Serverless E-Commerce Order Processing System on AWS

Objective

Design and deploy a secure, scalable, and cost-effective e-commerce backend using serverless AWS services. Key capabilities include product listing, shopping cart, and order placement functionalities, protected by AWS WAF and delivered via CloudFront.

Services Used

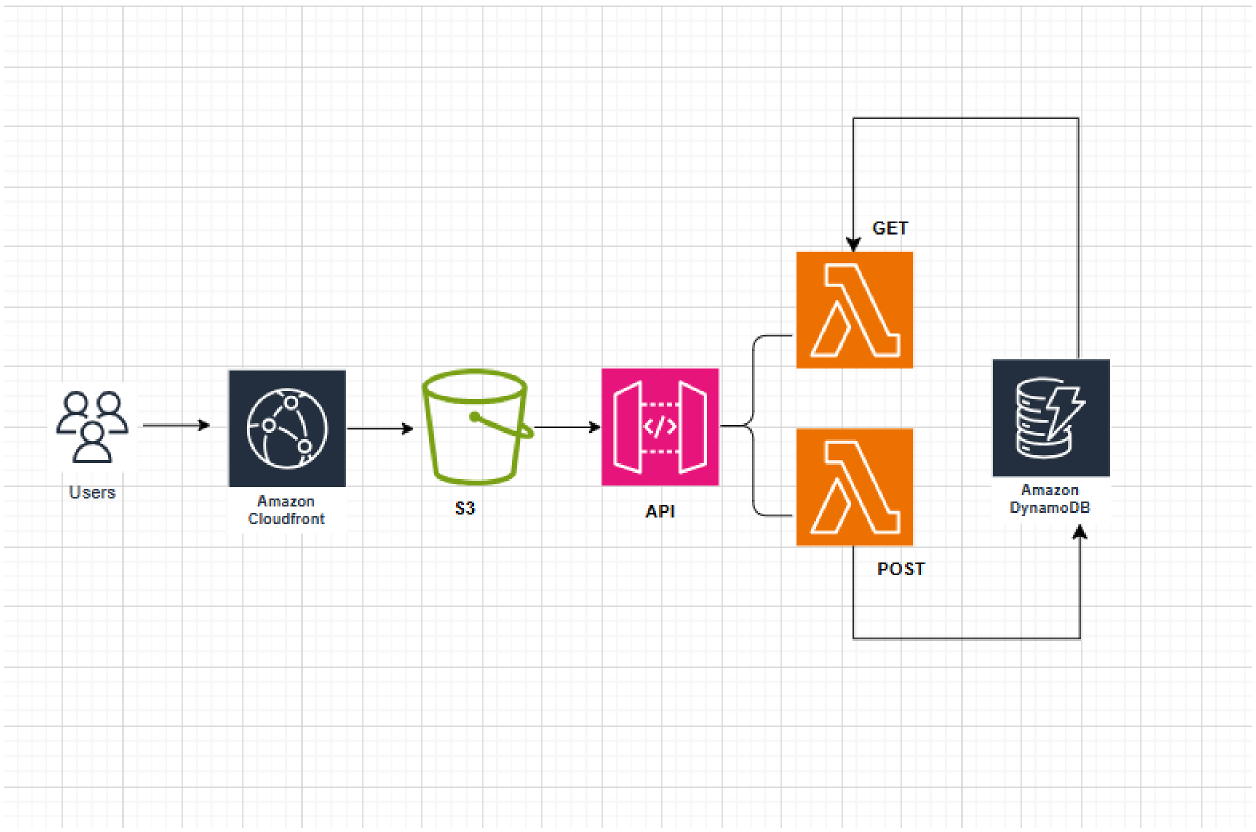
Amazon S3 + CloudFront

Amazon API Gateway

AWS Lambda

Amazon DynamoDB

Architecture Diagram



Step-by-Step Implementation

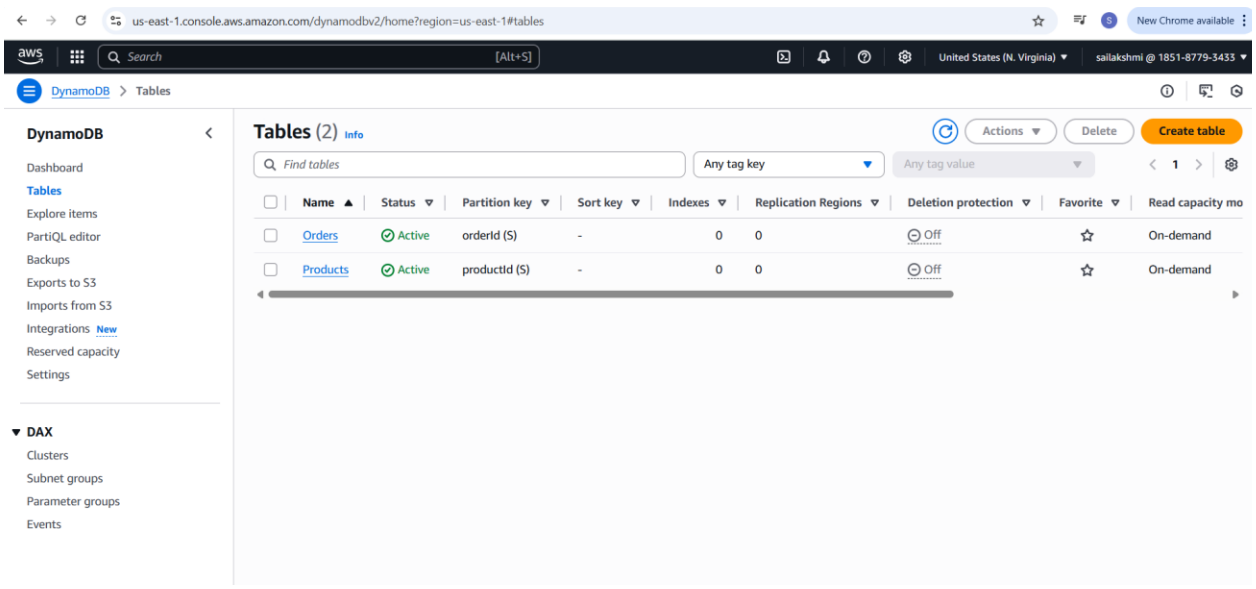
Step 1: Create DynamoDB Tables

Products Table

- Table name: Products
- Partition key: productId (String)

Orders Table

- Table name: Orders
- Partition key: orderId (String)



us-east-1.console.aws.amazon.com/dynamodbv2/home?region=us-east-1#tables

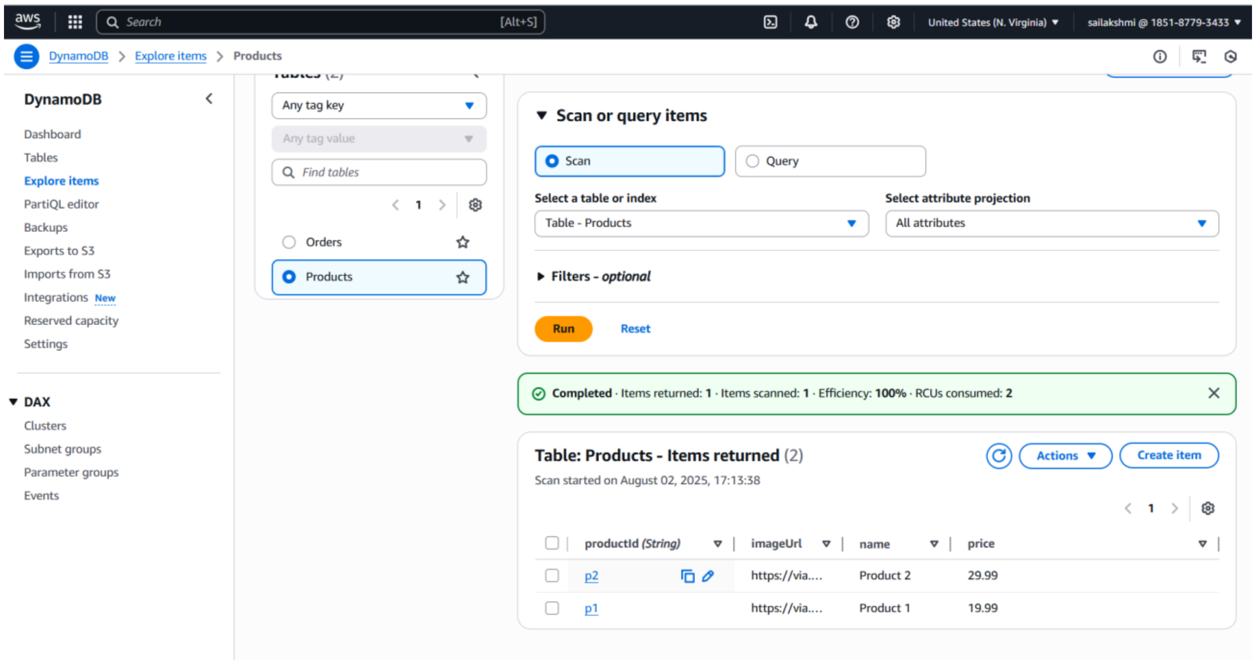
United States (N. Virginia) sailakshmi @ 1851-8779-3433

DynamoDB > Tables

Tables (2) Info

Name	Status	Partition key	Sort key	Indexes	Replication Regions	Deletion protection	Favorite	Read capacity mo
Orders	Active	orderId (S)	-	0	0	Off	☆	On-demand
Products	Active	productId (S)	-	0	0	Off	☆	On-demand

In products table add sample items


DynamoDB > Explore items > Products

Scan or query items

Scan [✓] Query

Select a table or index: Table - Products

Select attribute projection: All attributes

Filters - *optional*

Completed · Items returned: **1** · Items scanned: **1** · Efficiency: **100%** · RCUs consumed: **2**

