:20:17.594Z {"host":"d1x3z1nmpycslg.cloudfront.net","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7; rv:98.0) Gecko/20100101 Firefox/98.0","accept":"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8","accept-language":"en-US,en;q=0.9","accept-encoding":"gzip","referer":"https://d1x3z1nmpycslg.cloudfront.net/index.html","sec-ch-ua-platform":"macOS","sec-ch-ua-mobile":"?0","sec-ch-ua-platform-version":"","upgrade-insecure-requests":"1","sec-fetch-mode":"navigate","sec-fetch-site":"cross-site","sec-fetch-user":"?1","cloudfront-is-smarttv-viewer":"false","cloudfront-is-desktop-viewer":"true"}

Edge Services - Origin Content

Test the Edge DNS Resolution

1. To test if the distribution is ready to be used locally, you can lookup its CloudFront domain name in command line by nslookup command. Keep in mind that dxxxx.cloudfront.net name is unique for every distribution, this is why need to use your own distribution value for testing. Note how CloudFront returns multiple IPs for each DNS query to increase application resiliency.

```
nslookup d1wj00cdxoism.cloudfront.net
Server:      10.4.4.10
Address:     10.4.4.10#53

Non-authoritative answer:
Name:   d1wj00cdxoism.cloudfront.net
Address: 52.84.225.127
Name:   d1wj00cdxoism.cloudfront.net
Address: 52.84.225.132
Name:   d1wj00cdxoism.cloudfront.net
Address: 52.84.225.137
Name:   d1wj00cdxoism.cloudfront.net
Address: 52.84.225.183
```

2. When the propagation is complete, you can test the webpage on your browser as served by CloudFront using <http://dxxxx.cloudfront.net> . In the webpage, you can see the different headers that CloudFront has forwarded and appended to your API endpoint:

- **cloudfront-forwarded-proto**: Indicates the protocol used by the viewer to connect to CloudFront
- **cloudfront-is-mobile-viewer**: Indicates the viewer's device type
- **cloudfront-viewer-country**: Indicates the viewer's country
- **x-amz-cf-id**: a unique id for this request provided by CloudFront. IF you refresh the webpage, you will see that how the request id is changing. It's useful to log it on your webserver in general. Additionally, this id will be sent back to every viewer request and sent to CloudFront access logs. If you need to debug any issue you can open a support ticket and provide them with the req id.

Also note how CloudFront redirected the request to HTTPS.



3. If you use the developer tools of your favorite web browser, you can check the response headers sent by CloudFront. Three headers are interesting to check:
 - **x-amz-cf-id** which holds the request id assigned by CloudFront.
 - **x-amz-cf-pop** which indicates the CloudFront edge location that served your request. Each edge location is identified by a three-letter code and an arbitrarily assigned number, for example, DFW3. The three-letter code typically corresponds with the International Air Transport Association airport code for an airport near the edge location.

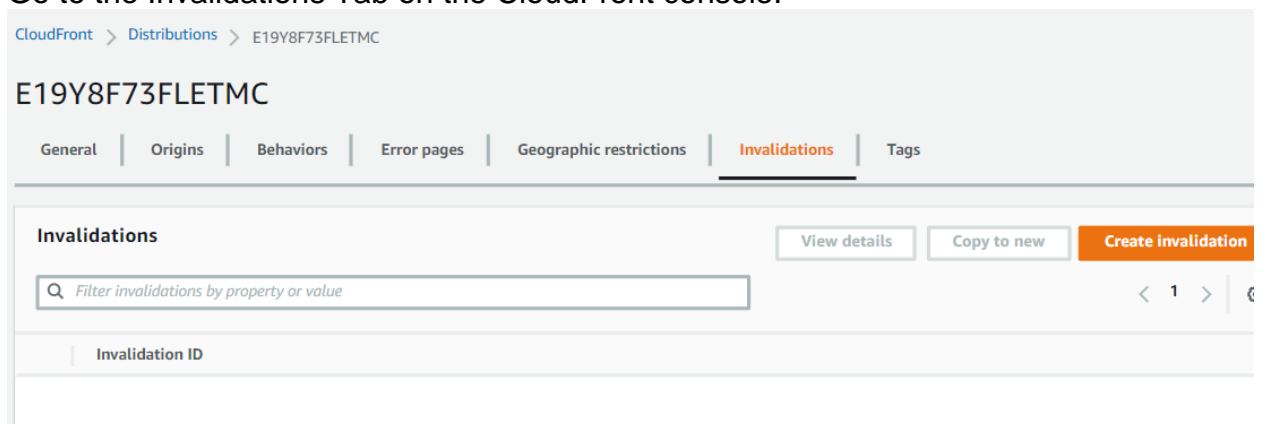
- **x-cache** which indicates whether the request was a cache hit or a cache miss. Normally, for your html file, you will get a 'Hit from CloudFront' value in the subsequent requests, but always 'Miss from CloudFront' for /api request since caching is disabled for this behavior.



