	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology	
Subject: Cloud Developing (01CT0720)	**Aim: Implement CloudFront for caching and application security.**	
Experiment No: 10	**Date:**	**Enrolment No: 92200133030**

Aim :- Implement CloudFront for caching and application security.

Lab overview and objectives

In this lab, you will create an Amazon CloudFront distribution to reduce network latency for café website users and deliver the website securely over HTTPS. You will also secure access to the website and REST API endpoints using AWS WAF, which is a service that provides a web application firewall. Finally, you will configure a CloudFront function on the website and adjust the cached file expiration time on the website content.

After completing this lab, you should be able to:

- Create a CloudFront distribution to cache Amazon Simple Storage Service (Amazon S3) objects
- Configure a website hosted on Amazon S3 to be available through HTTPS using CloudFront
- Secure access to the CloudFront distribution based on the network origin of the request
- Secure a REST API endpoint based on the network origin of the request using AWS WAF
- Configure a CloudFront function to affect website behavior from the edge
- Adjust max-age caching settings on a CloudFront distribution

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

Scenario

Sofía is pleased with how the café website development project is coming along. She has developed the core serverless application that displays menu items on the website. She also integrated the coffee suppliers web application into the main site and is using Amazon ElastiCache features for the suppliers part of the site.

However, she knows that some essential features are still missing. One feature is that the website still runs on HTTP and does not yet support HTTPS. Sofía also wants to ensure that the website will load quickly for users globally. She knows that AWS has many Regions and Availability Zones, but they also have edge locations, which are even closer to users around the globe. She decides to host the café website on a proper content delivery network (CDN), and she has opted to use the CloudFront service.

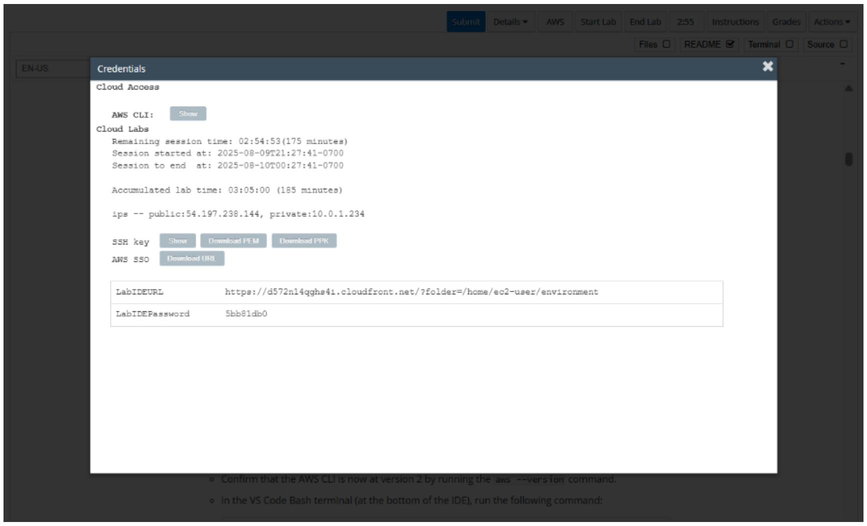
In this lab, you will again play the role of Sofía to continue to develop the café's web application.

Task 1: Preparing the lab

Connect to the VS Code IDE.

1. At the top of these instructions, choose Details followed by **AWS: Show**
2. Copy values from the table **similar** to the following and paste it into an editor of your choice for use later.
   a. **LabIDEURL**

b. **LabIDEPassword**

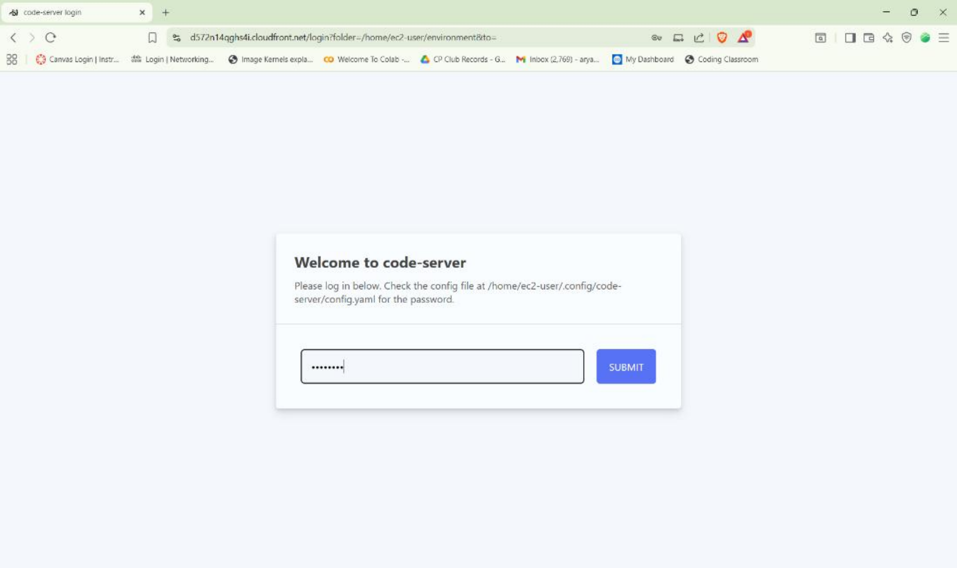


```
Cloud Access

  AWS CLI:
Cloud Labs
  Remaining session time: 02:54:53(175 minutes)
  Session started at: 2025-08-09T21:27:41-0700
  Session to end  at: 2025-08-10T00:27:41-0700

  Accumulated lab time: 03:05:00 (185 minutes)

  ips -- public:54.197.238.144, private:10.0.1.234

  SSH key
  AWS SSO
```

LabIDEURL	https://d572n14qghs4i.cloudfront.net/?folder=/home/ec2-user/environment
LabIDEPassword	5bb81db0

- Confirm that the AWS CLI is now at version 2 by running the `aws --version` command.
- In the VS Code Bash terminal (at the bottom of the IDE), run the following command:

3. In a new browser tab, paste the value for **LabIDEURL** to open the VS Code IDE.
4. On the prompt window **Welcome to code-server**, enter the value for **LabIDEPassword** you copied to the editor earlier, choose **Submit** to open the VS Code IDE.

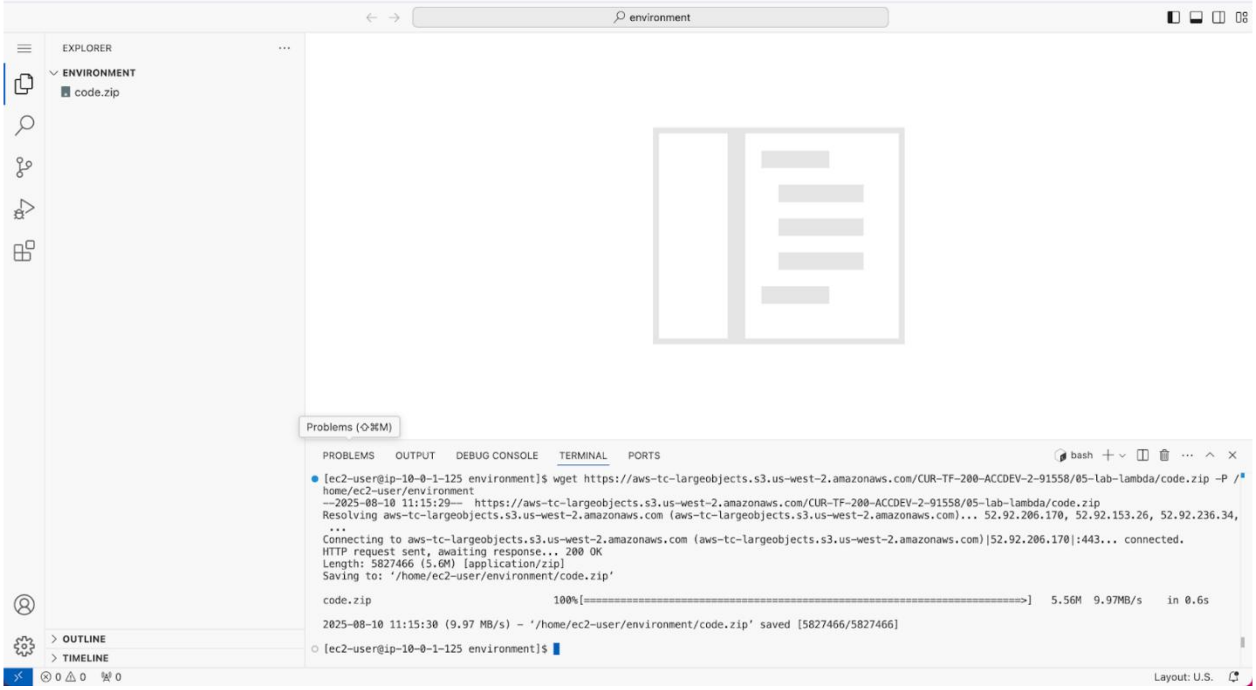


Welcome to code-server

Please log in below. Check the config file at /home/ec2-user/.config/code-server/config.yaml for the password.

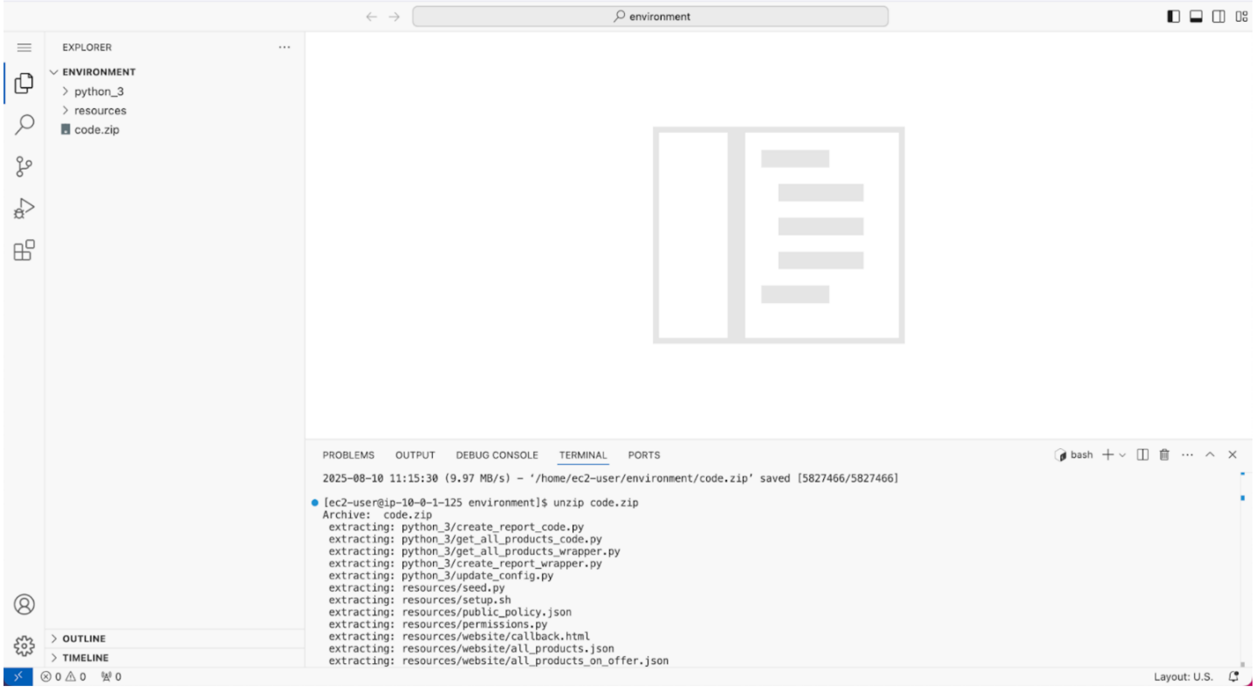
5. Download and extract the files that you need for this lab.
   - In the VS Code bash terminal (located at the bottom of the IDE), run the following commands:

```
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-
ACCDEV-2-91558/09-lab-cloudfront/code.zip -P /home/ec2-user/environment
```



```
● [ec2-user@ip-10-0-1-125 environment]$ wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2-91558/05-lab-lambda/code.zip -P /
home/ec2-user/environment
--2025-08-10 11:15:29--  https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2-91558/05-lab-lambda/code.zip
Resolving aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)... 52.92.206.170, 52.92.153.26, 52.92.236.34,
 ...
Connecting to aws-tc-largeobjects.s3.us-west-2.amazonaws.com (aws-tc-largeobjects.s3.us-west-2.amazonaws.com)|52.92.206.170|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5827466 (5.6M) [application/zip]
Saving to: '/home/ec2-user/environment/code.zip'

code.zip                      100%[======================================================================>]   5.56M  9.97MB/s    in 0.6s

2025-08-10 11:15:30 (9.97 MB/s) - '/home/ec2-user/environment/code.zip' saved [5827466/5827466]

○ [ec2-user@ip-10-0-1-125 environment]$ 
```

6. You should see that the **code.zip** file was downloaded to the VS Code IDE and is now in the left navigation pane.
   - Extract the file by running the following command:
     `unzip code.zip`