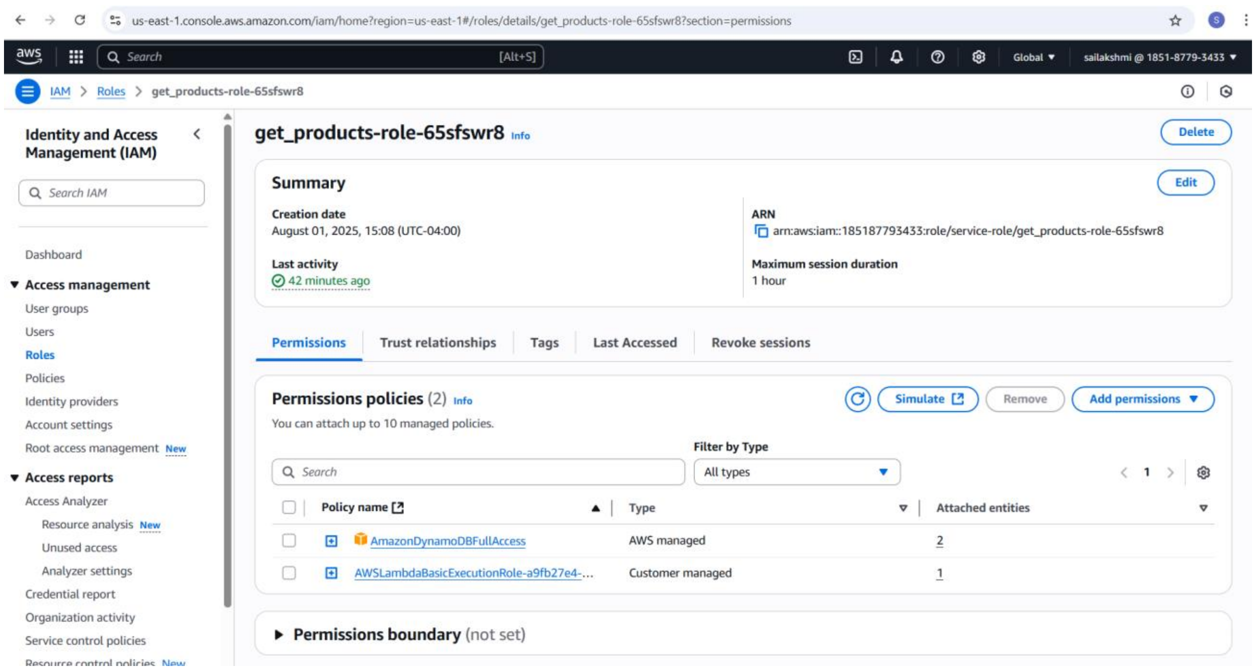
Table: Products - Items returned (2)

Scan started on August 02, 2025, 17:13:38

productId (String)	imageUrl	name	price
p2	https://via....	Product 2	29.99
p1	https://via....	Product 1	19.99

Step 2: Create IAM Role for Lambda

- Go to IAM → Roles → Create role
- Choose AWS service → Select Lambda
- Click Next: Permissions
- Attach policy AmazonDynamoDBFullAccess


us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/details/get_products-role-65sfswr8?section=permissions

IAM > Roles > get_products-role-65sfswr8

get_products-role-65sfswr8

Summary

Creation date: August 01, 2025, 15:08 (UTC-04:00)

ARN: arn:aws:iam::185187793433:role/service-role/get_products-role-65sfswr8

Last activity: 42 minutes ago

Maximum session duration: 1 hour

Permissions policies (2)

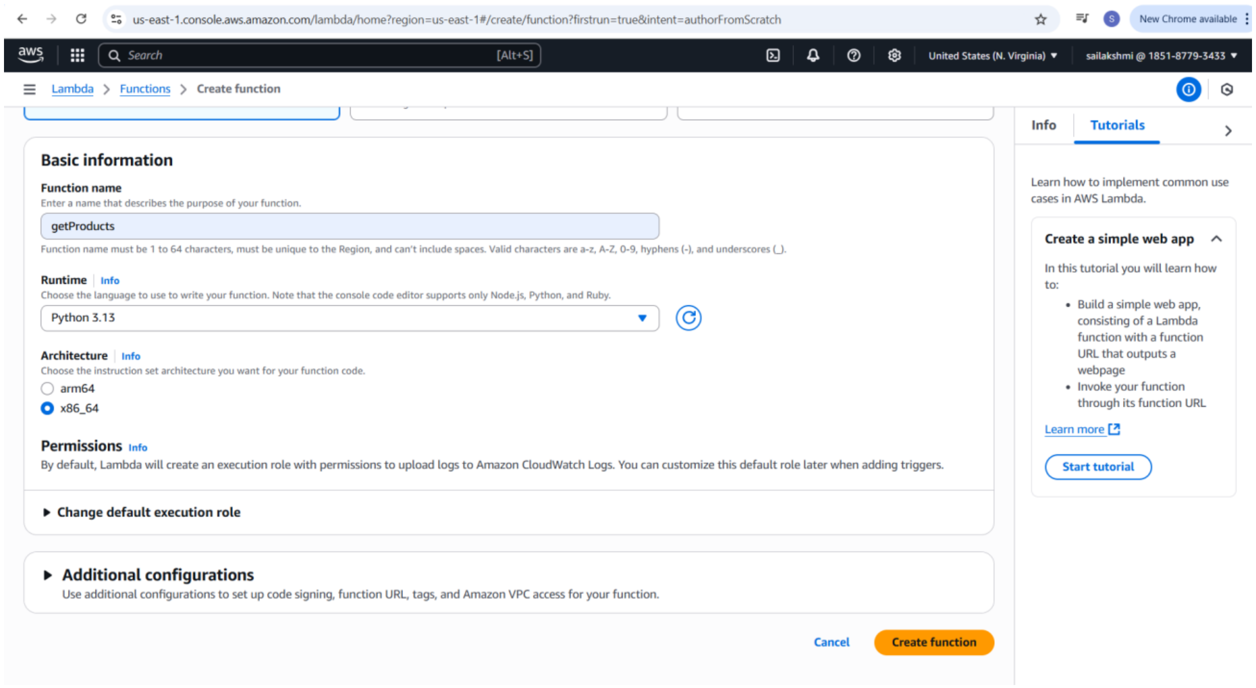
You can attach up to 10 managed policies.

Policy name	Type	Attached entities
AmazonDynamoDBFullAccess	AWS managed	2
AWSLambdaBasicExecutionRole-a9fb27e4-...	Customer managed	1

Permissions boundary (not set)

Step 3: Deploy Lambda Functions

Create a lambda functions for product



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/create/function?firstrun=true&intent=authorFromScratch

Lambda > Functions > Create function

Basic information

Function name
Enter a name that describes the purpose of your function.
getProducts
Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime Info
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.
Python 3.13

Architecture Info
Choose the instruction set architecture you want for your function code.
arm64 [✓ x86_64]

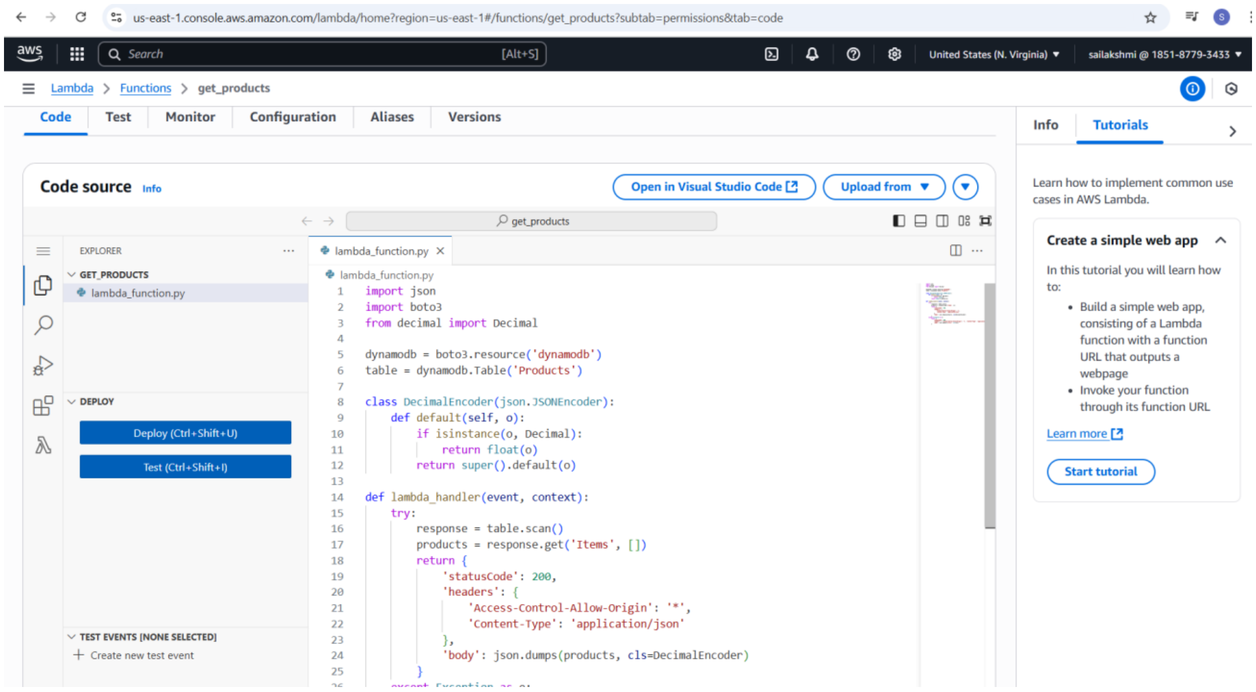
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▸ Change default execution role

▸ **Additional configurations**
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

Cancel | Create function

Info
Tutorials
Learn how to implement common use cases in AWS Lambda.
Create a simple web app
In this tutorial you will learn how to:
Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
Invoke your function through its function URL
Learn more
Start tutorial



us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/get_products?subtab=permissions&tab=code