Test Edge Cache Invalidations

As you saw previously, the main index.html page is in cache and resulting in a Hit from CloudFront. Suppose that you have to change the HTML file but you can't change the URL to point to the new version, in this case, you need to invalidate the page.

1. Go to the Invalidations Tab on the CloudFront console.



2. Select **Create invalidation**. Create an invalidation for your index.html. You can specify / in the Object paths because we already set index.html as default root object.

Create invalidation


Object paths

Add object paths

Add the path for each object that you want to remove from the CloudFront cache. You can use wildcards (*).

Remove


Add item

 To add a list of object paths, use the [bulk editor](#).


Cancel

Create invalidation

3. Update our invalidation for the root path with the forward slash (/index.html) as noted and select **Create Invalidation**.

 **Upload succeeded**
View details below.

Upload: status

 The information below will no longer be available after you navigate away from this page.

Summary

Destination

s3://edge-cache-origin-student04-s3bucket-v2xrtcempeao

Succeeded

 1 file, 494.0 B (100.00%)

4. Update the **index.html** that we created or downloaded in a text editor and upload that again to the S3 bucket that we created for the experiment. Since we're uploading a new object (even with the same name) we'll have to update the Permissions to make it readable again.

- ▼ Response Headers (497 B)

 - accept-ranges: bytes
 - content-length: 407
 - content-type: text/html
 - date: Wed, 31 Mar 2021 11:13:31 GMT
 - etag: "ff5d81dd54d8156e73d5d789d0652609"
 - last-modified: Wed, 31 Mar 2021 11:12:52 GMT
 - server: AmazonS3
 - via: 1.1 20859c946d4540573244991afc8ba6b1.cloudfront.net (CloudFront)
 - x-amz-cf-id: xdGyf_3U_yM234rxv2NXnJyqEOa27rWbaE0YXJNXWSE3-2NvvFMMMw==
 - x-amz-cf-pop: LHR62-C5
 - x-amz-version-id: bgZwl22bvnr.RYsNWEqWe0Uq3ElVJtQ6
 - x-cache: Miss from cloudfront

-

1. Test a random URL using your CloudFront domain name and you will get a 403 Forbidden response from S3 behind CloudFront because the file does not exist.

By default, CloudFront caches this response for 5 minutes.

← → ↻ 🔒 https://d1wj00cdxoism.cloudfront.net/random

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>BE8E155EBE78989B</RequestId>
  <HostId>
    Oym/PnbkouqSzSCypW84yj8dHYM/8n65OTOZLj/YBOhl2wrNE3iDic/JE3behPj7LR6JNGFuzv0=
  </HostId>
</Error>
```


2. Create an **error.html** file on your computer using a text editor with the below HTML content, and upload it to your S3 bucket like you did earlier for index.html.