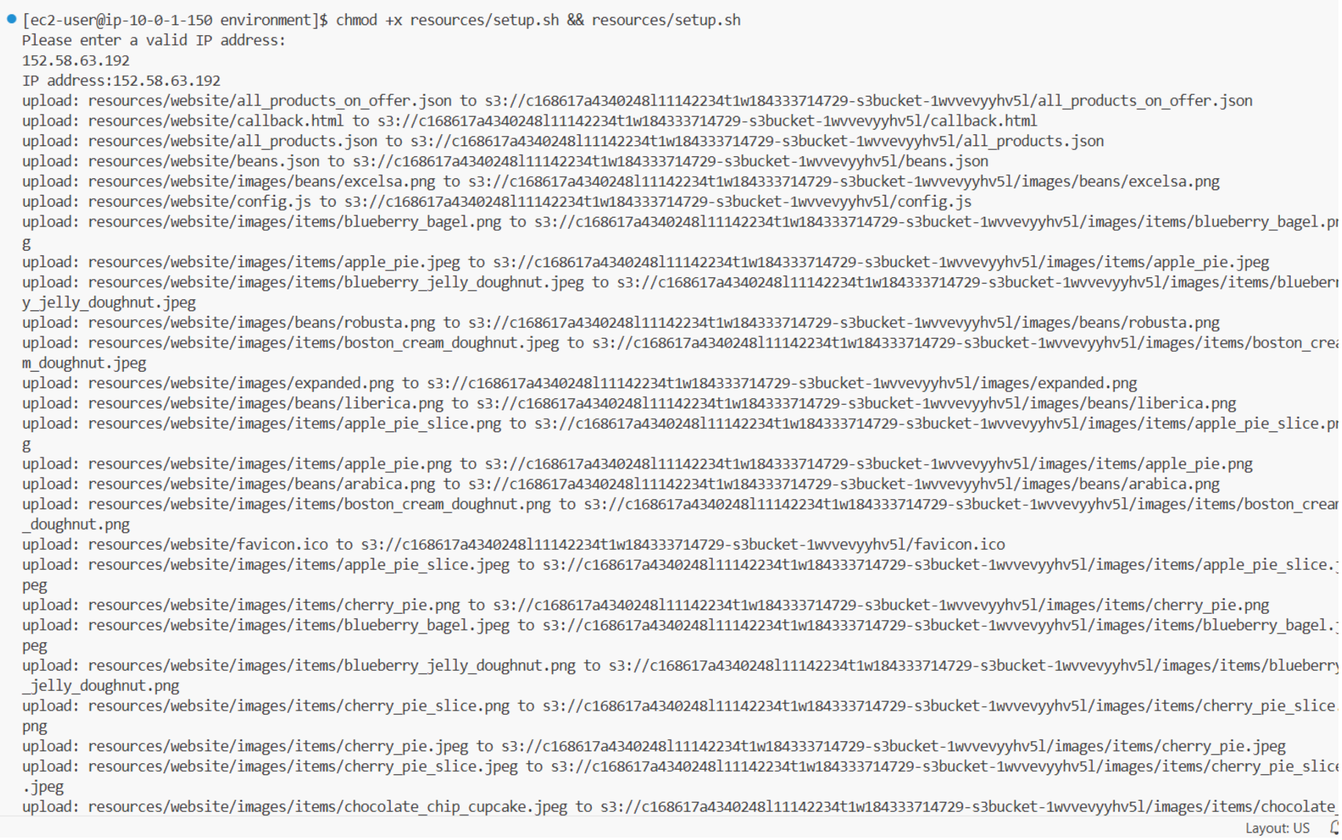


```
2025-08-10 11:15:30 (9.97 MB/s) - '/home/ec2-user/environment/code.zip' saved [5827466/5827466]

● [ec2-user@ip-10-0-1-125 environment]$ unzip code.zip
Archive:  code.zip
  extracting: python_3/create_report_code.py
  extracting: python_3/get_all_products_code.py
  extracting: python_3/get_all_products_wrapper.py
  extracting: python_3/create_report_wrapper.py
  extracting: python_3/update_config.py
  extracting: resources/seed.py
  extracting: resources/setup.sh
  extracting: resources/public_policy.json
  extracting: resources/permissions.py
  extracting: resources/website/callback.html
  extracting: resources/website/all_products.json
  extracting: resources/website/all_products_on_offer.json
```

7. Run a script that upgrades the version of the AWS CLI installed on the VS Code IDE.
   - To set permissions on the script and then run it, run the following commands in the Bash terminal:

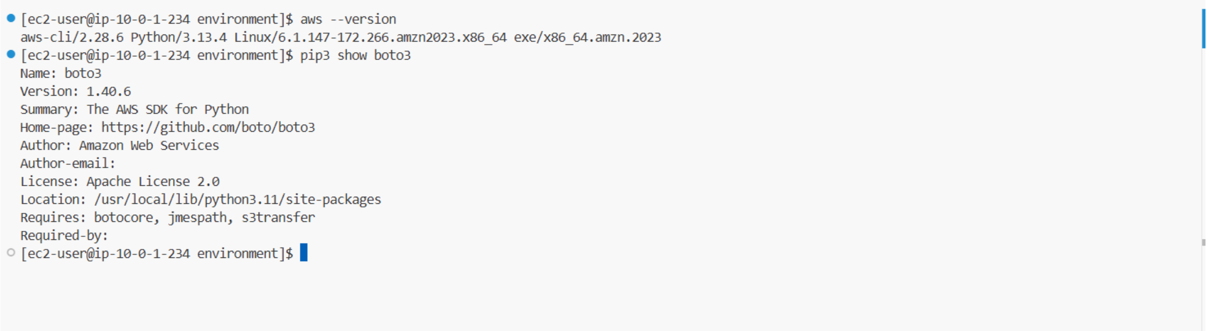
     ```
     chmod +x ./resources/setup.sh && ./resources/setup.sh
     ```

   When prompted for an IP address, enter the IPv4 address that the internet uses to contact your computer. You can find your IPv4 address at https://whatismyipaddress.com.



```
[ec2-user@ip-10-0-1-150 environment]$ chmod +x resources/setup.sh && resources/setup.sh
Please enter a valid IP address:
152.58.63.192
IP address:152.58.63.192
upload: resources/website/all_products_on_offer.json to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/all_products_on_offer.json
upload: resources/website/callback.html to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/callback.html
upload: resources/website/all_products.json to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/all_products.json
upload: resources/website/beans.json to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/beans.json
upload: resources/website/images/beans/excelsa.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/beans/excelsa.png
upload: resources/website/config.js to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/config.js
upload: resources/website/images/items/blueberry_bagel.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/blueberry_bagel.pn
g
upload: resources/website/images/items/apple_pie.jpeg to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/apple_pie.jpeg
upload: resources/website/images/items/blueberry_jelly_doughnut.jpeg to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/blueberr
y_jelly_doughnut.jpeg
upload: resources/website/images/beans/robusta.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/beans/robusta.png
upload: resources/website/images/items/boston_cream_doughnut.jpeg to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/boston_crea
m_doughnut.jpeg
upload: resources/website/images/expanded.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/expanded.png
upload: resources/website/images/beans/liberica.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/beans/liberica.png
upload: resources/website/images/items/apple_pie_slice.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/apple_pie_slice.pn
g
upload: resources/website/images/items/apple_pie.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/apple_pie.png
upload: resources/website/images/beans/arabica.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/beans/arabica.png
upload: resources/website/images/items/boston_cream_doughnut.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/boston_crear
_doughnut.png
upload: resources/website/favicon.ico to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/favicon.ico
upload: resources/website/images/items/apple_pie_slice.jpeg to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/apple_pie_slice.
peg
upload: resources/website/images/items/cherry_pie.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/cherry_pie.png
upload: resources/website/images/items/blueberry_bagel.jpeg to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/blueberry_bagel.
peg
upload: resources/website/images/items/blueberry_jelly_doughnut.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/blueberry
_jelly_doughnut.png
upload: resources/website/images/items/cherry_pie_slice.png to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/cherry_pie_slice
png
upload: resources/website/images/items/cherry_pie.jpeg to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/cherry_pie.jpeg
upload: resources/website/images/items/cherry_pie_slice.jpeg to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/cherry_pie_slice
.jpeg
upload: resources/website/images/items/chocolate_chip_cupcake.jpeg to s3://c168617a4340248l11142234t1w184333714729-s3bucket-1wvvevyyhv5l/images/items/chocolate_
```

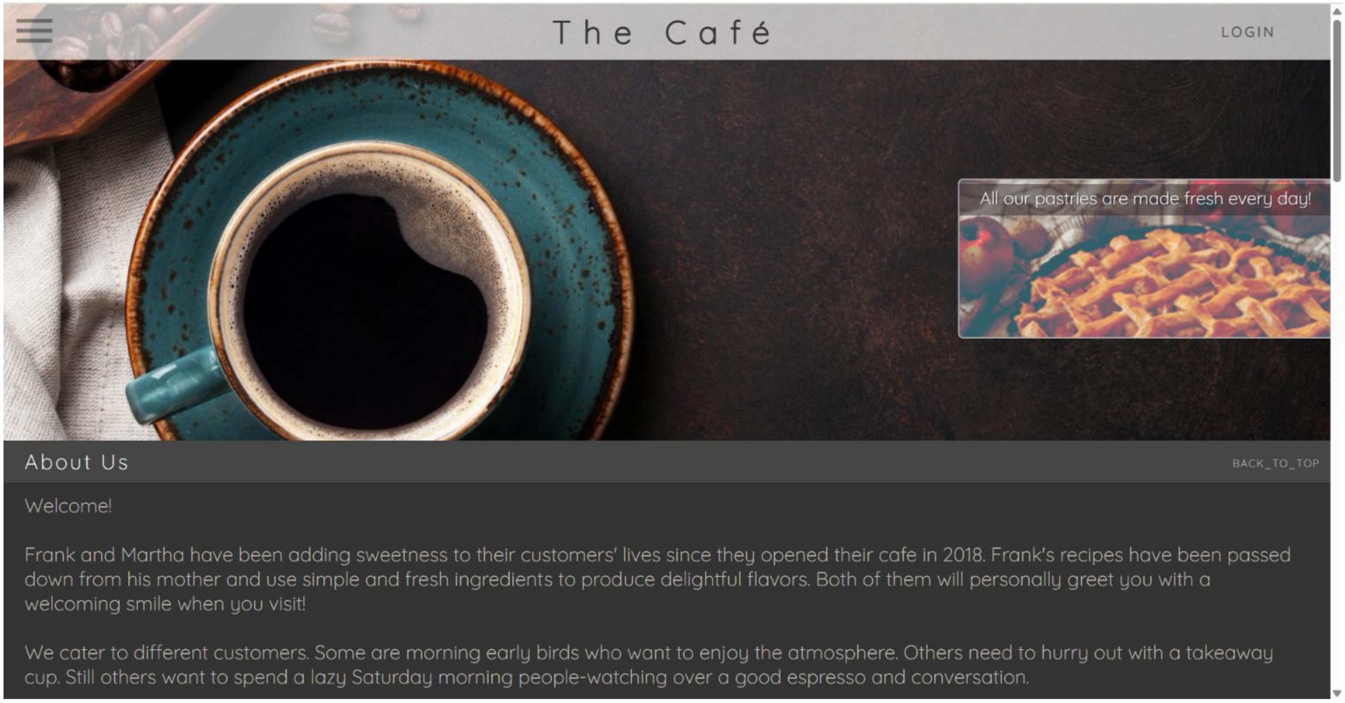
8. Verify the AWS CLI version and also verify that the SDK for Python is installed.
   - Confirm that the AWS CLI is now at version 2 by running the **`aws --version`** command.
   - In the VS Code Bash terminal (at the bottom of the IDE), run the following command:
     `pip3 show boto3`



```
[ec2-user@ip-10-0-1-234 environment]$ aws --version
aws-cli/2.28.6 Python/3.13.4 Linux/6.1.147-172.266.amzn2023.x86_64 exe/x86_64.amzn.2023
[ec2-user@ip-10-0-1-234 environment]$ pip3 show boto3
Name: boto3
Version: 1.40.6
Summary: The AWS SDK for Python
Home-page: https://github.com/boto/boto3
Author: Amazon Web Services
Author-email:
License: Apache License 2.0
Location: /usr/local/lib/python3.11/site-packages
Requires: botocore, jmespath, s3transfer
Required-by:
[ec2-user@ip-10-0-1-234 environment]$ 
```

9. Note the metadata settings on the objects stored in the S3 bucket, and then verify that you can access the café website.

- Navigate to the Amazon S3 console.
- Choose the link for the bucket that has *-s3bucket* in the name.
- Choose the **index.html** link.
- Scroll down to the **Metadata** section.
- Two key-value pairs are listed. The **Cache-Control** key-value pair has a value of **max-age=0**. This was set when you ran *setup.sh* in an earlier step. Line 21 of that script ran the command aws s3 cp ./resources/website s3://$bucket/ --recursive --cache-control "max-age=0" to set this metadata value on every file that it uploaded to the bucket. You'll learn about the significance of this setting later in the lab.
- At the top of the page, open the **Object URL** in a new browser tab.
- The café website displays. If it doesn't, see the following troubleshooting tip.



Welcome!

Frank and Martha have been adding sweetness to their customers' lives since they opened their cafe in 2018. Frank's recipes have been passed down from his mother and use simple and fresh ingredients to produce delightful flavors. Both of them will personally greet you with a welcoming smile when you visit!