Lambda > Functions > get_products

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

lambda_function.py

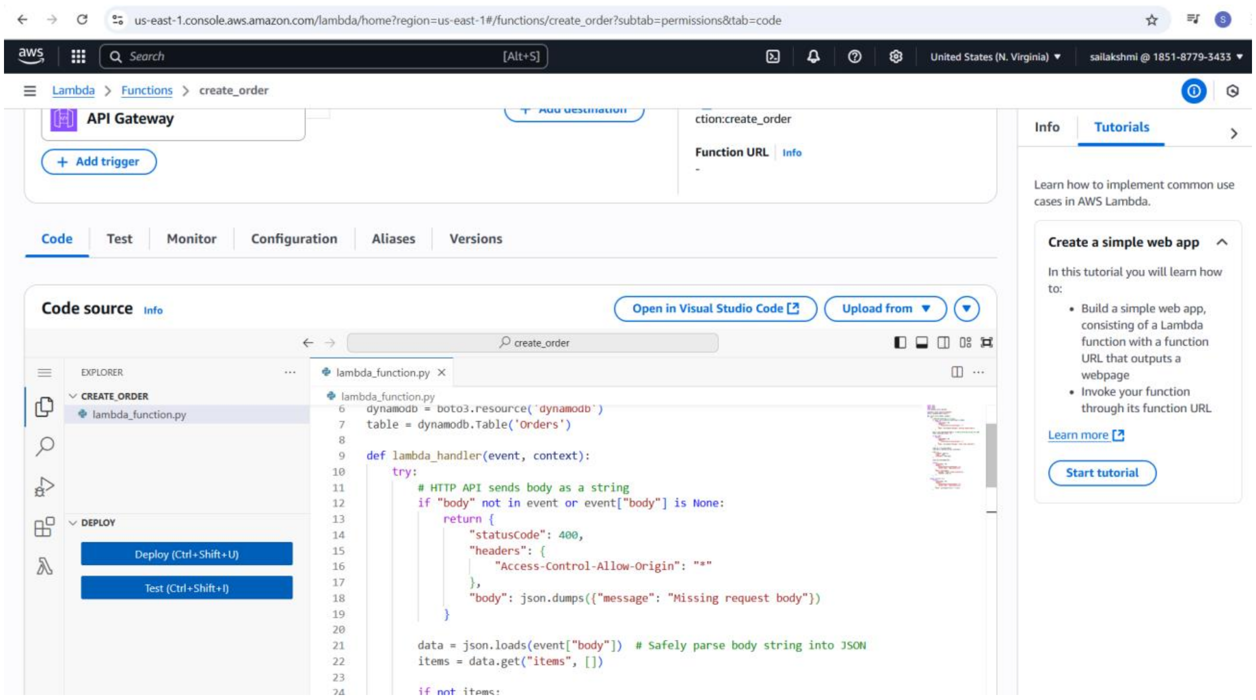
```python
import json
import boto3
from decimal import Decimal

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Products')

class DecimalEncoder(json.JSONEncoder):
    def default(self, o):
        if isinstance(o, Decimal):
            return float(o)
        return super().default(o)

def lambda_handler(event, context):
    try:
        response = table.scan()
        products = response.get('Items', [])
        return {
            'statusCode': 200,
            'headers': {
                'Access-Control-Allow-Origin': '*',
                'Content-Type': 'application/json'
            },
            'body': json.dumps(products, cls=DecimalEncoder)
        }
```

Create a second lambda function for order


us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions/create_order?subtab=permissions&tab=code

Lambda > Functions > create_order

API Gateway

Function URL Info

Code source Info

lambda_function.py

```
dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('Orders')

def lambda_handler(event, context):
    try:
        # HTTP API sends body as a string
        if "body" not in event or event["body"] is None:
            return {
                "statusCode": 400,
                "headers": {
                    "Access-Control-Allow-Origin": "*"
                },
                "body": json.dumps({"message": "Missing request body"})
            }

        data = json.loads(event["body"])  # Safely parse body string into JSON
        items = data.get("items", [])

        if not items:
```

Learn how to implement common use cases in AWS Lambda.

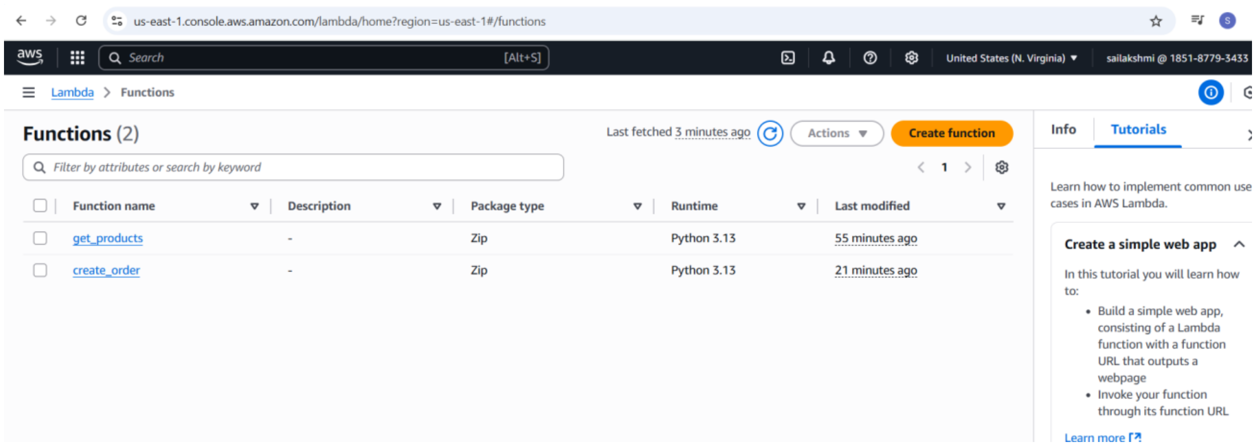
Create a simple web app

In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Start tutorial


us-east-1.console.aws.amazon.com/lambda/home?region=us-east-1#/functions

Lambda > Functions

Functions (2)

Last fetched 3 minutes ago

Function name	Description	Package type	Runtime	Last modified
get_products	-	Zip	Python 3.13	55 minutes ago
create_order	-	Zip	Python 3.13	21 minutes ago

Learn how to implement common use cases in AWS Lambda.

Create a simple web app

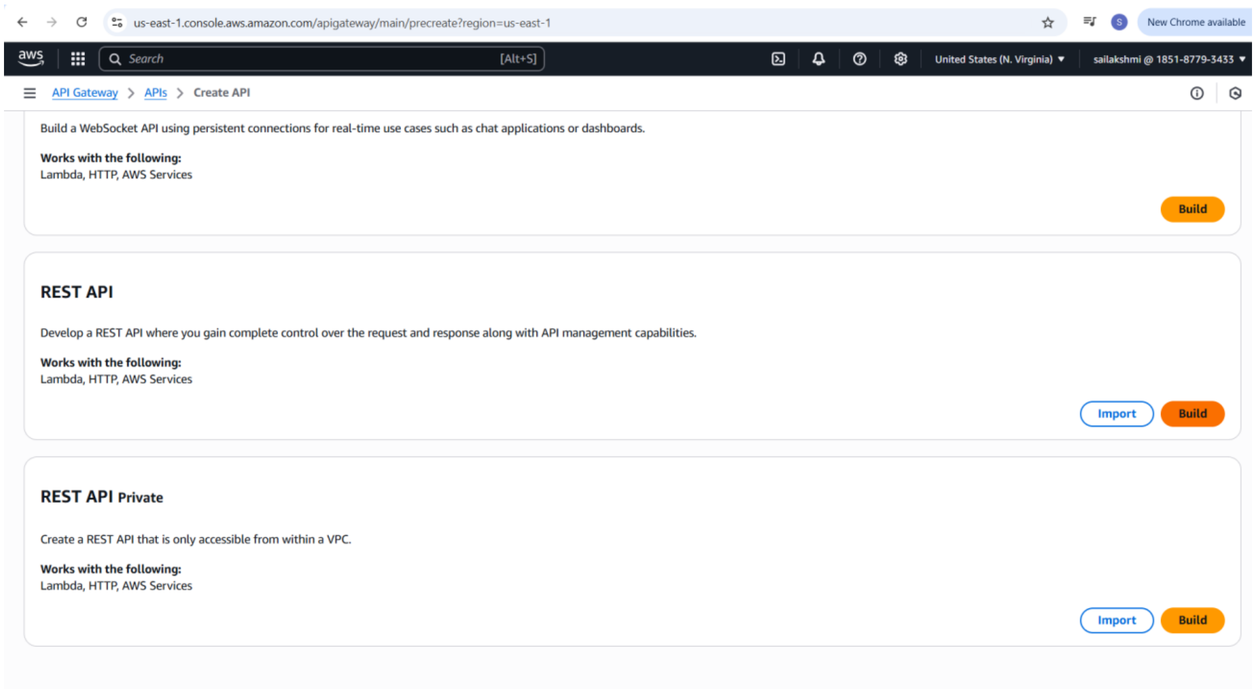
In this tutorial you will learn how to:

- Build a simple web app, consisting of a Lambda function with a function URL that outputs a webpage
- Invoke your function through its function URL

Learn more

Step 4: Set Up API Gateway

Create a HTTP API in API Gateway:


us-east-1.console.aws.amazon.com/apigateway/main/precreate?region=us-east-1

API Gateway > APIs > Create API

Build a WebSocket API using persistent connections for real-time use cases such as chat applications or dashboards.

Works with the following:
Lambda, HTTP, AWS Services

REST API

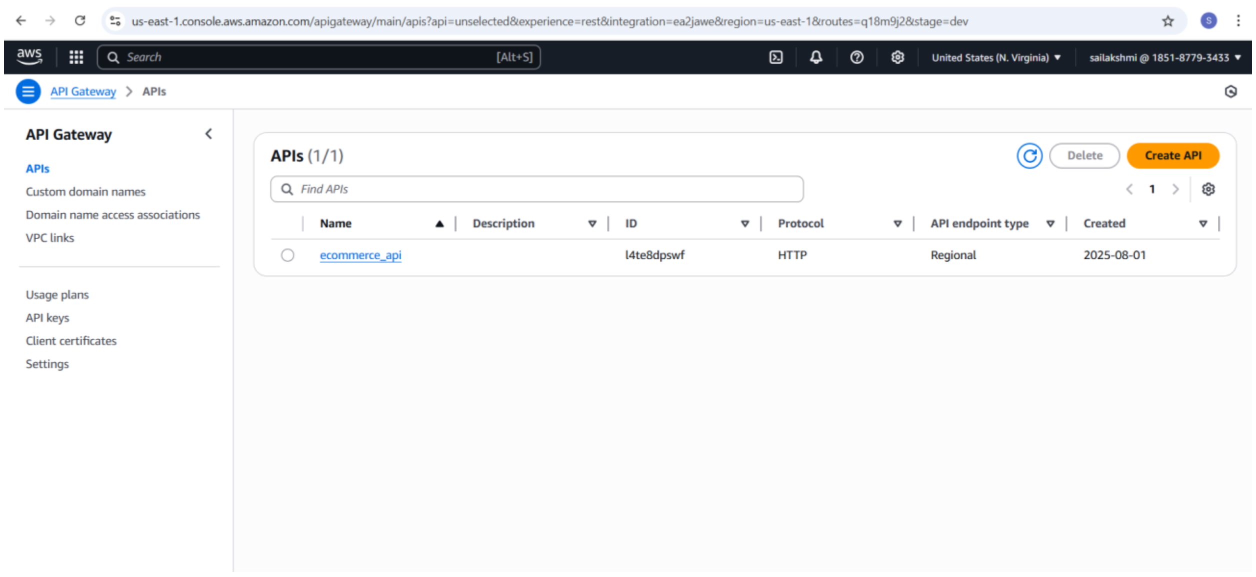
Develop a REST API where you gain complete control over the request and response along with API management capabilities.