```
<html lang="en">
  <body>
    <h1>Edge Cache Error</h1>
    Oops, this is a nice error page!
  </body>
</html>
```

<input type="checkbox"/>	Name ▾	Last modified ▾	Size ▾
<input type="checkbox"/>	 error.html	Jul 19, 2019 2:54:07 PM GMT+0800	109.0 B
<input type="checkbox"/>	 index.html	Jul 19, 2019 1:58:10 PM GMT+0800	463.0 B

3. When uploading update the permissions on the upload or update the security after you've uploaded to make the error.html public



▼ Permissions
Grant public access and access to other AWS accounts.

Access control list (ACL)
Grant basic read/write permissions to other AWS accounts. [Learn more](#) 

 AWS recommends using S3 bucket policies or IAM policies for access control. [Learn more](#) 

Access control list (ACL)
☒ Choose from predefined ACLs
☐ Specify individual ACL permissions

Predefined ACLs
☐ Private (recommended)
Only the object owner will have read and write access.
☒ Grant public-read access
Anyone in the world will be able to access the specified objects. The object owner will have read and write access. [Learn more](#)

 **Granting public-read access is not recommended**
Anyone in the world will be able to access the specified objects. [Learn more](#) 
☒ I understand the risk of granting public-read access to the specified objects.

4. On the CloudFront console, within your distribution, go to the Error Pages tab and click the **Create Custom Error Response** button.
5. Configure the custom error response with the following settings.
 - HTTP Error Code: 403 Forbidden
 - Error Caching Minimum TTL (seconds) : 60
 - Customize Error Response : Yes
 - Response Page Path : /error.html
 - HTTP Response Code: 200 OK

Create custom error response

Error response [Info](#)

HTTP error code
Customize the custom error response when the origin sends this error code.

403: Forbidden ▼

Error caching minimum TTL
Enter the error caching minimum time to live (TTL), in seconds.

60

Customize error response
Send a custom error response instead of the error received from the origin.

☐ No
☒ Yes

Response page path
Enter the path to the custom error response page.

/error.html

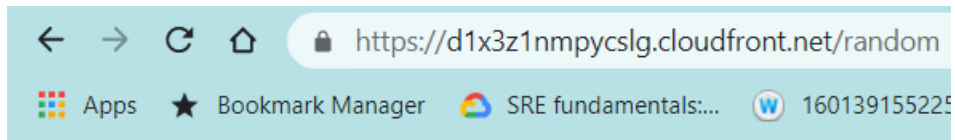
HTTP Response code
Choose the HTTP status code to return to the viewer. CloudFront can return a different status code to the viewer than what it received from the origin.

200: OK ▼

Cancel

Create custom error response

6. Test your custom error page, by requesting a random page from CloudFront. You may need a few minutes to wait for distribution to update and propagate to edge locations. Make sure that you use a different random value from the previous test, otherwise you will get the same cached version if you test within 5 minute.



Edge Cache Error

Oops, this is a nice error page!

Optional – More Custom Error Exploration