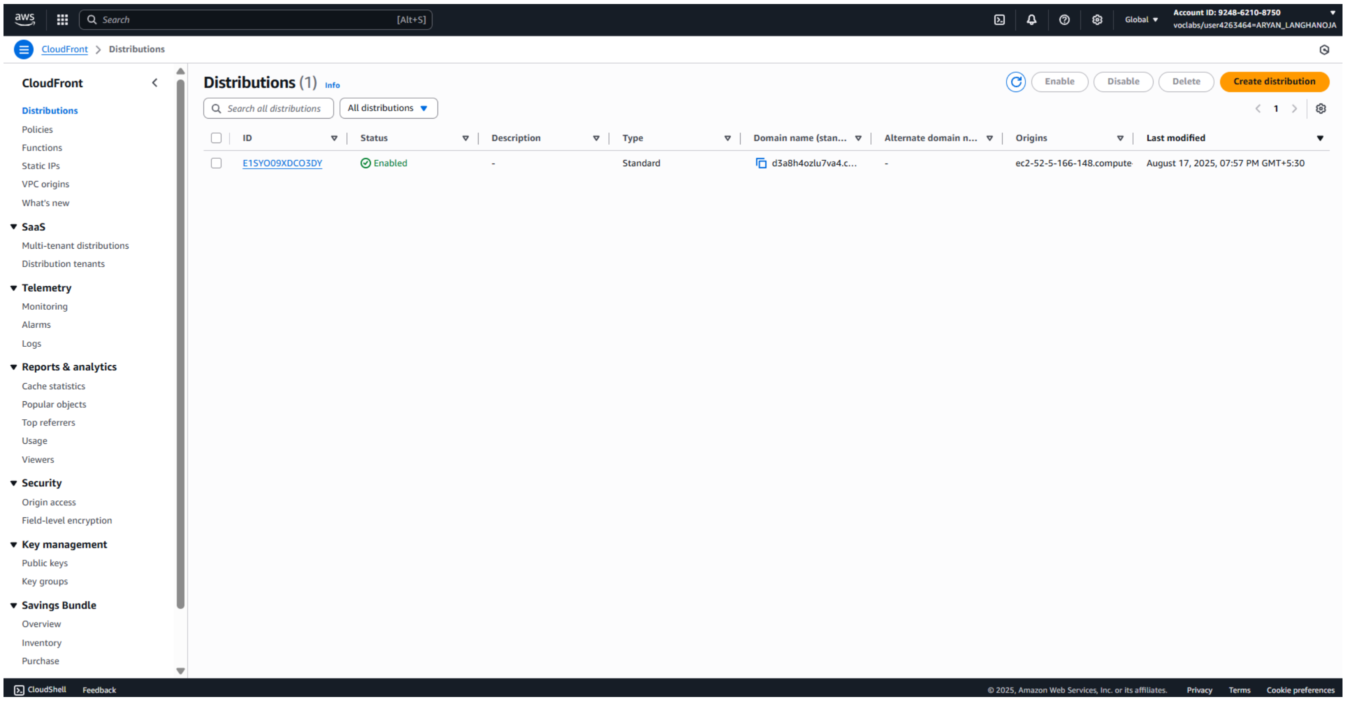
We cater to different customers. Some are morning early birds who want to enjoy the atmosphere. Others need to hurry out with a takeaway cup. Still others want to spend a lazy Saturday morning people-watching over a good espresso and conversation.

Task 2: Configuring a distribution for static website content

In this task, you will configure the café website, which is hosted on Amazon S3, to be available through a CloudFront distribution. With the distribution, you can enable HTTPS access to the website.

10. Begin to configure a CloudFront distribution for the café website hosted on Amazon S3.

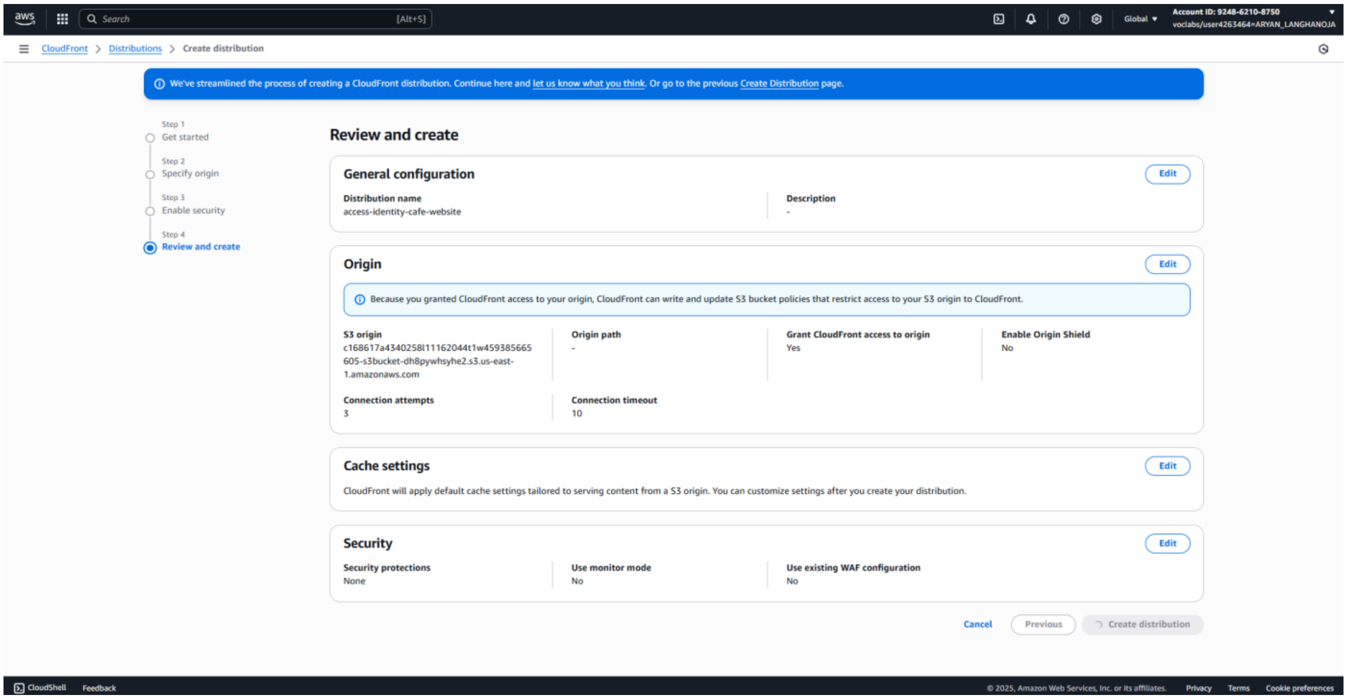
- Navigate to the CloudFront console.
- Choose **Create a CloudFront distribution**.



Distributions (1) Info

ID	Status	Description	Type	Domain name (stan...	Alternate domain n...	Origins	Last modified
E1SYO09XDCO3DY	Enabled	-	Standard	d3a8h4ozlu7va4.c...	-	ec2-52-5-166-148.compute	August 17, 2025, 07:57 PM GMT+5:30

11. Configure the **Origin** settings for the distribution.
    - **Note**: If a setting is not specified in the following steps, use the default.
    - **Origin domain**: Search for and choose the S3 bucket that has *-s3bucket* in the name. This bucket contains the website code.
    - In the **Origin access** settings.
    - Choose **Legacy access identities**.
    - Choose **Create new OAI**.
    - **Name**: Enter access-identity-cafe-website
    - Choose **Create**.
    - **Bucket policy**: Choose **Yes, update the bucket policy**.
    - In the **Add custom header** section, choose **Add header** and configure:
    - **Header name**: Enter cf
    - **Value**: Enter 1
    - **Note**: The custom header is not an important requirement for this lab, but it will be useful in a later lab.



We've streamlined the process of creating a CloudFront distribution. Continue here and let us know what you think. Or go to the previous Create Distribution page.

Review and create

General configuration

Distribution name: access-identity-cafe-website
Description: -

Origin

Because you granted CloudFront access to your origin, CloudFront can write and update S3 bucket policies that restrict access to your S3 origin to CloudFront.

S3 origin: c168617a4340258l11162044t1w459385665605-s3bucket-dh8pywhsyhe2.s3.us-east-1.amazonaws.com
Origin path: -
Grant CloudFront access to origin: Yes
Enable Origin Shield: No
Connection attempts: 3
Connection timeout: 10

Cache settings

CloudFront will apply default cache settings tailored to serving content from a S3 origin. You can customize settings after you create your distribution.

Security

Security protections: None
Use monitor mode: No
Use existing WAF configuration: No

12. Configure the **Default cache behavior** settings.
    a. **Path pattern**: Keep the **Default (*)** setting. This means that all requests will go to the origin.
    b. **Viewer protocol policy**: Choose **Redirect HTTP to HTTPS**. This will help users find the website, even if they load the HTTP URL.
    c. **Allowed HTTP methods**: Keep the default **GET, HEAD** setting.
    d. **Restrict viewer access**: Keep the default **No** setting. This will be a public website.

13. Configure the **Cache key and origin requests** settings.
    a. Choose **Legacy cache settings**.
    b. Keep the settings for **Headers**, **Query strings**, and **Cookies** as **None**.
    c. **Object caching**: Choose **Use origin cache headers**.

Note: Recall the metadata key-value pair with key **Cache-Control** and value **max-age=0**. This metadata is set on all of the objects stored in the origin S3 bucket. By choosing **Use origin cache headers** for this distribution, you ensure that this distribution will inherit this setting applied to all of the objects in the Amazon S3 origin.

    d. Under **Web Application Firewall (WAF)** choose **Do not enable security protections**.

14. Configure the **Settings** section of the distribution.
    a. **Price class**: Keep the default **Use all edge locations** setting.
    b. **Alternative domain name (CNAME)**: Do not enter anything.

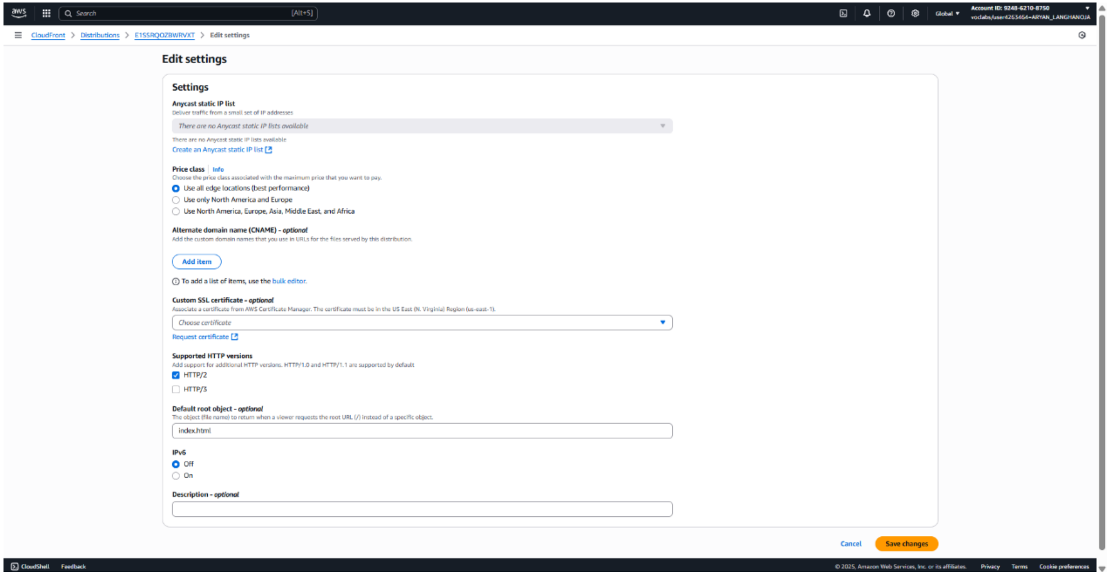
Note: You could specify a unique domain name here; however, this lab doesn't have one.

    c. **Custom SSL certificate**: Choose **None** from dropdown.

Note: If you owned a domain and had a TLS or SSL certificate from a certificate authority, you would upload it here. For this lab, you will use the default CloudFront certificate (*.cloudfront.net), so you do not need to specify a custom certificate.

d. **Supported HTTP versions**: Select **HTTP/2**.
e. **Default root object**: Enter index.html
f. **IPv6**: Off

Note: This is the café website homepage hosted in the S3 bucket.



Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA

Edit settings

Settings

Anycast static IP list
Deliver traffic from a small set of IP addresses
There are no Anycast static IP lists available
There are no Anycast static IP lists available
Create an Anycast static IP list

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

Alternate domain name (CNAME) - *optional*
Add the custom domain names that you use in URLs for the files served by this distribution.

To add a list of items, use the bulk editor.