Works with the following:
Lambda, HTTP, AWS Services

REST API Private

Create a REST API that is only accessible from within a VPC.

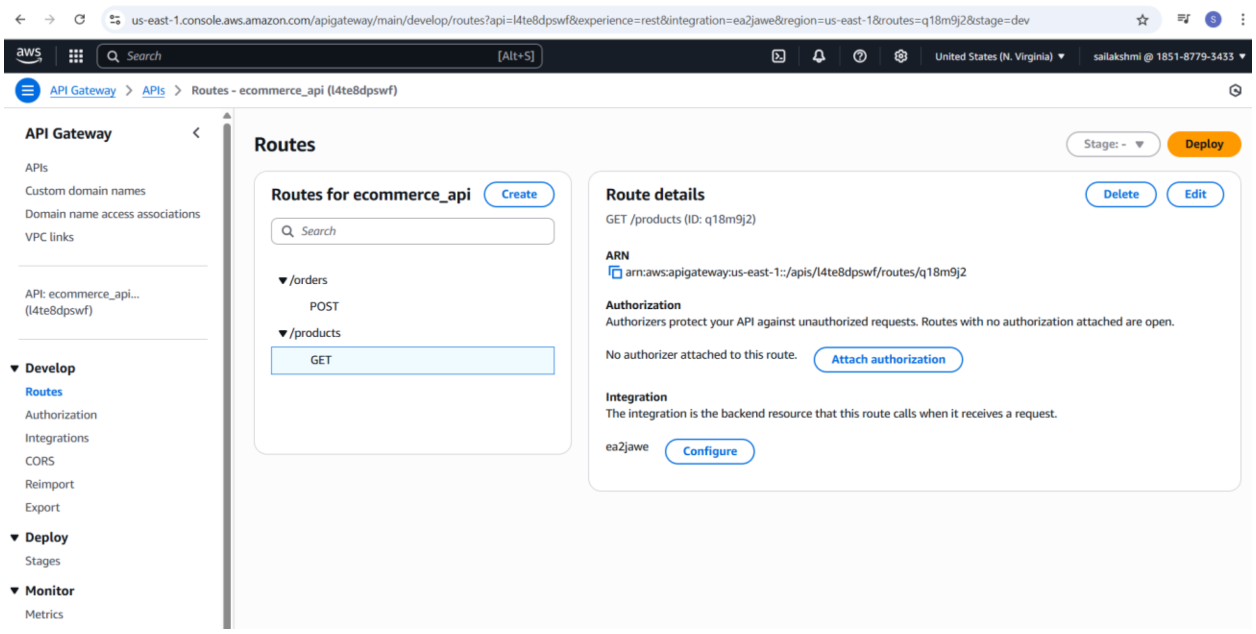
Works with the following:
Lambda, HTTP, AWS Services


us-east-1.console.aws.amazon.com/apigateway/main/apis?api=unselected&experience=rest&integration=ea2jawe&region=us-east-1&routes=q18m9j2&stage=dev

APIs (1/1)

Name	Description	ID	Protocol	API endpoint type	Created
ecommerce_api		l4te8dpswf	HTTP	Regional	2025-08-01

Create resources and methods (GET /products, POST /order).


us-east-1.console.aws.amazon.com/apigateway/main/develop/routes?api=l4te8dpswf&experience=rest&integration=ea2jawe&region=us-east-1&routes=q18m9j2&stage=dev
United States (N. Virginia)
sailakshmi @ 1851-8779-3433
API Gateway > APIs > Routes - ecommerce_api (l4te8dpswf)
API Gateway
APIs
Custom domain names
Domain name access associations
VPC links
API: ecommerce_api... (l4te8dpswf)
Develop
Routes
Authorization
Integrations
CORS
Reimport
Export
Deploy
Stages
Monitor
Metrics
Routes
Stage: -
Deploy
Routes for ecommerce_api
Create
Search
/orders
POST
/products
GET
Route details
Delete
Edit

GET /products (ID: q18m9j2)

ARN
arn:aws:apigateway:us-east-1::/apis/l4te8dpswf/routes/q18m9j2

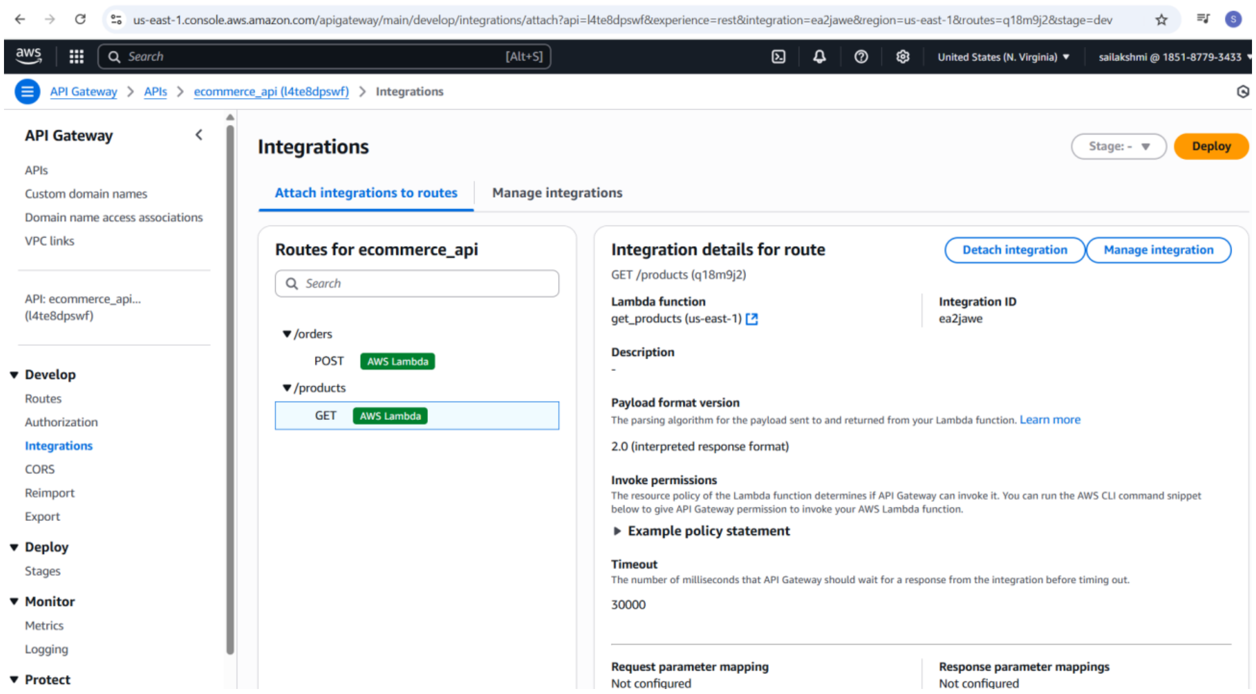
Authorization
Authorizers protect your API against unauthorized requests. Routes with no authorization attached are open.

No authorizer attached to this route. Attach authorization

Integration
The integration is the backend resource that this route calls when it receives a request.

ea2jawe Configure

Integrate each method with a respective Lambda function


us-east-1.console.aws.amazon.com/apigateway/main/develop/integrations/attach?api=l4te8dpswf&experience=rest&integration=ea2jawe&region=us-east-1&routes=q18m9j2&stage=dev
United States (N. Virginia)
sailakshmi @ 1851-8779-3433
API Gateway > APIs > ecommerce_api (l4te8dpswf) > Integrations
API Gateway
APIs
Custom domain names
Domain name access associations
VPC links
API: ecommerce_api... (l4te8dpswf)
Develop
Routes
Authorization
Integrations
CORS
Reimport
Export
Deploy
Stages
Monitor
Metrics
Logging
Protect
Integrations
Stage: -
Deploy
Attach integrations to routes
Manage integrations
Routes for ecommerce_api
Search
/orders
POST AWS Lambda
/products
GET AWS Lambda
Integration details for route
Detach integration
Manage integration
Example policy statement

GET /products (q18m9j2)

Lambda function
get_products (us-east-1)

Integration ID
ea2jawe

Description
-

Payload format version
The parsing algorithm for the payload sent to and returned from your Lambda function. Learn more

2.0 (interpreted response format)

Invoke permissions
The resource policy of the Lambda function determines if API Gateway can invoke it. You can run the AWS CLI command snippet below to give API Gateway permission to invoke your AWS Lambda function.