Create another custom error page that will be triggered when the origin is not reachable by CloudFront. Use the following settings:

- HTTP Error Code: 504 Gateway Timeout
- Error Caching Minimum TTL (seconds) : 5
- Customize Error Response : Yes
- Response Page Path: /error.html
- HTTP Response Code: 200 OK

Custom Error Response Settings

HTTP Error Code	504: Gateway Timeout	
Error Caching Minimum TTL (seconds)	5	
Customize Error Response	<input checked="" type="radio"/> Yes <input type="radio"/> No	
Response Page Path	/error.html	
HTTP Response Code	200: OK	

Go to the EC2 console and block inbound traffic to the EC2 instance which is hosting the Nodejs api.

Edit inbound rules ✕

Type ⁱ	Protocol ⁱ	Port Range ⁱ	Source ⁱ	Description ⁱ	
HTTP ▾	TCP	80	Custom ▾ 0.0.0.0/0	e.g. SSH for Admin Desktop	✕
HTTP ▾	TCP	80	Custom ▾ :::/0	e.g. SSH for Admin Desktop	✕

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

8 - Test your index.html page on the browser. Wait for a few moments until the API call fails gracefully to the custom error page.

Configure Origin Group

In this section, you will configure an origin group to provide rerouting during a failover event. You can associate an origin group with a cache behavior to have requests routed from a primary origin to a secondary origin for failover.

1. In S3 console, create a new S3 bucket in a different region, for example, us-west-1. Give it a unique name, remember S3 bucket names are globally unique, so add a personalized suffix, such as **cloudfrontlab-s3bucket-secondary-**. Uncheck the “block all public access” box.

Create bucket

1 Name and region

2 Configure options

3 Set permissions

4 Review

Name and region

Bucket name ⓘ

cloudfrontlab-s3bucket-secondary

Region

US West (N. California) ▾

Copy settings from an existing bucket

Select bucket (optional)33 Buckets ▾

✓ Name and region

✓ Configure options

3 Set permissions

4 Review

⚠

Disabling Block all public access may result in this bucket and the objects within becoming public

AWS recommends that you block all public access to your bucket, unless public access is required for specific and verified use cases such as static website hosting.

☒ I acknowledge that the current settings may result in this bucket and the objects within becoming public

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.

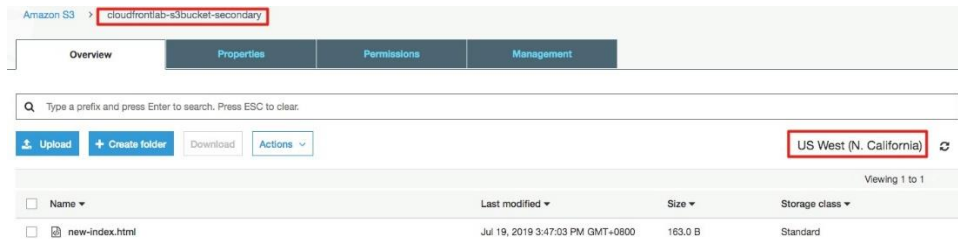
Block public and cross-account access to buckets and objects through any public bucket or access point policies

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

2. Create a new-index.html file on your computer with the below HTML content and upload it to your new S3 bucket in us-west-1 from the S3 console. Make the new-index.html public by selecting the option from Object actions dropdown menu.