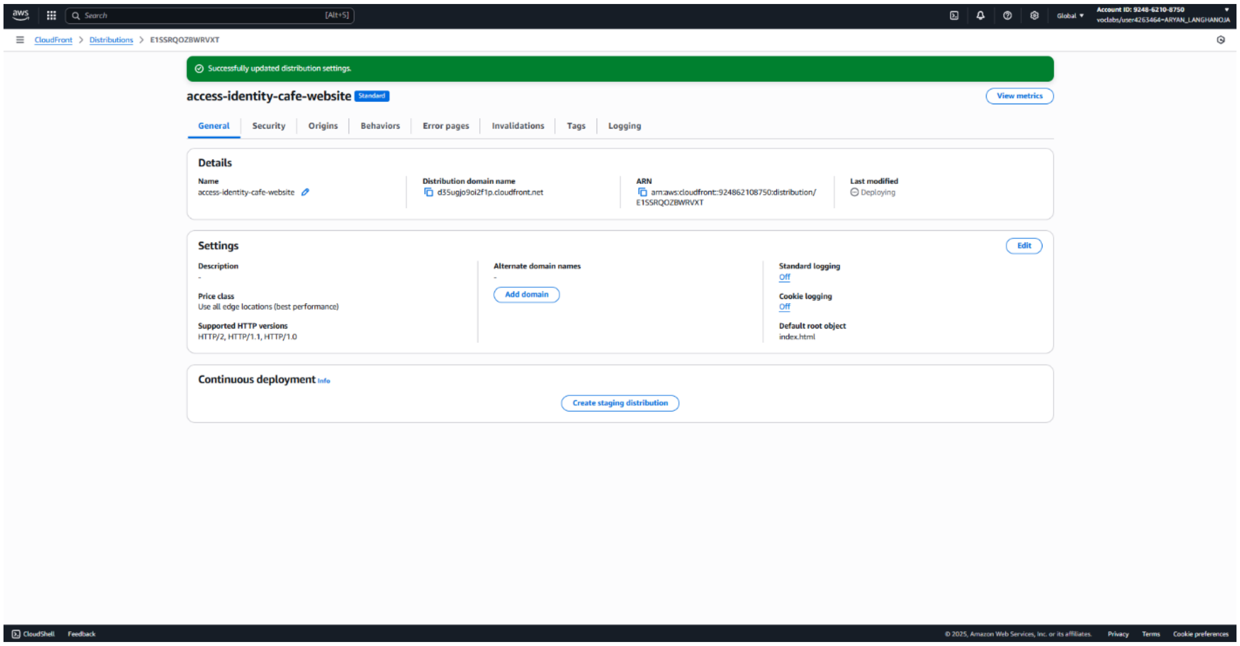
Custom SSL certificate - *optional*
Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).
Choose certificate

Supported HTTP versions
Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default
☑ HTTP/2
□ HTTP/3

Default root object - *optional*
The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.
index.html

IPv6
● Off
○ On

Description - *optional*



Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA

Successfully updated distribution settings.

access-identity-cafe-website

Details

Name	Distribution domain name	ARN	Last modified
access-identity-cafe-website	d35ugjo9oi2f1p.cloudfront.net	arn:aws:cloudfront::924862108750:distribution/E1SSRQOZBWRVXT	Deploying

Settings

Description
-

Price class
Use all edge locations (best performance)

Supported HTTP versions
HTTP/2, HTTP/1.1, HTTP/1.0

Alternate domain names
-

Standard logging
Off

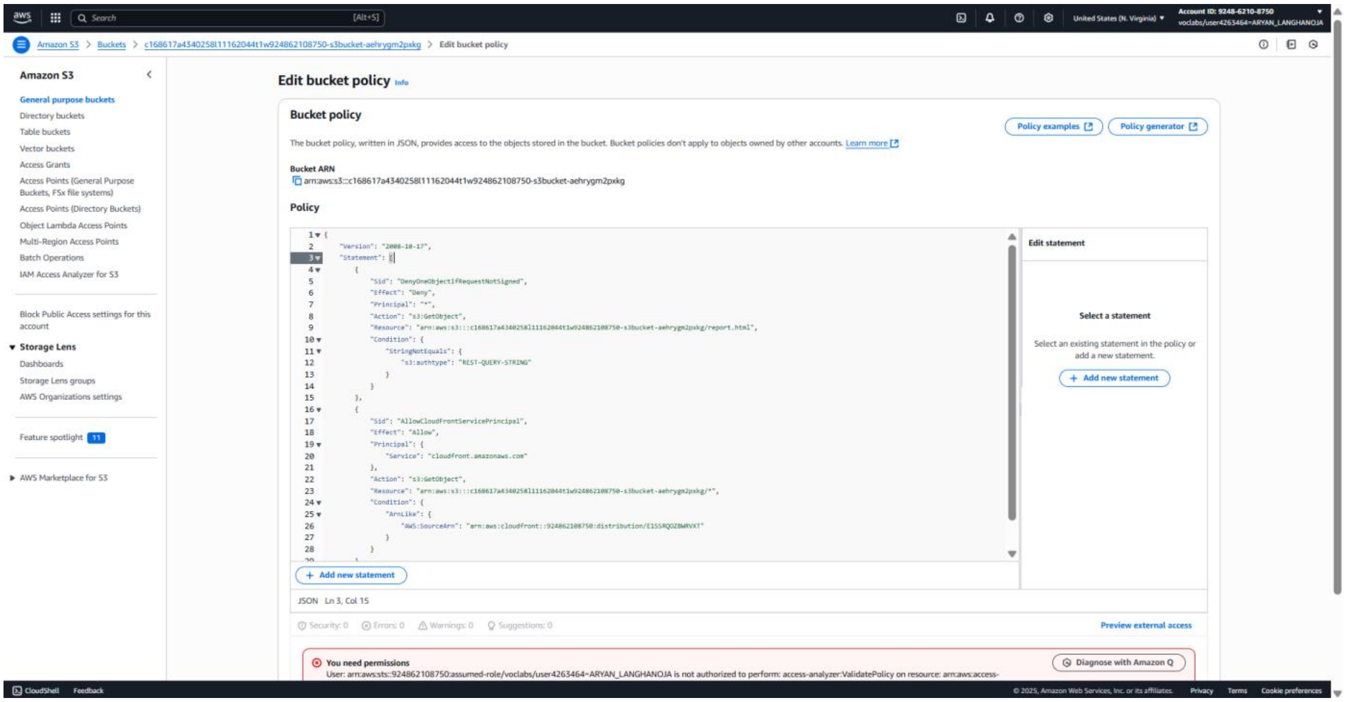
Cookie logging
Off

Default root object
index.html

Continuous deployment Info

15. Update the bucket policy.

- In a new browser tab, navigate to the Amazon S3 console.
- Choose the link for the bucket that has *-s3bucket* in the name.
- Choose the **Permissions** tab.
- In the **Bucket policy** section, choose **Edit**.
- In the **Policy** code section, delete lines 4 through 17


aws
Search [Alt+S]
United States (N. Virginia)
Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA
Amazon S3 > Buckets > c168617a4340258l11162044t1w924862108750-s3bucket-aehrygm2pxkg > Edit bucket policy
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Vector buckets
Access Grants
Access Points (General Purpose Buckets, FSx file systems)
Access Points (Directory Buckets)
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
AWS Marketplace for S3
Edit bucket policy Info
Policy examples
Policy generator
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
+ Add new statement
+ Add new statement
JSON Ln 3, Col 15
Security: 0
Errors: 0
Warnings: 0
Suggestions: 0
Preview external access
Diagnose with Amazon Q
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Bucket ARN

arn:aws:s3:::c168617a4340258l11162044t1w924862108750-s3bucket-aehrygm2pxkg

Policy

```
1  {
2      "Version": "2008-10-17",
3      "Statement": [
4          {
5              "Sid": "DenyOneObjectIfRequestNotSigned",
6              "Effect": "Deny",
7              "Principal": "*",
8              "Action": "s3:GetObject",
9              "Resource": "arn:aws:s3:::c168617a4340258l11162044t1w924862108750-s3bucket-aehrygm2pxkg/report.html",
10             "Condition": {
11                 "StringNotEquals": {
12                     "s3:authtype": "REST-QUERY-STRING"
13                 }
14             }
15         },
16         {
17             "Sid": "AllowCloudFrontServicePrincipal",
18             "Effect": "Allow",
19             "Principal": {
20                 "Service": "cloudfront.amazonaws.com"
21             },
22             "Action": "s3:GetObject",
23             "Resource": "arn:aws:s3:::c168617a4340258l11162044t1w924862108750-s3bucket-aehrygm2pxkg/*",
24             "Condition": {
25                 "ArnLike": {
26                     "AWS:SourceArn": "arn:aws:cloudfront::924862108750:distribution/E1SSRQOZ8WRVXT"
27                 }
28             }
```

You need permissions

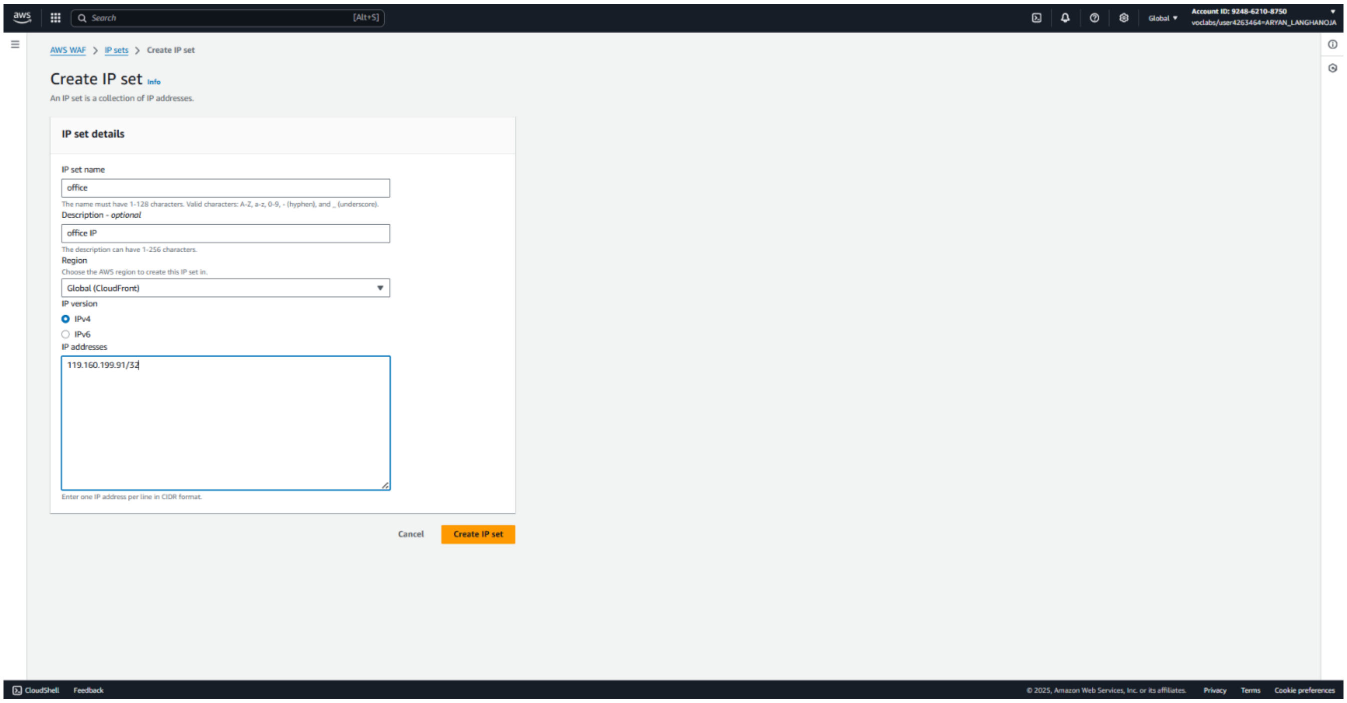
User: arn:aws:sts::924862108750:assumed-role/voclabs/user4263464=ARYAN_LANGHANOJA is not authorized to perform: access-analyzer:ValidatePolicy on resource: arn:aws:access-

Task 3: Securing network access to the distribution using AWS WAF

16. First, create an IP set for your IP address.
    a. In the AWS Management Console, search for waf and choose **WAF & Shield**.
    b. In the left navigation pane, choose **IP sets**.
    c. Choose **Create IP set** and configure:
        i. **IP set name**: Enter office
        ii. **Description**: Enter office IP
        iii. **Region**: Choose **Global (CloudFront)**.
        iv. **IP addresses**: Enter <ip-address>/32 where <ip-address> is your public IPv4 address, as identified by whatismyipaddress.com.

            Note: Be sure to include the /32 at the end of the IP address.

        v. Choose **Create IP set**.



AWS WAF > IP sets > Create IP set

Create IP set Info

An IP set is a collection of IP addresses.

IP set details

IP set name: office

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - *optional*: office IP

The description can have 1-256 characters.

Region

Choose the AWS region to create this IP set in.

Global (CloudFront)

IP version

● IPv4

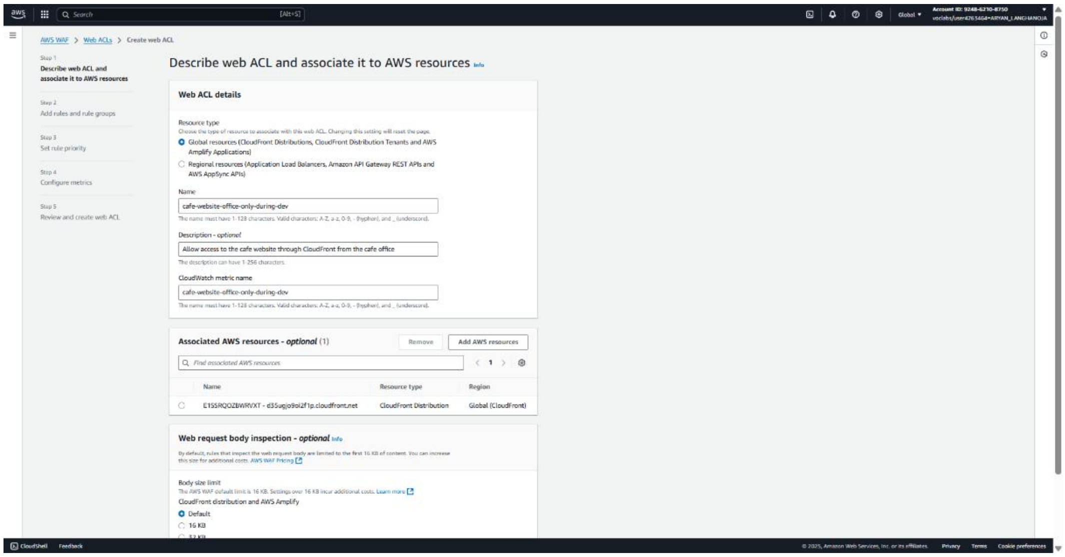
○ IPv6

IP addresses

```
119.160.199.91/32
```

Enter one IP address per line in CIDR format.

17. Begin to create a web ACL.
    a. In the left navigation pane, choose **Web ACLs**.
    b. Choose **Create web ACL**.
    c. In the **Web ACL details** section, configure:
        i. **Resource type**: Choose **CloudFront distributions**.
        ii. **Name**: Enter `cafe-website-office-only-during-dev`
        iii. **Description**: Enter `Allow access to the cafe website through CloudFront from the cafe office`
        iv. **CloudWatch metric name**: Enter `cafe-website-office-only-during-dev`
    d. In the **Associated AWS resources** section, configure:
        i. Choose **Add AWS resources**.
        ii. Select the CloudFront distribution that you created.
        iii. Choose **Add**.
        iv. Select the CloudFront distribution again, and then choose **Next**.