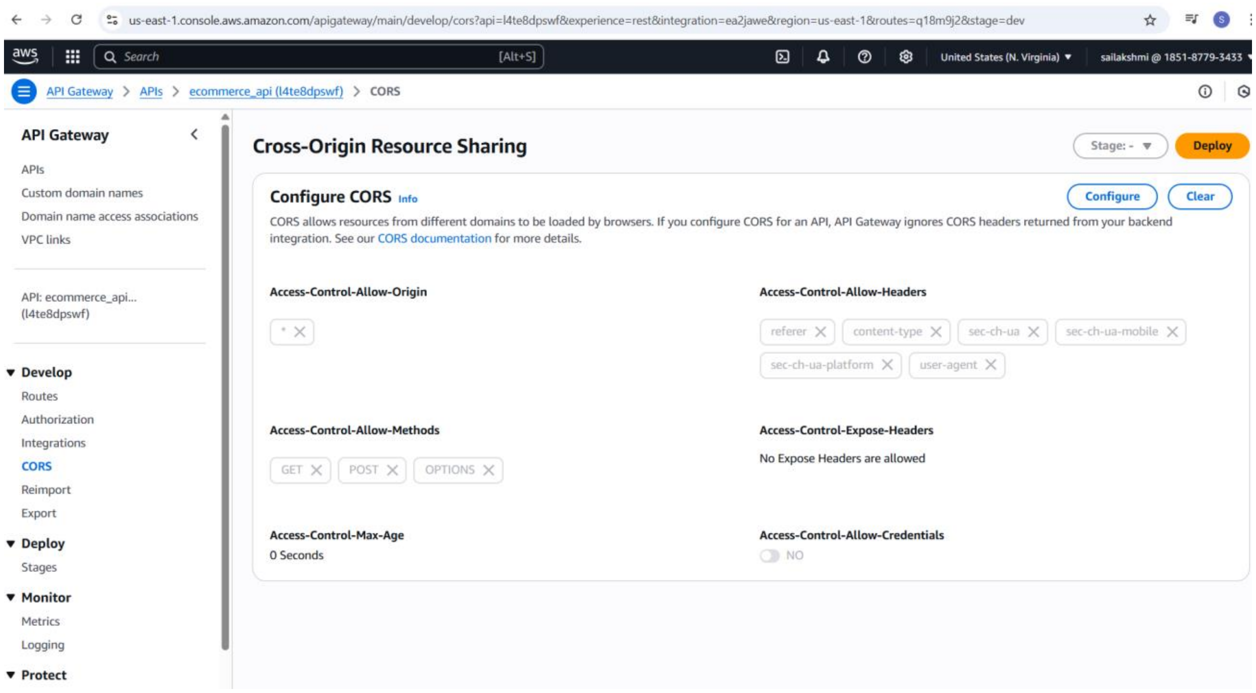
Timeout
The number of milliseconds that API Gateway should wait for a response from the integration before timing out.

30000

Request parameter mapping
Not configured

Response parameter mappings
Not configured

Enable CORS on all endpoints.


us-east-1.console.aws.amazon.com/apigateway/main/develop/cors?api=l4te8dpswf&experience=rest&integration=ea2jawe&region=us-east-1&routes=q18m9j2&stage=dev
Search [Alt+S]
United States (N. Virginia)
sailakshmi @ 1851-8779-3433
API Gateway > APIs > ecommerce_api (l4te8dpswf) > CORS
API Gateway
APIs
Custom domain names
Domain name access associations
VPC links
API: ecommerce_api... (l4te8dpswf)
Develop
Routes
Authorization
Integrations
CORS
Reimport
Export
Deploy
Stages
Monitor
Metrics
Logging
Protect
Stage: -
Deploy
Configure
Clear

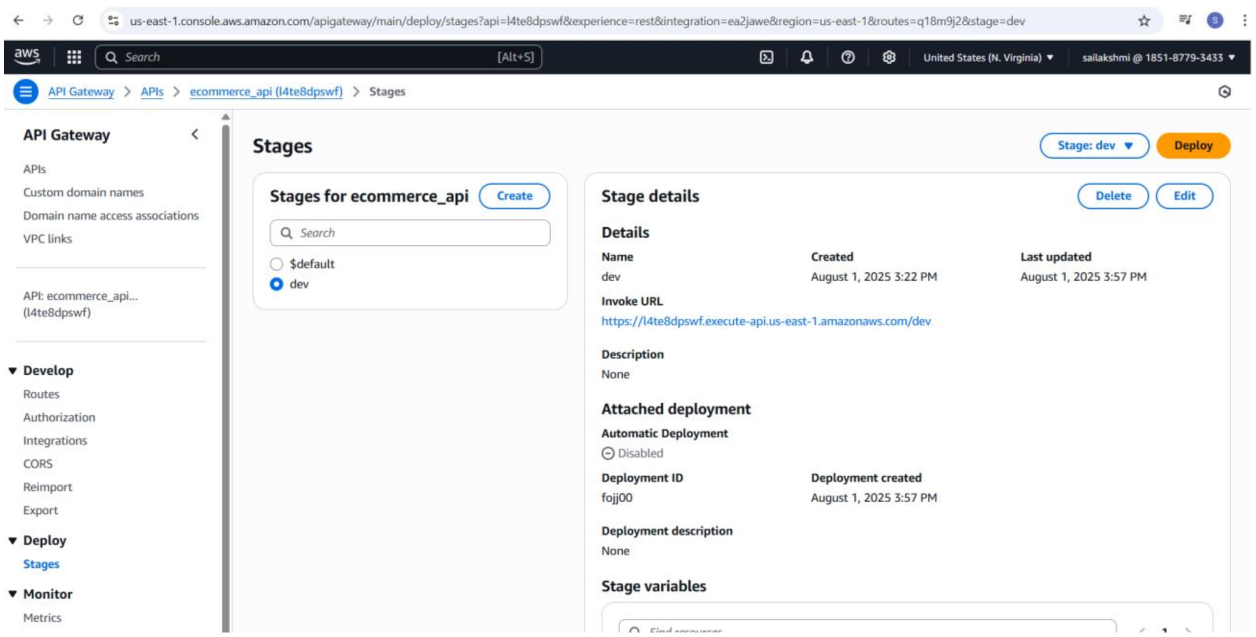
Cross-Origin Resource Sharing

Configure CORS Info

CORS allows resources from different domains to be loaded by browsers. If you configure CORS for an API, API Gateway ignores CORS headers returned from your backend integration. See our CORS documentation for more details.

Access-Control-Allow-Origin	*
Access-Control-Allow-Headers	referer, content-type, sec-ch-ua, sec-ch-ua-mobile, sec-ch-ua-platform, user-agent
Access-Control-Allow-Methods	GET, POST, OPTIONS
Access-Control-Expose-Headers	No Expose Headers are allowed
Access-Control-Max-Age	0 Seconds
Access-Control-Allow-Credentials	NO

Create a stage and deploy


us-east-1.console.aws.amazon.com/apigateway/main/deploy/stages?api=l4te8dpswf&experience=rest&integration=ea2jawe&region=us-east-1&routes=q18m9j2&stage=dev
Search [Alt+S]
United States (N. Virginia)
sailakshmi @ 1851-8779-3433
API Gateway > APIs > ecommerce_api (l4te8dpswf) > Stages
API Gateway
APIs
Custom domain names
Domain name access associations
VPC links
API: ecommerce_api... (l4te8dpswf)
Develop
Routes
Authorization
Integrations
CORS
Reimport
Export
Deploy
Stages
Monitor
Metrics
Stage: dev
Deploy
Create
Delete
Edit
Search

Stages

Stages for ecommerce_api

- $default
- dev [✓]

Stage details

Details

Name	Created	Last updated
dev	August 1, 2025 3:22 PM	August 1, 2025 3:57 PM

Invoke URL
https://l4te8dpswf.execute-api.us-east-1.amazonaws.com/dev

Description
None

Attached deployment

Automatic Deployment
Disabled

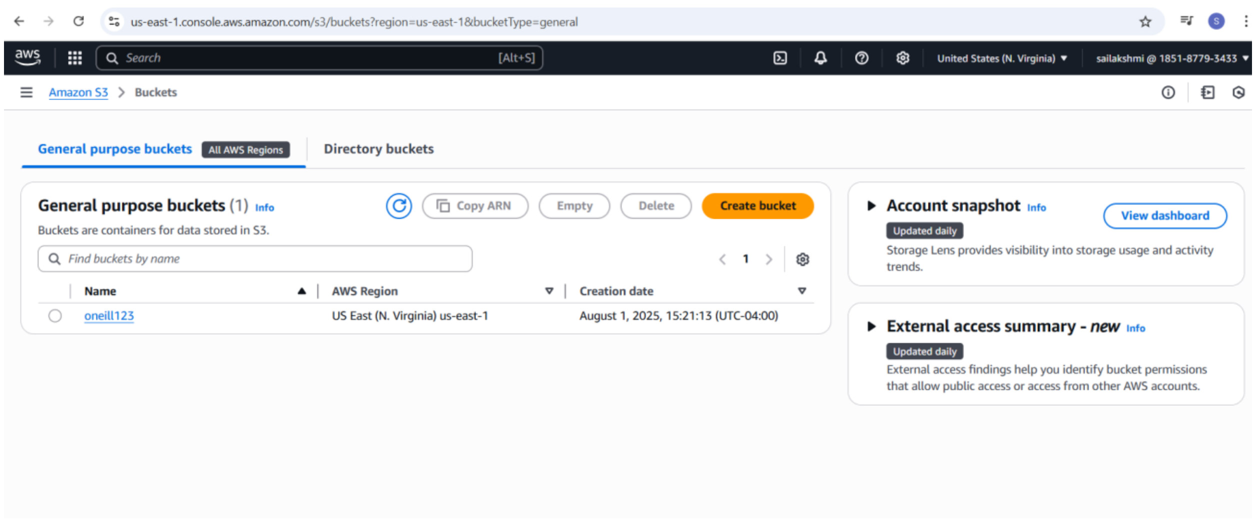
Deployment ID	Deployment created
fojj00	August 1, 2025 3:57 PM

Deployment description
None

Stage variables

Step 5: Frontend Deployment (S3 + CloudFront + WAF)

- Create S3 Bucket for Static Website
- Enable static website hosting.
- Upload your index.html file


us-east-1.console.aws.amazon.com/s3/buckets?region=us-east-1&bucketType=general

Amazon S3 > Buckets

General purpose buckets (All AWS Regions) | Directory buckets

General purpose buckets (1) Info

Buckets are containers for data stored in S3.