```
<html lang="en">
  <body>
    <h1>CloudFront Lab</h1>
    Hi, this is a page from my secondary Origin! We now support
    Origin group and failover!
```

```
</body>
</html>
```



3. Go back to your CloudFront distribution and create a new Origin with the newly created S3 bucket in us-west-1.
- Origin Domain Name: The website hosting Endpoint of Your New S3 bucket (like <http://cloudfrontlab-s3bucket-secondary.s3-website-us-west-1.amazonaws.com>)

Origin Settings

Origin Domain Name	<input type="text" value="s3bucket-secondary.s3.amazonaws.com"/>					
Origin Path	<input type="text"/>					
Origin ID	<input type="text" value="S3-cloudfrontlab-s3bucket-secondary"/>					
Restrict Bucket Access	<input type="radio"/> Yes <input checked="" type="radio"/> No					
Origin Custom Headers	<table><tr><th>Header Name</th><th>Value</th></tr><tr><td><input type="text"/></td><td><input type="text"/></td></tr></table>	Header Name	Value	<input type="text"/>	<input type="text"/>	
Header Name	Value					
<input type="text"/>	<input type="text"/>					

4. Create an Origin Group with a primary and secondary origin. Go to CloudFront Origins and Origin Groups Tab, click Create Origin Group.

General **Origins and Origin Groups** Behaviors Error Pages Restrictions Invalidations Tags

Origins

Create Origin Edit Delete

	Origin Domain Name and Path	Origin ID	Origin Type	Origin Access Identity
<input type="checkbox"/>	cloudfrontlab-s3bucket-secondary.s3-website-us-west-1.amazo	S3-cloudfrontlab-s3b	S3 Origin	-
<input type="checkbox"/>	ec2-34-230-14-155.compute-1.amazonaws.com	Custom-ec2-34-230-	Custom Origin	-
<input type="checkbox"/>	cloudfrontlab-s3bucket-17xkmsdvqboj9.s3.amazonaws.com	S3-cloudfrontlab-s3b	S3 Origin	origin-access-identity/cloudfront/E3T683T5GYHTIX

Origin Groups

Create an origin group to provide rerouting during a failover event. You can associate an origin group with a cache behavior to have requests routed from a primary origin to a secondary origin for failover. You must have two origins for your distribution before you can create an origin group. Please note that with an origin group, you can only use GET, HEAD, and OPTIONS HTTP methods in your cache behavior. [Learn more](#)

Create Origin Group Edit Delete

Origin Group ID	Origins
-----------------	---------

- Use S3-cloudfrontlab-s3bucket as primary origin, and S3-cloudfrontlab-s3bucket-secondary as secondary origin.

For failover criteria, choose 404 Not Found and 403 Forbidden.

Create Origin Group

Create an origin group to provide rerouting during a failover event. You can associate an origin group with a cache behavior to have requests routed from a primary origin to a secondary origin for failover. You must have two origins for your distribution before you can create an origin group. Please note that with an origin group, you can only use GET, HEAD, and OPTIONS HTTP methods in your cache behavior. [Learn more](#)

Origins * Custom-ec2-34-230-14-155.compute-1.amazonaws.com Add

Priority	Origin ID	
▲ ▼ 1 (Primary)	S3-cloudfrontlab-s3bucket	✕
▲ ▼ 2	S3-cloudfrontlab-s3bucket-secondary	✕

You must add two origins for the origin group.
Arrange the origins in priority order, based on which origin you want CloudFront to send requests to first. The origin that you list first is the primary origin.

Failover criteria * Select the status codes to use as the failover criteria. When the codes that you select are returned by the primary origin, requests are rerouted to the secondary origin.

☐ 500 Internal Server Error ☐ 504 Gateway Timeout
☐ 502 Bad Gateway ☒ 404 Not Found
☐ 503 Service Unavailable ☒ 403 Forbidden

Origin Group ID * OriginGroup-S3-cloudfrontlab-s3bucket

- Edit the default behavior of the distribution to use the new Origin Group, so that we can test failover. Go to CloudFront Behaviors Tab, select

Default(*), and click Edit.

Edit the behavior to use the Origin Group we created in previous step.

CloudFront Distributions > E1L5BO08BU16W1

General Origins and Origin Groups **Behaviors** Error Pages Restrictions Invalidations Tags

CloudFront compares a request for an object with the path patterns in your cache behaviors based on the order of the cache behaviors in your distribution. Arrange cache behaviors in the order in which you want CloudFront to evaluate them.

Create Behavior **Edit** Delete Change Precedence: Move Up Move Down Save

	Precedence	Path Pattern	Origin or Origin Group	Viewer Protocol Policy	Forwarded Query Strings
<input type="checkbox"/>	0	/api	Custom-ec2-34-230-14-155.compute-1.	Redirect HTTP to HTTPS	Yes
<input checked="" type="checkbox"/>	1	Default (*)	S3-cloudfrontlab-s3bucket	Redirect HTTP to HTTPS	No

Edit Behavior

Default Cache Behavior Settings

Path Pattern Default (*)

Origin or Origin Group OriginGroup-S3-cloudfrontlab-s3bucket

Viewer Protocol Policy
☐ HTTP and HTTPS
☒ Redirect HTTP to HTTPS
☐ HTTPS Only

Allowed HTTP Methods
☒ GET, HEAD
☐ GET, HEAD, OPTIONS
☐ GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Field-level Encryption Config

Cached HTTP Methods GET, HEAD (Cached by default)

7. After the distribution status changed to Deployed, request new-index.html page from CloudFront, you can see your secondary S3 bucket origin serve your request correctly.

← → ↺ 🔒 https://d1wj00cdxoism.cloudfront.net/new-index.html

CloudFront Lab

Hi, this is a page from my secondary Origin! We now support Origin group and failover!