AWS WAF > Web ACLs > Create web ACL

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

Describe web ACL and associate it to AWS resources

Web ACL details

Resource type
Choose the type of resource to associate with this web ACL. Changing this setting will reset the page.
● Global resources (CloudFront Distributions, CloudFront Distribution Tenants and AWS Amplify Applications)
○ Regional resources (Application Load Balancers, Amazon API Gateway REST APIs and AWS AppSync APIs)

Name: cafe-website-office-only-during-dev
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - *optional*: Allow access to the cafe website through CloudFront from the cafe office
The description can have 1-256 characters.

CloudWatch metric name: cafe-website-office-only-during-dev
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

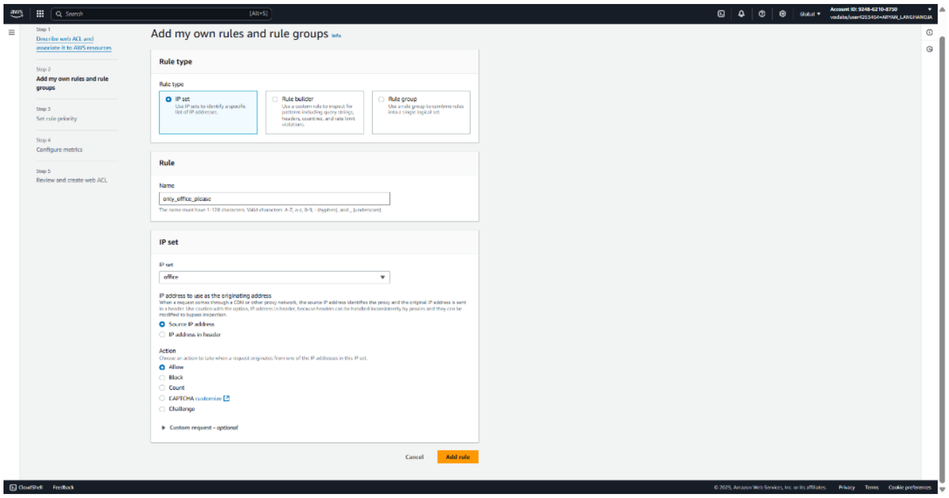
Associated AWS resources - *optional* (1)

Name	Resource type	Region
E1SSRQOZBWRVXT - d35ugjo9oi2f1p.cloudfront.net	CloudFront Distribution	Global (CloudFront)

Web request body inspection - *optional*
By default, rules that inspect the web request body are limited to the first 16 KB of content. You can increase this size for additional costs. AWS WAF Pricing

Body size limit
The AWS WAF default limit is 16 KB. Settings over 16 KB incur additional costs. Learn more
CloudFront distribution and AWS Amplify
● Default
○ 16 KB
○ 32 KB

18. Add a rule to the web ACL configuration to allow requests from the *office* IP set.
    a. In the **Rules** section, choose **Add rules**, **Add my own rules and rule groups**.
    b. **Rule type**: Choose **IP set**.
    c. **Name**: Enter only_office_please
    d. **IP set:** Choose the *office* IP set that you just created.
    e. **IP address to use as the originating address**: Keep the default **Source IP address** setting.
    f. **Action:** Choose **Allow**.
    g. Choose **Add rule**.



Step 1 Describe web ACL and associate it to AWS resources
Step 2 Add my own rules and rule groups
Step 3 Set rule priority
Step 4 Configure metrics
Step 5 Review and create web ACL

Add my own rules and rule groups

Rule type
● IP set — Use IP sets to identify a specific list of IP addresses.
○ Rule builder — Use a custom rule to inspect for patterns including query strings, headers, countries, and rate limit violations.
○ Rule group — Use a rule group to combine rules into a single logical set.

Rule
Name: only_office_please
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

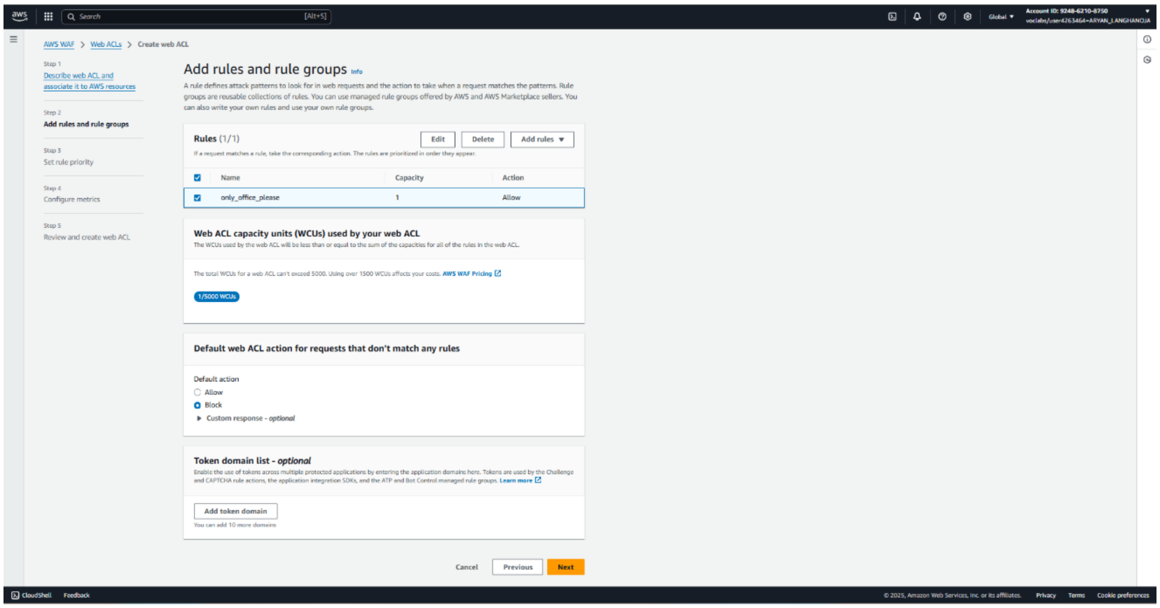
IP set
IP set: office

IP address to use as the originating address
When a request comes through a CDN or other proxy network, the source IP address identifies the proxy and the original IP address is sent in a header. Use caution with the option, IP address in header, because headers can be handled inconsistently by proxies and they can be modified to bypass inspection.
● Source IP address
○ IP address in header

Action
Choose an action to take when a request originates from one of the IP addresses in this IP set.
● Allow
○ Block
○ Count
○ CAPTCHA customize
○ Challenge

Custom request - *optional*

19. Update the new web ACL rule to block any requests that don't match the rule.
    a. In the **Rules** section, select the **only_office_please** rule.
    b. In the **Default web ACL action** section, for **Default action**, choose **Block**.
    c. Choose **Next**.



aws
Search [Alt+S]
Global
Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA
AWS WAF > Web ACLs > Create web ACL
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
Edit
Delete
Add rules ▾
Add token domain
Cancel
Previous
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Add rules and rule groups Info

A rule defines attack patterns to look for in web requests and the action to take when a request matches the patterns. Rule groups are reusable collections of rules. You can use managed rule groups offered by AWS and AWS Marketplace sellers. You can also write your own rules and use your own rule groups.

Rules (1/1)

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

☑	Name	Capacity	Action
☑	only_office_please	1	Allow

Web ACL capacity units (WCUs) used by your web ACL

The WCUs used by the web ACL will be less than or equal to the sum of the capacities for all of the rules in the web ACL.

The total WCUs for a web ACL can't exceed 5000. Using over 1500 WCUs affects your costs. AWS WAF Pricing

1/5000 WCUs

Default web ACL action for requests that don't match any rules

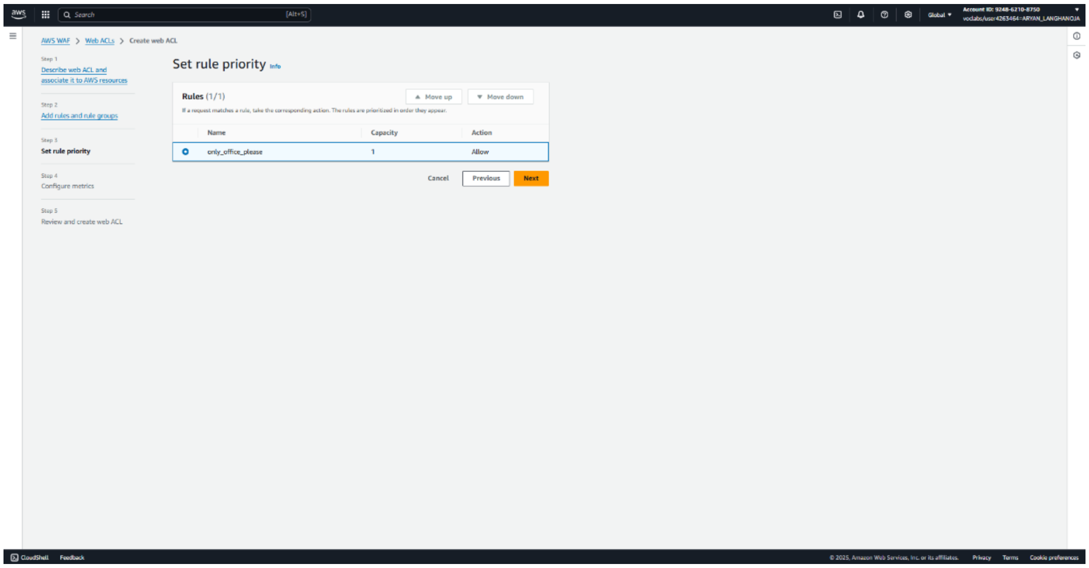
Default action
○ Allow
● Block
▸ Custom response - *optional*

Token domain list - *optional*

Enable the use of tokens across multiple protected applications by entering the application domains here. Tokens are used by the Challenge and CAPTCHA rule actions, the application integration SDKs, and the ATP and Bot Control managed rule groups. Learn more

You can add 10 more domains

20. Set the rule priority, configure metrics, and create the web ACL.
    a. Choose the **only_office_please** rule.
    b. Choose **Next**.
    c. Keep all of the default metrics settings, and choose **Next** again.
    d. Review the settings, and at the bottom of the page, choose **Create web ACL**.



aws
Search [Alt+S]
Global
Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA
AWS WAF > Web ACLs > Create web ACL
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
▲ Move up
▼ Move down
Cancel
Previous
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

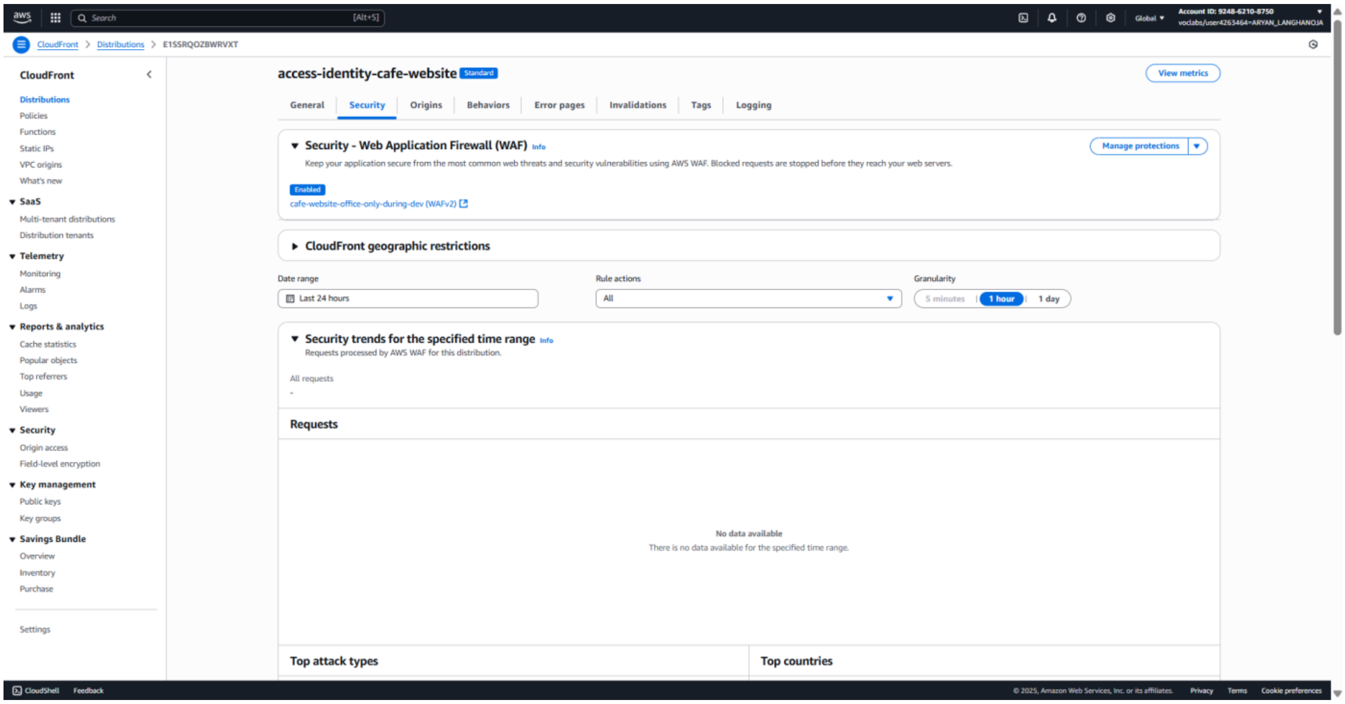
Set rule priority Info

Rules (1/1)

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

	Name	Capacity	Action
●	only_office_please	1	Allow

21. Confirm that the web ACL configuration has been applied to the CloudFront distribution.
    a. Return to the CloudFront console.
    b. In the left navigation pane, choose **Distributions**.
    c. **Note**: The **Last modified** column might display *Deploying* for the distribution. The deployment will complete within about 5 minutes; however, you can proceed to the next step without waiting.
    d. Choose the link for the distribution ID.
    e. In the **Security** section, notice that the distribution now has an **AWS WAF** value.