Copy ARN | Empty | Delete | Create bucket

Name	AWS Region	Creation date
oneill123	US East (N. Virginia) us-east-1	August 1, 2025, 15:21:13 (UTC-04:00)

Account snapshot Info — View dashboard

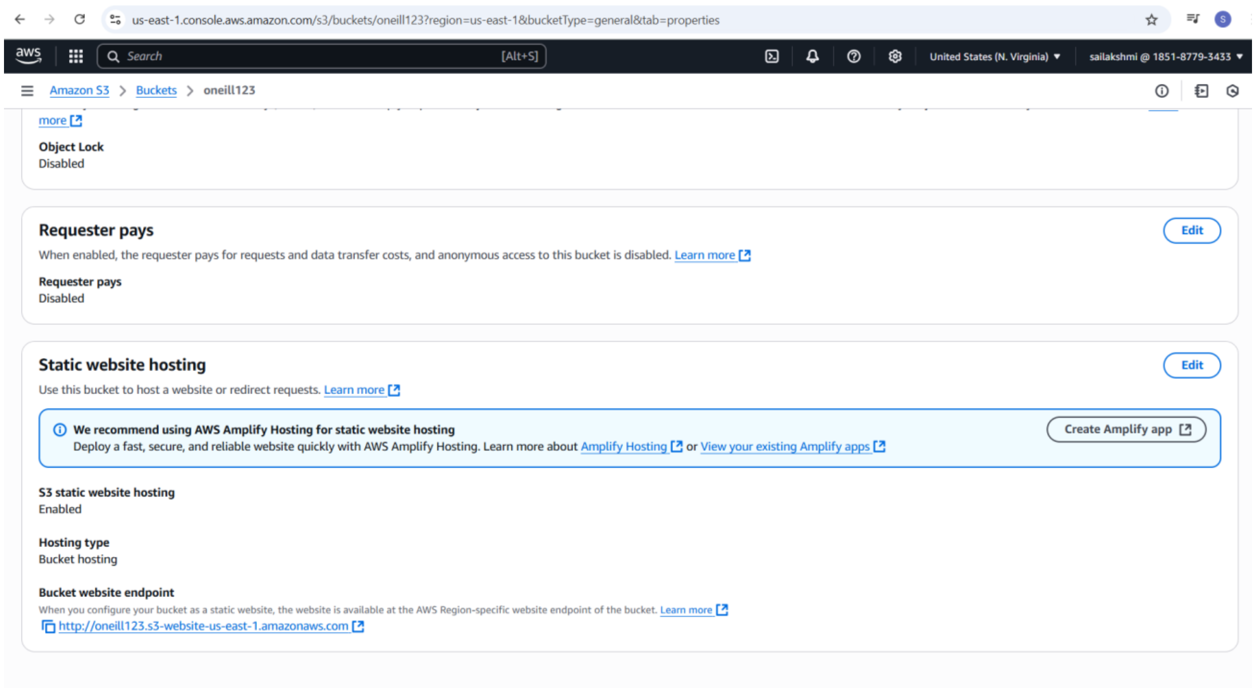
Updated daily

Storage Lens provides visibility into storage usage and activity trends.

External access summary - *new* Info

Updated daily

External access findings help you identify bucket permissions that allow public access or access from other AWS accounts.


us-east-1.console.aws.amazon.com/s3/buckets/oneill123?region=us-east-1&bucketType=general&tab=properties

Amazon S3 > Buckets > oneill123

more

Object Lock
Disabled

Requester pays — Edit

When enabled, the requester pays for requests and data transfer costs, and anonymous access to this bucket is disabled. Learn more

Requester pays
Disabled

Static website hosting — Edit

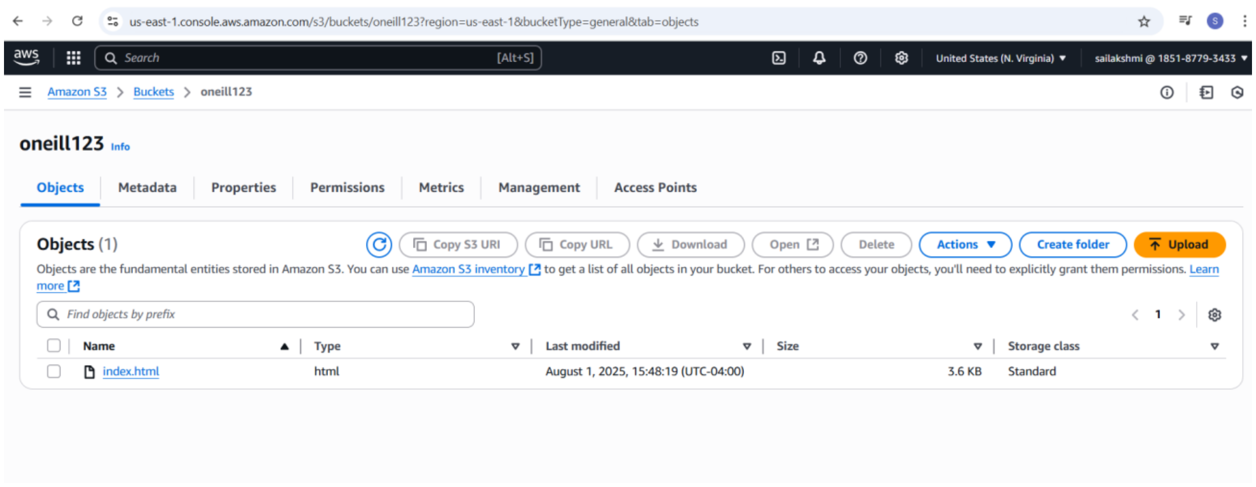
Use this bucket to host a website or redirect requests. Learn more

We recommend using AWS Amplify Hosting for static website hosting
Deploy a fast, secure, and reliable website quickly with AWS Amplify Hosting. Learn more about Amplify Hosting or View your existing Amplify apps — Create Amplify app

S3 static website hosting
Enabled

Hosting type
Bucket hosting

Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://oneill123.s3-website-us-east-1.amazonaws.com


us-east-1.console.aws.amazon.com/s3/buckets/oneill123?region=us-east-1&bucketType=general&tab=objects

Amazon S3 > Buckets > oneill123

oneill123 Info

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

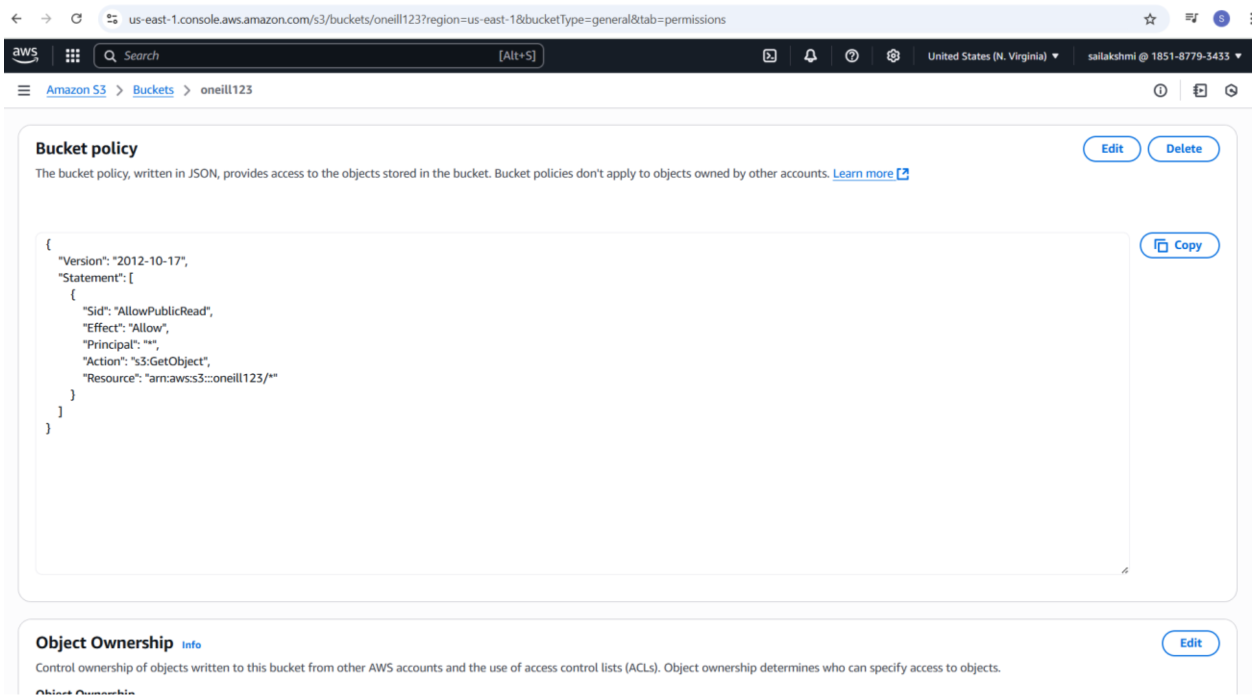
Objects (1)

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

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

Name	Type	Last modified	Size	Storage class
index.html	html	August 1, 2025, 15:48:19 (UTC-04:00)	3.6 KB	Standard

Bucket policy


us-east-1.console.aws.amazon.com/s3/buckets/oneill123?region=us-east-1&bucketType=general&tab=permissions
Search
[Alt+S]
United States (N. Virginia)
sailakshmi @ 1851-8779-3433
Amazon S3
Buckets
oneill123
Edit
Delete
Copy
Info
Edit