CloudFront > Distributions > E1SSRQOZBWRVXT

access-identity-cafe-website Standard

Security - Web Application Firewall (WAF)

Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.

Enabled

cafe-website-office-only-during-dev (WAFv2)

CloudFront geographic restrictions

Date range: Last 24 hours

Rule actions: All

Granularity: 5 minutes | 1 hour | 1 day

Security trends for the specified time range

Requests processed by AWS WAF for this distribution.

All requests

-

Requests

No data available

There is no data available for the specified time range.

Top attack types

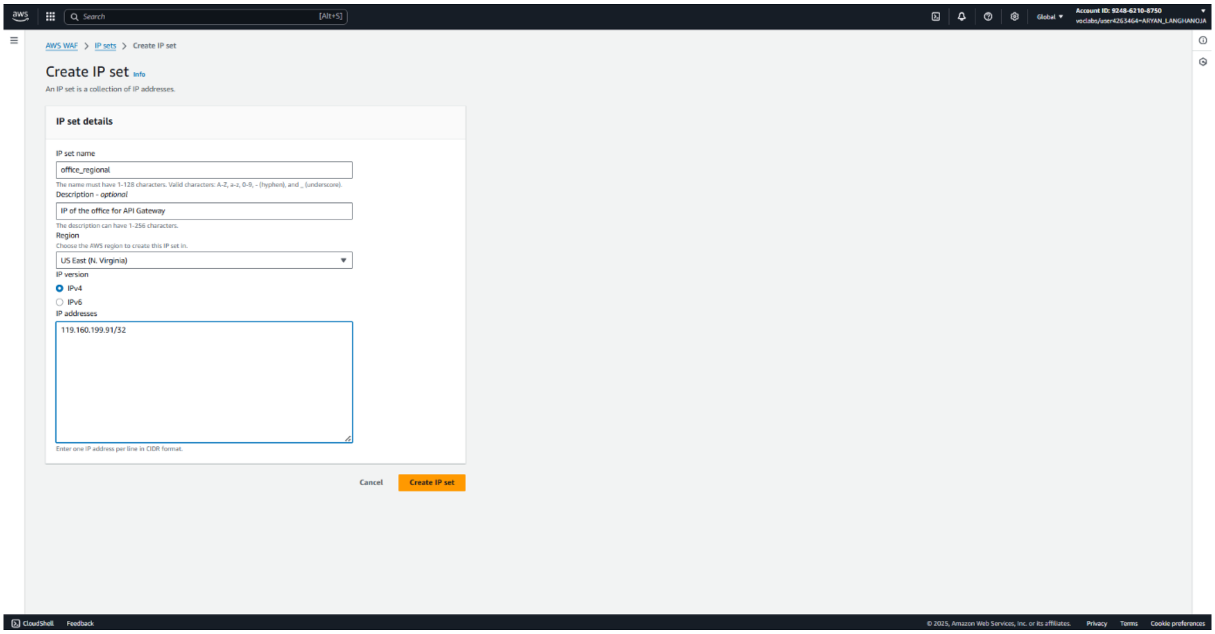
Top countries

Task 4: Securing a REST API endpoint using AWS WAF

22. Create a *regional* AWS WAF IP set.
    a. Return to the WAF & Shield console.
    b. In the left navigation pane, choose **IP sets**.
    c. Choose **Create IP set** and configure:
        i. **IP set name**: Enter office_regional
        ii. **Description**: Enter IP of the office for API Gateway
        iii. **Region**: Choose **US East (N. Virginia)**.
        iv. **IP addresses**: Enter <ip-address>/32 where <ip-address> is your public IPv4 address, as identified by whatismyipaddress.com.

Note: Be sure to include the /32 at the end of the IP address.

    d. Choose **Create IP set**.


aws
Search [Alt+S]
Global
Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA
AWS WAF > IP sets > Create IP set
Info
Cancel
Create IP set
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Create IP set

An IP set is a collection of IP addresses.

IP set details

IP set name: office_regional

The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - *optional*: IP of the office for API Gateway

The description can have 1-256 characters.

Region: Choose the AWS region to create this IP set in. US East (N. Virginia)

IP version

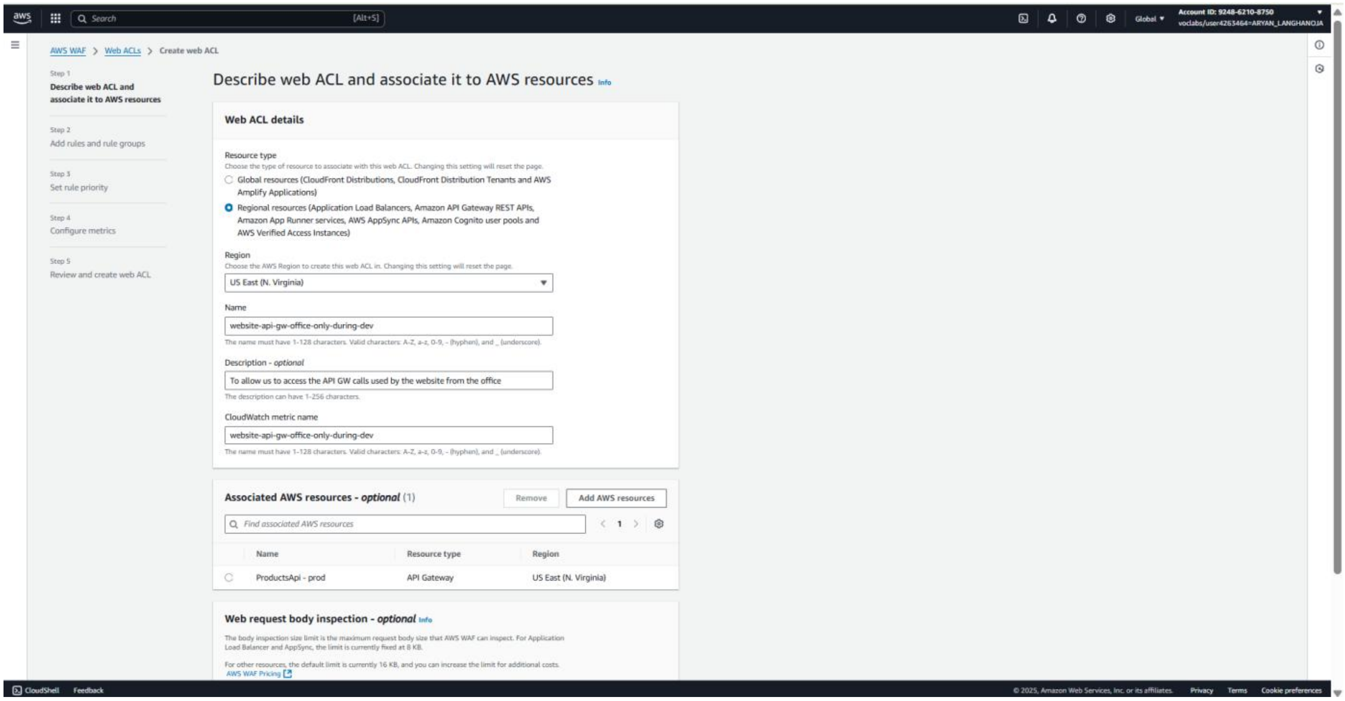
● IPv4

○ IPv6

IP addresses: 119.160.199.91/32

Enter one IP address per line in CIDR format.

23. Begin to create a *regional* web ACL.
    a. In the left navigation pane, choose **Web ACLs**.
    b. Choose **Create web ACL**.
    c. In the **Web ACL details** section, configure:
        i. **Resource type**: Choose **Regional resources**.
        ii. **Region**: Choose **US East (N. Virginia)**.
        iii. **Name**: Enter website-api-gw-office-only-during-dev
        iv. **Description**: Enter To allow us to access the API GW calls used by the website from the office
        v. **CloudWatch metric name**: Enter website-api-gw-office-only-during-dev
    d. In the **Associated AWS resources** section, configure:
        i. Choose **Add AWS resources**.
        ii. For Resource type, select **Amazon API Gateway**.
        iii. Select the **ProductsApi - prod** API Gateway resource.
        iv. Choose **Add**.
        v. Select the **ProductsApi - prod** resource again, and then choose **Next**.



AWS WAF > Web ACLs > Create web ACL

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

Describe web ACL and associate it to AWS resources

Web ACL details

Resource type
Choose the type of resource to associate with this web ACL. Changing this setting will reset the page.
○ Global resources (CloudFront Distributions, CloudFront Distribution Tenants and AWS Amplify Applications)
● Regional resources (Application Load Balancers, Amazon API Gateway REST APIs, Amazon App Runner services, AWS AppSync APIs, Amazon Cognito user pools and AWS Verified Access Instances)

Region
Choose the AWS Region to create this web ACL in. Changing this setting will reset the page.
US East (N. Virginia)

Name
website-api-gw-office-only-during-dev
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

Description - *optional*
To allow us to access the API GW calls used by the website from the office
The description can have 1-256 characters.

CloudWatch metric name
website-api-gw-office-only-during-dev
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

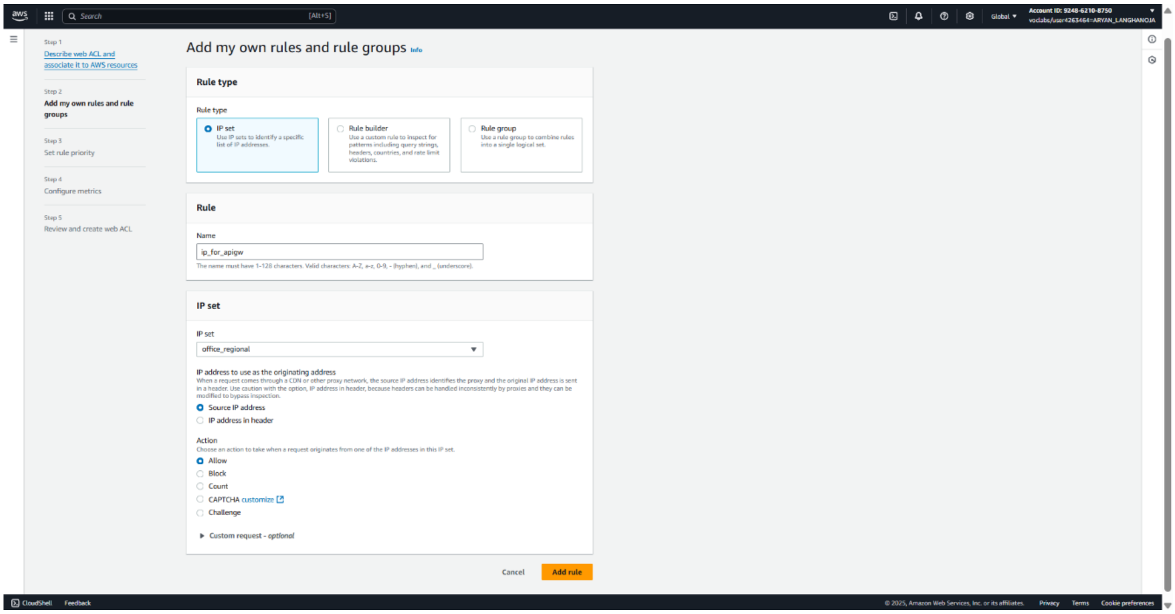
Associated AWS resources - *optional* (1)

	Name	Resource type	Region
○	ProductsApi - prod	API Gateway	US East (N. Virginia)

Web request body inspection - *optional*
The body inspection size limit is the maximum request body size that AWS WAF can inspect. For Application Load Balancer and AppSync, the limit is currently fixed at 8 KB.
For other resources, the default limit is currently 16 KB, and you can increase the limit for additional costs. AWS WAF Pricing

Search
[Alt+S]
Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA
Global
Info
Remove
Add AWS resources
Find associated AWS resources
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

24. Add a rule to the web ACL configuration to allow requests from API Gateway.
    a. In the **Rules** section, choose **Add rules**, **Add my own rules and rule groups**.
    b. **Rule type**: Choose **IP set**.
    c. **Name**: Enter ip_for_apigw
    d. **IP set**: Choose the *office_regional* IP set that you just created.
    e. **IP address to use as the originating address**: Keep the default **Source IP address** setting.
    f. **Action**: Choose **Allow**.
    g. Choose **Add rule**.



Step 1 Describe web ACL and associate it to AWS resources
Step 2 Add my own rules and rule groups
Step 3 Set rule priority
Step 4 Configure metrics
Step 5 Review and create web ACL

Add my own rules and rule groups

Rule type

Rule type
● IP set — Use IP sets to identify a specific list of IP addresses.
○ Rule builder — Use a custom rule to inspect for patterns including query strings, headers, countries, and rate limit violations.
○ Rule group — Use a rule group to combine rules into a single logical set.

Rule

Name
ip_for_apigw
The name must have 1-128 characters. Valid characters: A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

IP set

IP set
office_regional

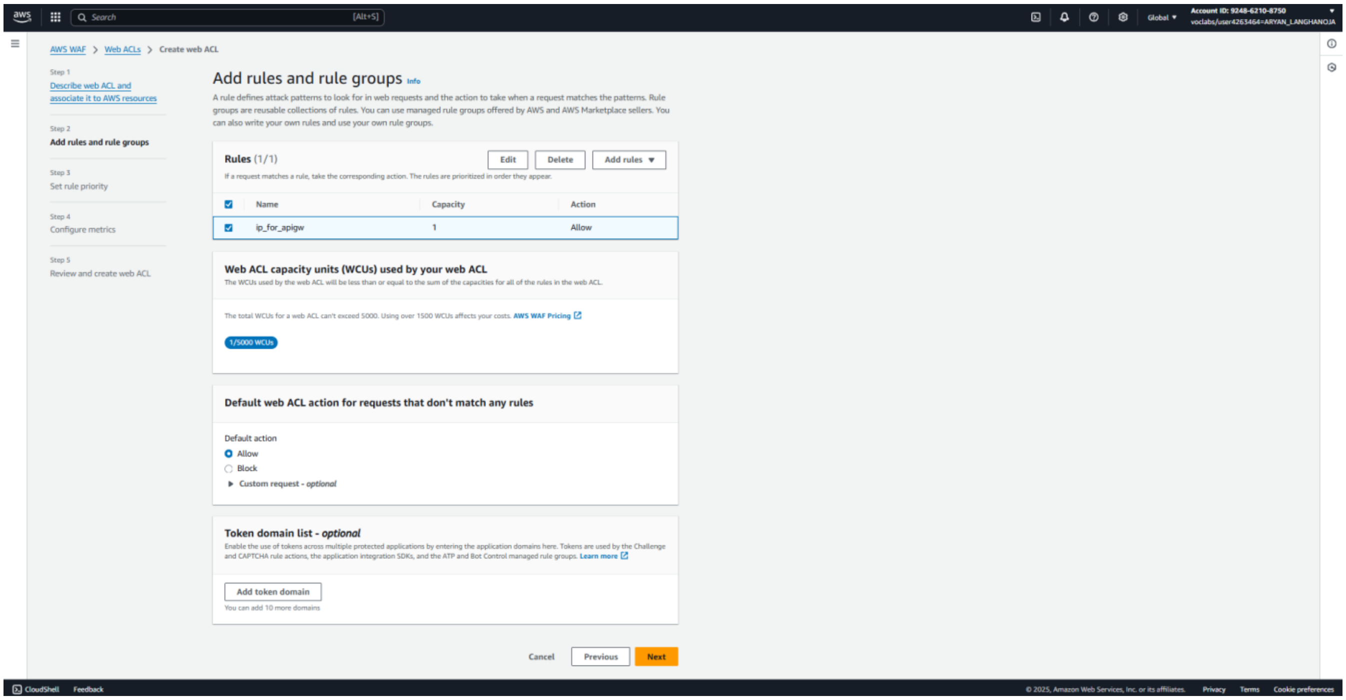
IP address to use as the originating address
When a request comes through a CDN or other proxy network, the source IP address identifies the proxy and the original IP address is sent in a header. Use caution with the option, IP address in header, because headers can be handled inconsistently by proxies and they can be modified to bypass inspection.
● Source IP address
○ IP address in header

Action
Choose an action to take when a request originates from one of the IP addresses in this IP set.
● Allow
○ Block
○ Count
○ CAPTCHA customize
○ Challenge

▶ Custom request - *optional*

Search
[Alt+S]
Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA
Global
Info
Cancel
Add rule
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

25. Update the new web ACL rule to block any requests that don't match the rule.
    a. In the **Rules** section, select the **ip_for_apigw** rule.
    b. In the **Default web ACL action** section, for **Default action**, choose **Block**.
    c. Choose **Next**.



Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA

AWS WAF > Web ACLs > Create web ACL

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

Add rules and rule groups Info

A rule defines attack patterns to look for in web requests and the action to take when a request matches the patterns. Rule groups are reusable collections of rules. You can use managed rule groups offered by AWS and AWS Marketplace sellers. You can also write your own rules and use your own rule groups.

Rules (1/1)

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

☑	Name	Capacity	Action
☑	ip_for_apigw	1	Allow

Web ACL capacity units (WCUs) used by your web ACL

The WCUs used by the web ACL will be less than or equal to the sum of the capacities for all of the rules in the web ACL.

The total WCUs for a web ACL can't exceed 5000. Using over 1500 WCUs affects your costs. AWS WAF Pricing

1/5000 WCUs

Default web ACL action for requests that don't match any rules

Default action
● Allow
○ Block
▸ Custom request - *optional*

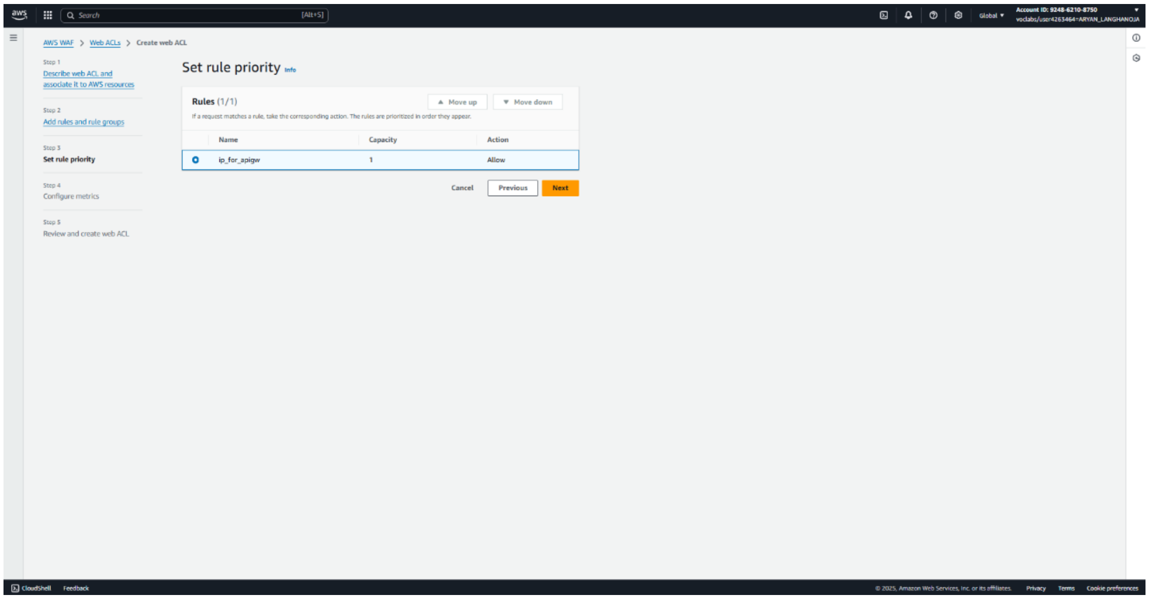
Token domain list - *optional*

Enable the use of tokens across multiple protected applications by entering the application domains here. Tokens are used by the Challenge and CAPTCHA rule actions, the application integration SDKs, and the ATP and Bot Control managed rule groups. Learn more

You can add 10 more domains

© 2025, Amazon Web Services, Inc. or its affiliates.

26. Set the rule priority, configure metrics, and create the web ACL.
    a. Choose the **ip_for_apigw** rule.
    b. Choose **Next**.
    c. Keep all of the default metrics settings, and choose **Next** again.
    d. Review the settings, and at the bottom of the page, choose **Create web ACL**.


aws
Search [Alt+S]
Account ID: 9248-6210-8750
voclabs/user4263464=ARYAN_LANGHANOJA
AWS WAF > Web ACLs > Create web ACL
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
Move up
Move down
Cancel
Previous
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

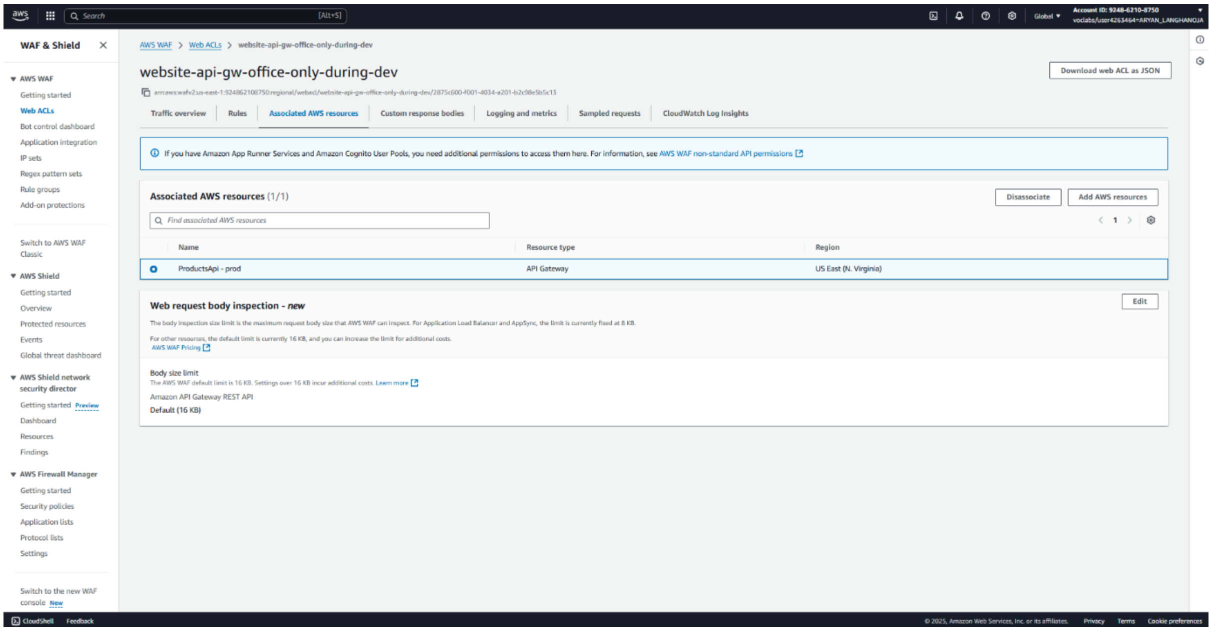
Set rule priority Info

Rules (1/1)

If a request matches a rule, take the corresponding action. The rules are prioritized in order they appear.

	Name	Capacity	Action
○	ip_for_apigw	1	Allow

27. When the web ACL creation process is complete, check the resources associated with the ACL.
    a. Choose the link for the *website-api-gw-office-only-during-dev* ACL, which you just created.
    b. Choose the **Associated AWS resources** tab.
    c. Confirm that the *ProductsApi - prod* resource is listed. If it isn't, add it:
        i. Choose **Add AWS resources**.
        ii. Choose the *ProductsApi - prod* resource.
        iii. Choose **Add**.



website-api-gw-office-only-during-dev

arn:aws:wafv2:us-east-1:924862108750:regional/webacl/website-api-gw-office-only-during-dev/2875c600-f001-4034-a201-b2c98e5b5c13

If you have Amazon App Runner Services and Amazon Cognito User Pools, you need additional permissions to access them here. For information, see AWS WAF non-standard API permissions

Associated AWS resources (1/1)

	Name	Resource type	Region
○	ProductsApi - prod	API Gateway	US East (N. Virginia)

Web request body inspection - *new*

The body inspection size limit is the maximum request body size that AWS WAF can inspect. For Application Load Balancer and AppSync, the limit is currently fixed at 8 KB.

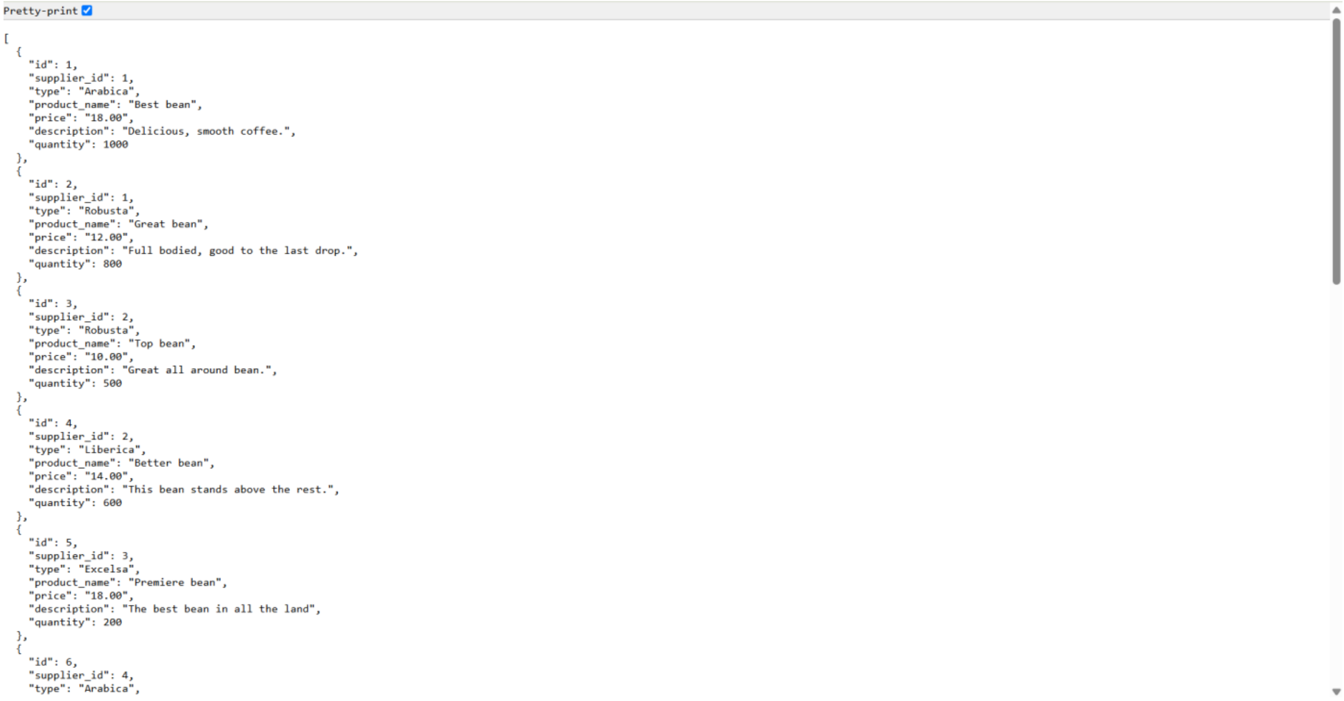
For other resources, the default limit is currently 16 KB, and you can increase the limit for additional costs.
AWS WAF Pricing

Body size limit
The AWS WAF default limit is 16 KB. Settings over 16 KB incur additional costs. Learn more

Amazon API Gateway REST API
Default (16 KB)

28. Test the new ACL from your computer.

- In a new browser tab, go to the API Gateway console.
- Choose the link for the **ProductsApi**.
- In the left navigation pane, choose **Stages**.
- In the **Stages** navigation pane, expand the **prod** stage.
- Under **/bean_products**, choose **GET**.
- Copy the **Invoke URL**, which has the format *https://.execute-api.us-east-1.amazonaws.com/prod/bean_products*, and load the URL in a new browser tab.
- A JSON-formatted document with product information displays. This is the expected behavior.