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

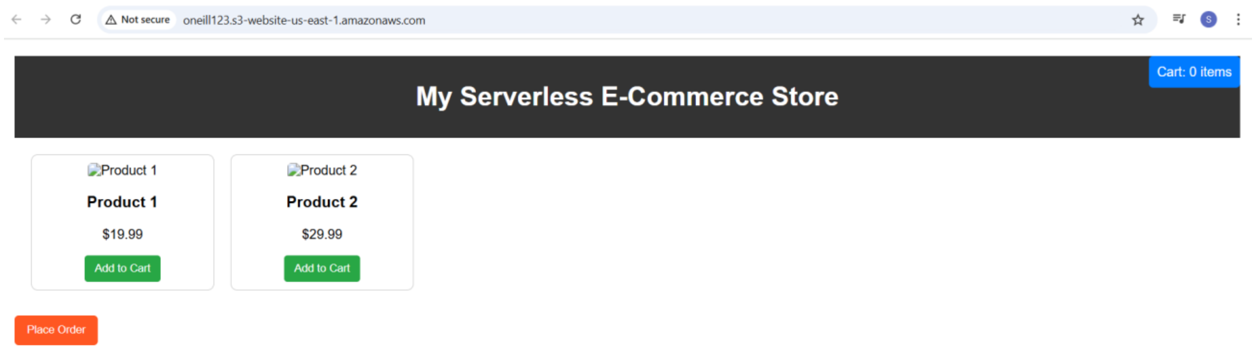
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::oneill123/*"
    }
  ]
}
```

Object Ownership

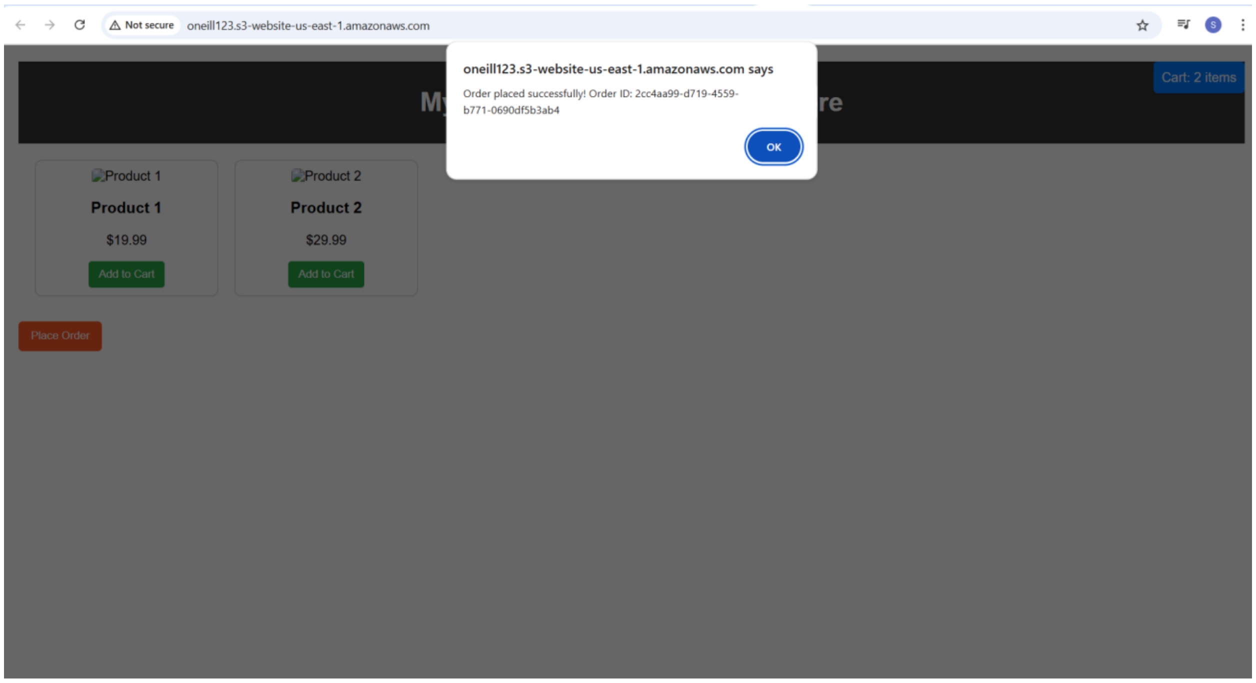
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

Step 6: Test Your Application

- Open the S3 website URL in a browser.
- Products should load dynamically.
- Add items to cart.
- Click Place Order to send order to backend.
- Check DynamoDB Orders table to see placed orders.



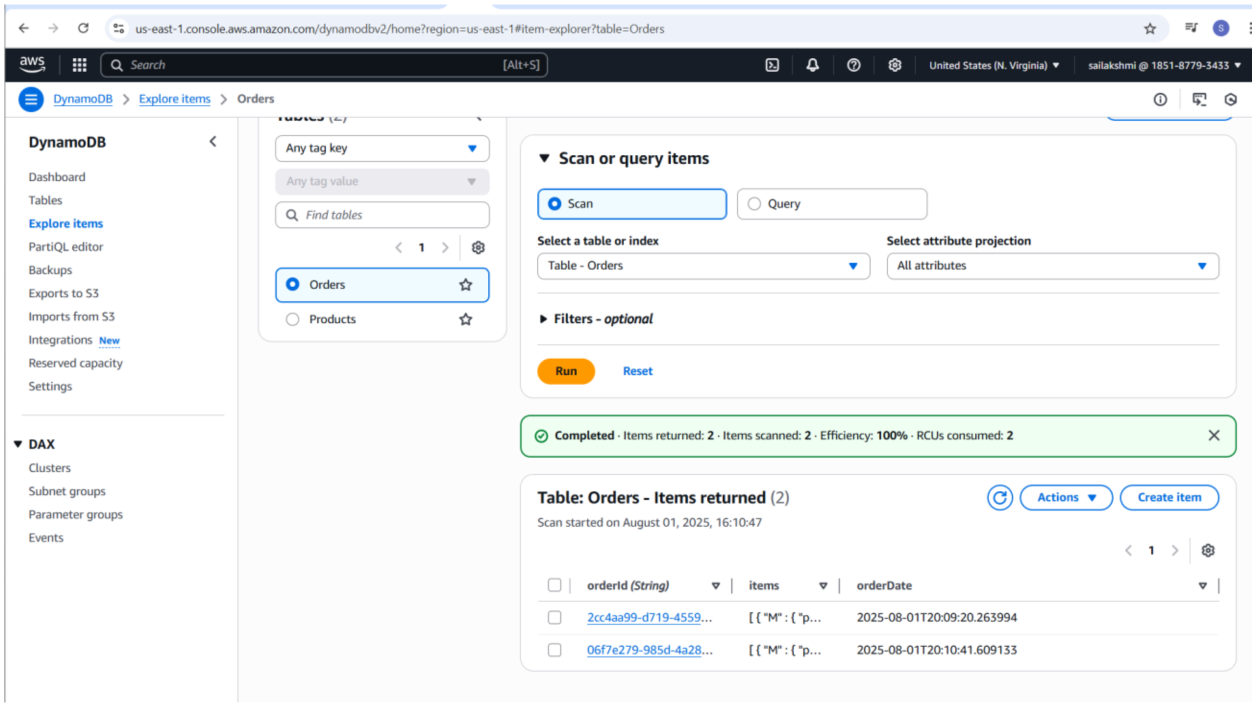
When u add the items the cart count will increase and click on place order



oneill123.s3-website-us-east-1.amazonaws.com says

Order placed successfully! Order ID: 2cc4aa99-d719-4559-b771-0690df5b3ab4

Check in DynamoDB



Select a table or index: Table - Orders

Select attribute projection: All attributes

Filters - *optional*

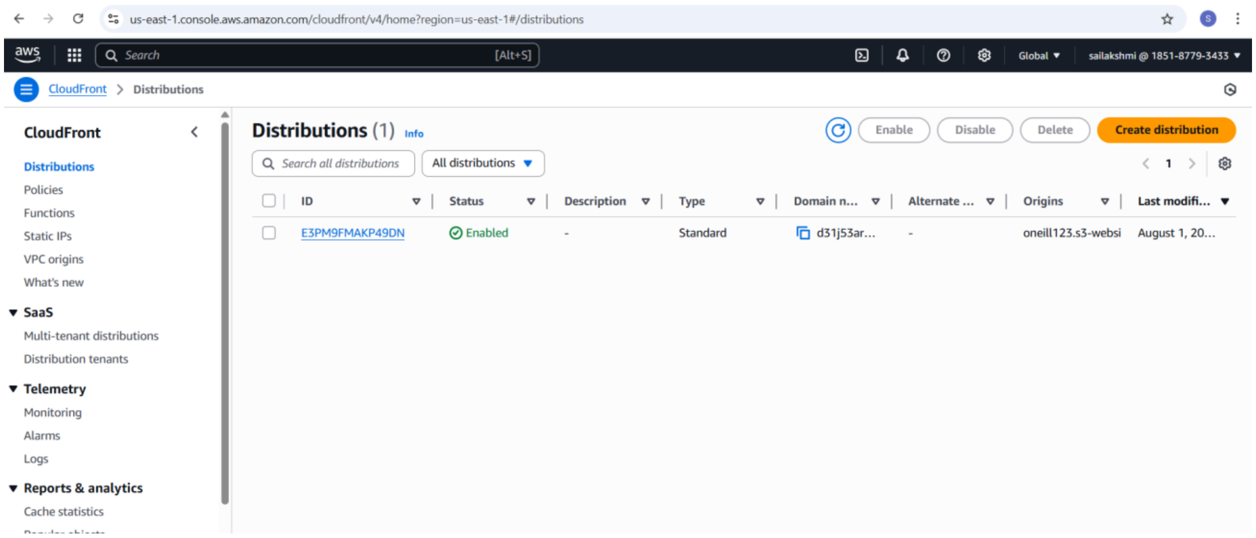
Completed · Items returned: 2 · Items scanned: 2 · Efficiency: 100% · RCUs consumed: 2

Table: Orders - Items returned (2)

Scan started on August 01, 2025, 16:10:47

orderId (String)	items	orderDate
2cc4aa99-d719-4559…	[{ "M" : { "p…	2025-08-01T20:09:20.263994
06f7e279-985d-4a28…	[{ "M" : { "p…	2025-08-01T20:10:41.609133

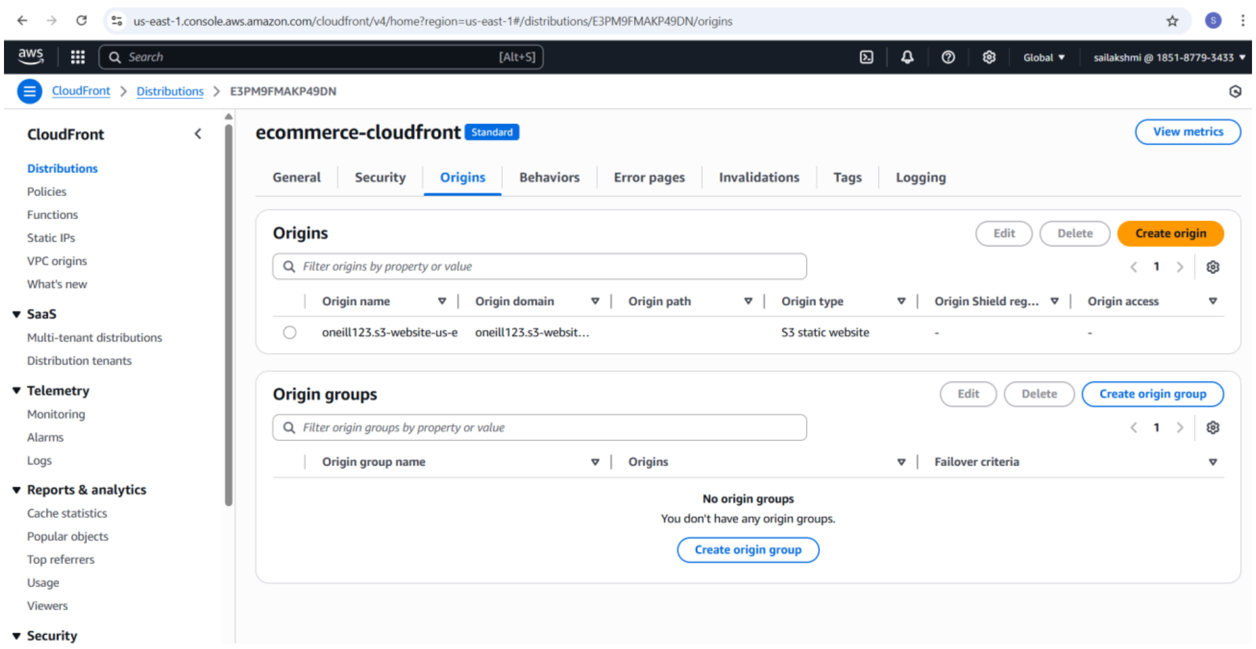
Step 7 : Create a CloudFront distribution for your S3 website



Distributions (1)

ID	Status	Description	Type	Domain n…	Alternate …	Origins	Last modifi…
E3PM9FMAKP49DN	Enabled	-	Standard	d31j53ar…	-	oneill123.s3-websi	August 1, 20…

For Origin Domain, select your S3 static website endpoint


us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions/E3PM9FMAKP49DN/origins

CloudFront > Distributions > E3PM9FMAKP49DN