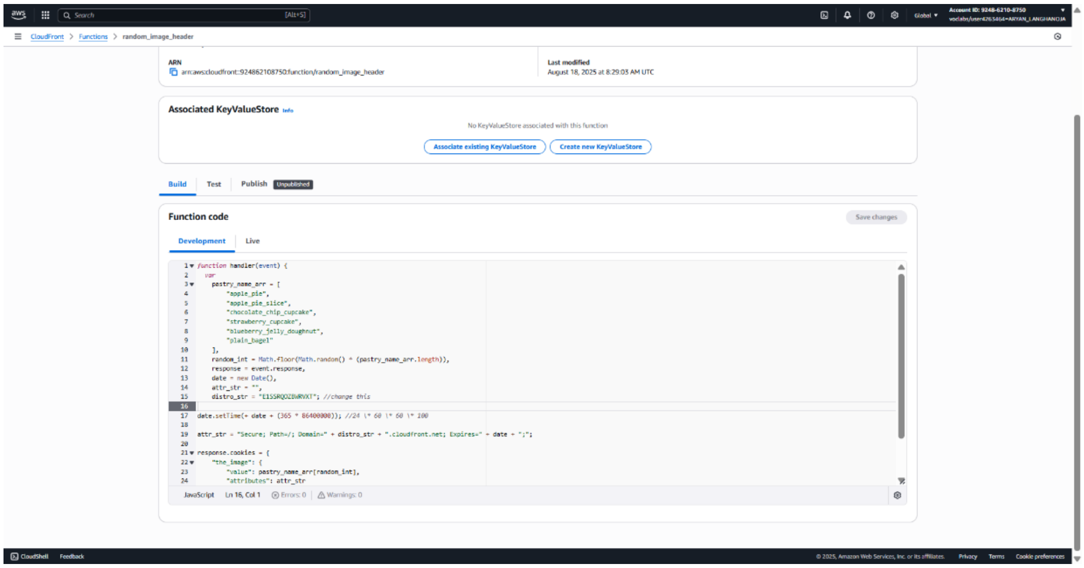


```
[
  {
    "id": 1,
    "supplier_id": 1,
    "type": "Arabica",
    "product_name": "Best bean",
    "price": "18.00",
    "description": "Delicious, smooth coffee.",
    "quantity": 1000
  },
  {
    "id": 2,
    "supplier_id": 1,
    "type": "Robusta",
    "product_name": "Great bean",
    "price": "12.00",
    "description": "Full bodied, good to the last drop.",
    "quantity": 800
  },
  {
    "id": 3,
    "supplier_id": 2,
    "type": "Robusta",
    "product_name": "Top bean",
    "price": "10.00",
    "description": "Great all around bean.",
    "quantity": 500
  },
  {
    "id": 4,
    "supplier_id": 2,
    "type": "Liberica",
    "product_name": "Better bean",
    "price": "14.00",
    "description": "This bean stands above the rest.",
    "quantity": 600
  },
  {
    "id": 5,
    "supplier_id": 3,
    "type": "Excelsa",
    "product_name": "Premiere bean",
    "price": "18.00",
    "description": "The best bean in all the land",
    "quantity": 200
  },
  {
    "id": 6,
    "supplier_id": 4,
    "type": "Arabica",
```

Task 5: Configuring a CloudFront function on the website

29. Create a CloudFront function.

- Navigate to the CloudFront console.
- In the left navigation pane, choose **Functions**.
- Choose **Create function**.
- For **Name**, enter `random_image_header`
- Choose **Create function**.



ARN
arn:aws:cloudfront::924862108750:function/random_image_header

Last modified
August 18, 2025 at 8:29:03 AM UTC

Associated KeyValueStore

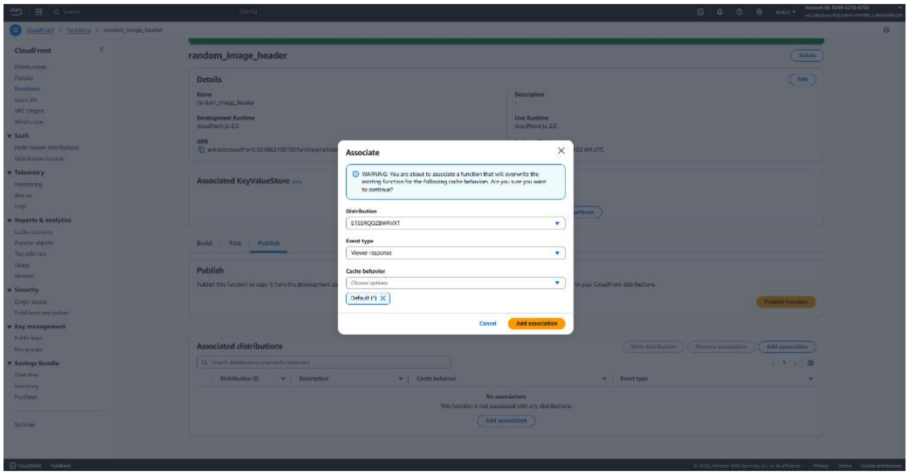
No KeyValueStore associated with this function

Function code

```
function handler(event) {
  var
    pastry_name_arr = [
        "apple_pie",
        "apple_pie_slice",
        "chocolate_chip_cupcake",
        "strawberry_cupcake",
        "blueberry_jelly_doughnut",
        "plain_bagel"
    ],
    random_int = Math.floor(Math.random() * (pastry_name_arr.length)),
    response = event.response,
    date = new Date(),
    attr_str = "",
    distro_str = "E1SSRQOZBWRVXT"; //change this

date.setTime(+ date + (365 * 86400000)); //24 \* 60 \* 60 \* 100

attr_str = "Secure; Path=/; Domain=" + distro_str + ".cloudfront.net; Expires=" + date + ";";

response.cookies = {
    "the_image": {
        "value": pastry_name_arr[random_int],
        "attributes": attr_str
```

30. Publish the CloudFront function, and associate it with the CloudFront distribution.

- Choose the **Publish** tab.
- Choose **Publish function**.
- A message at the top of the page indicates that the *random_image_header* function was successfully published.
- In the **Associated distributions** section, choose **Add association** and configure:
    - **Distribution**: Choose the distribution.
    - **Event type**: Choose **Viewer Response**.
    - **Cache behavior**: Select **Default (*)**.
    - Choose **Add association**.
- In the left navigation pane, choose **Distributions**.
- Notice that the distribution is being deployed again.



Associate

WARNING: You are about to associate a function that will overwrite the existing function for the following cache behaviors. Are you sure you want to continue?

Distribution: E1SSRQOZBWRVXT

Event type: Viewer response

Cache behavior: Default (*)

No associations

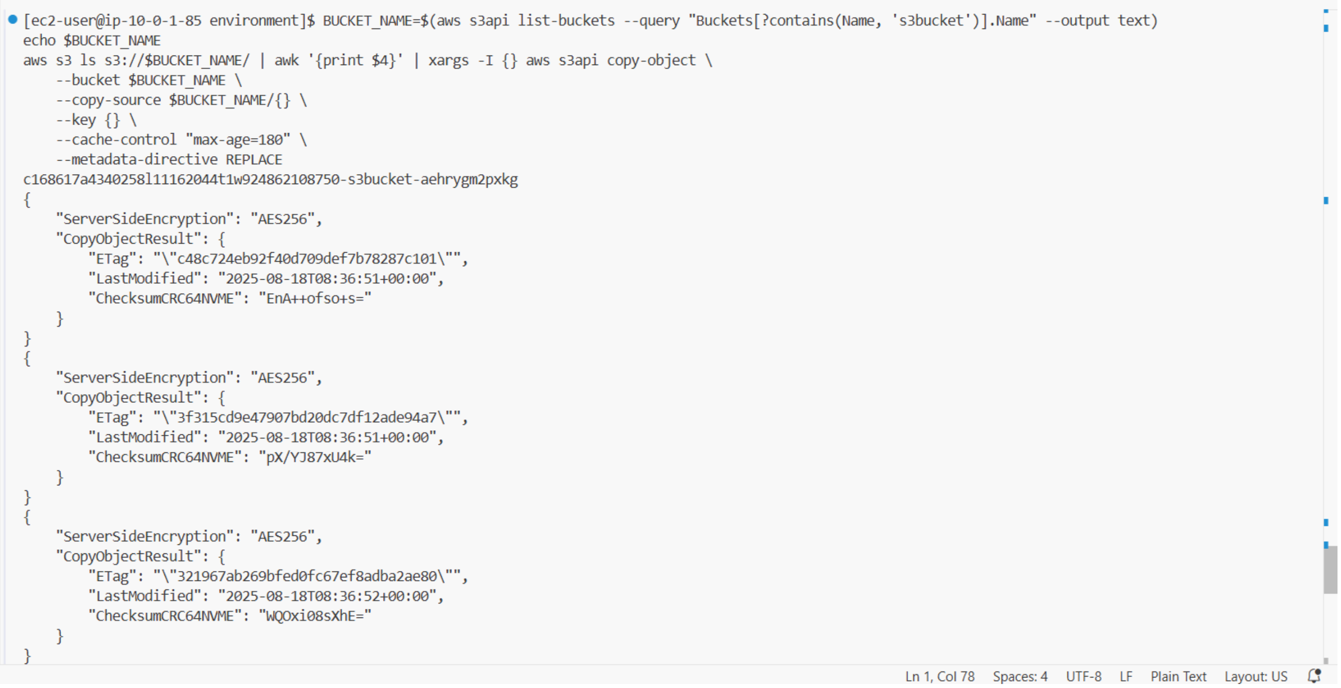
This function is not associated with any distributions.

Task 6: Adjusting the cache duration

31. Edit the Cache-Control header set on each object in the S3 bucket.

- Go to the VS Code IDE bash terminal and run the following command:

```
BUCKET_NAME=$(aws s3api list-buckets --query "Buckets[?contains(Name, 's3bucket')].Name" --output text)
echo $BUCKET_NAME
aws s3 ls s3://$BUCKET_NAME/ | awk '{print $4}' | xargs -I {} aws s3api copy-object \
    --bucket $BUCKET_NAME \
    --copy-source $BUCKET_NAME/{} \
    --key {} \
    --cache-control "max-age=180" \
    --metadata-directive REPLACE
```



```
[ec2-user@ip-10-0-1-85 environment]$ BUCKET_NAME=$(aws s3api list-buckets --query "Buckets[?contains(Name, 's3bucket')].Name" --output text)
echo $BUCKET_NAME
aws s3 ls s3://$BUCKET_NAME/ | awk '{print $4}' | xargs -I {} aws s3api copy-object \
    --bucket $BUCKET_NAME \
    --copy-source $BUCKET_NAME/{} \
    --key {} \
    --cache-control "max-age=180" \
    --metadata-directive REPLACE
c168617a4340258l11162044t1w924862108750-s3bucket-aehrygm2pxkg
{
    "ServerSideEncryption": "AES256",
    "CopyObjectResult": {
        "ETag": "\"c48c724eb92f40d709def7b78287c101\"",
        "LastModified": "2025-08-18T08:36:51+00:00",
        "ChecksumCRC64NVME": "EnA++ofso+s="
    }
}
{
    "ServerSideEncryption": "AES256",
    "CopyObjectResult": {
        "ETag": "\"3f315cd9e47907bd20dc7df12ade94a7\"",
        "LastModified": "2025-08-18T08:36:51+00:00",
        "ChecksumCRC64NVME": "pX/YJ87xU4k="
    }
}
{
    "ServerSideEncryption": "AES256",
    "CopyObjectResult": {
        "ETag": "\"321967ab269bfed0fc67ef8adba2ae80\"",
        "LastModified": "2025-08-18T08:36:52+00:00",
        "ChecksumCRC64NVME": "WQOxi08sXhE="
    }
}
```

Conclusion:-

- I connected to the AWS IDE and set up the lab environment.
- I downloaded the required files and upgraded the AWS CLI.
- I verified the S3 bucket and opened the café website.
- I created a CloudFront distribution for the website.
- I enabled HTTPS access using CloudFront.
- I updated the S3 bucket policy for secure access.
- I created an IP set in AWS WAF for my system IP.
- I created a Web ACL to allow only my IP to access the site.
- I secured the API Gateway endpoint using AWS WAF.
- I tested the API Gateway URL and verified it was working.
- I created and deployed a CloudFront function.
- I updated cache settings on S3 objects for better performance.