CloudFront: Distributions, Policies, Functions, Static IPs, VPC origins, What's new; SaaS: Multi-tenant distributions, Distribution tenants; Telemetry: Monitoring, Alarms, Logs; Reports & analytics: Cache statistics, Popular objects, Top referrers, Usage, Viewers; Security

ecommerce-cloudfront Standard

General | Security | Origins | Behaviors | Error pages | Invalidations | Tags | Logging

Origins

Origin name	Origin domain	Origin path	Origin type	Origin Shield reg...	Origin access
oneill123.s3-website-us-e	oneill123.s3-websit...		S3 static website	-	-

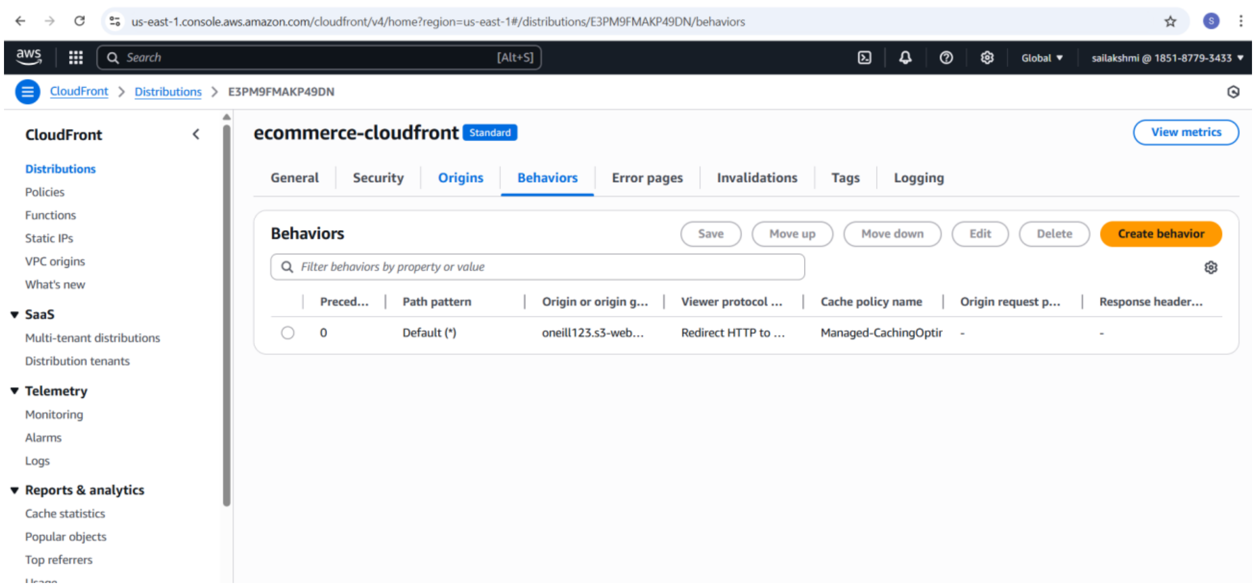
Origin groups

Origin group name	Origins	Failover criteria

No origin groups
You don't have any origin groups.

Under Default Cache Behavior,

- Viewer Protocol Policy: Redirect HTTP to HTTPS (recommended)
- Allowed HTTP Methods: GET, HEAD, OPTIONS (for static content)


us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions/E3PM9FMAKP49DN/behaviors

CloudFront > Distributions > E3PM9FMAKP49DN

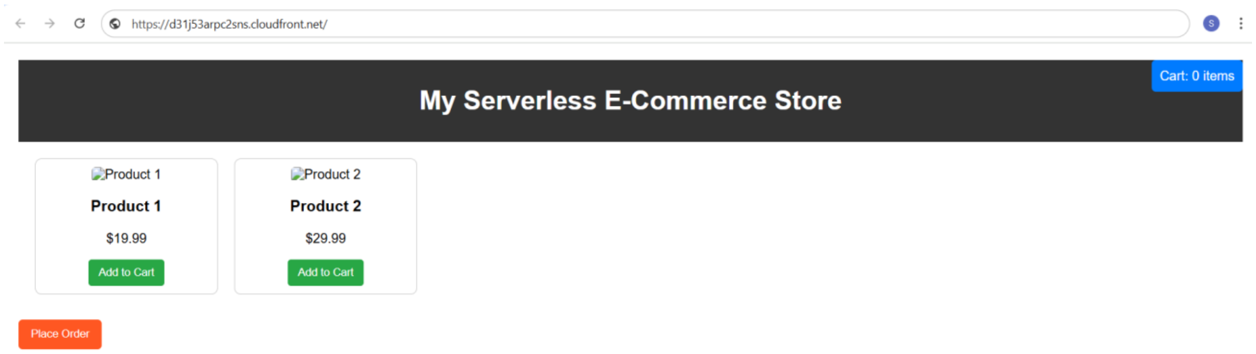
ecommerce-cloudfront Standard

General | Security | Origins | Behaviors | Error pages | Invalidations | Tags | Logging

Behaviors

Preced...	Path pattern	Origin or origin g...	Viewer protocol ...	Cache policy name	Origin request p...	Response header...
0	Default (*)	oneill123.s3-web...	Redirect HTTP to ...	Managed-CachingOptir	-	-

- Click Create Distribution and wait for deployment
- After deployed, you get a CloudFront domain name access the link



https://d31j53arpc2sns.cloudfront.net/

Cart: 0 items

My Serverless E-Commerce Store

Product 1

Product 1

$19.99

Add to Cart

Product 2

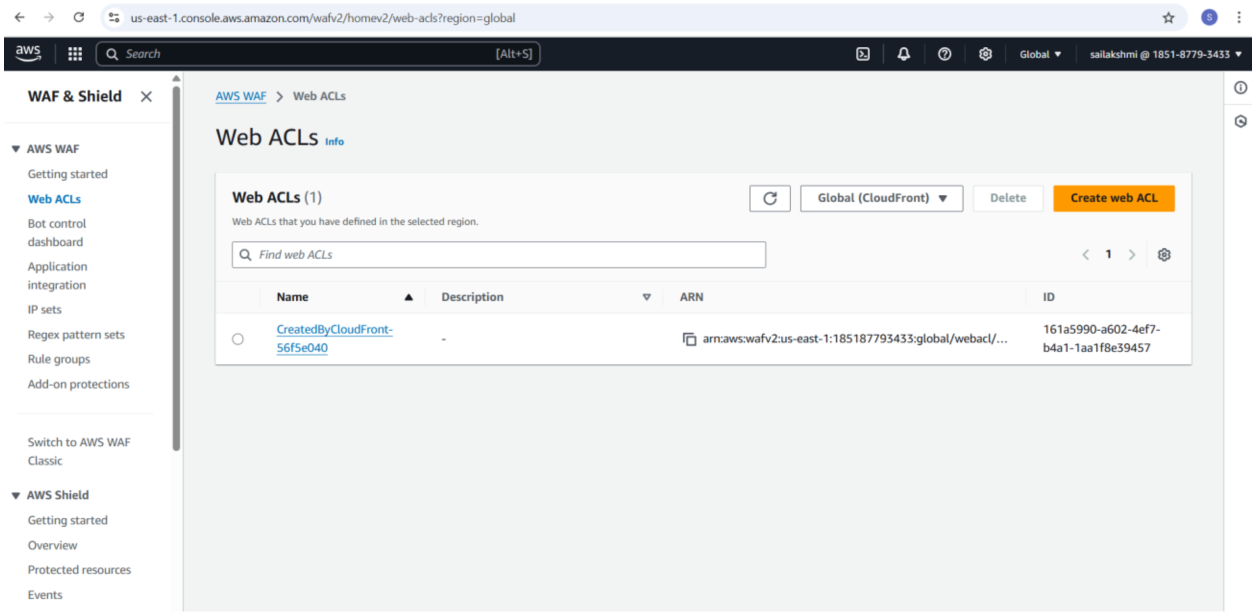
Product 2

$29.99

Add to Cart

Place Order

WAF



us-east-1.console.aws.amazon.com/wafv2/homev2/web-acls?region=global

AWS WAF > Web ACLs

Web ACLs Info

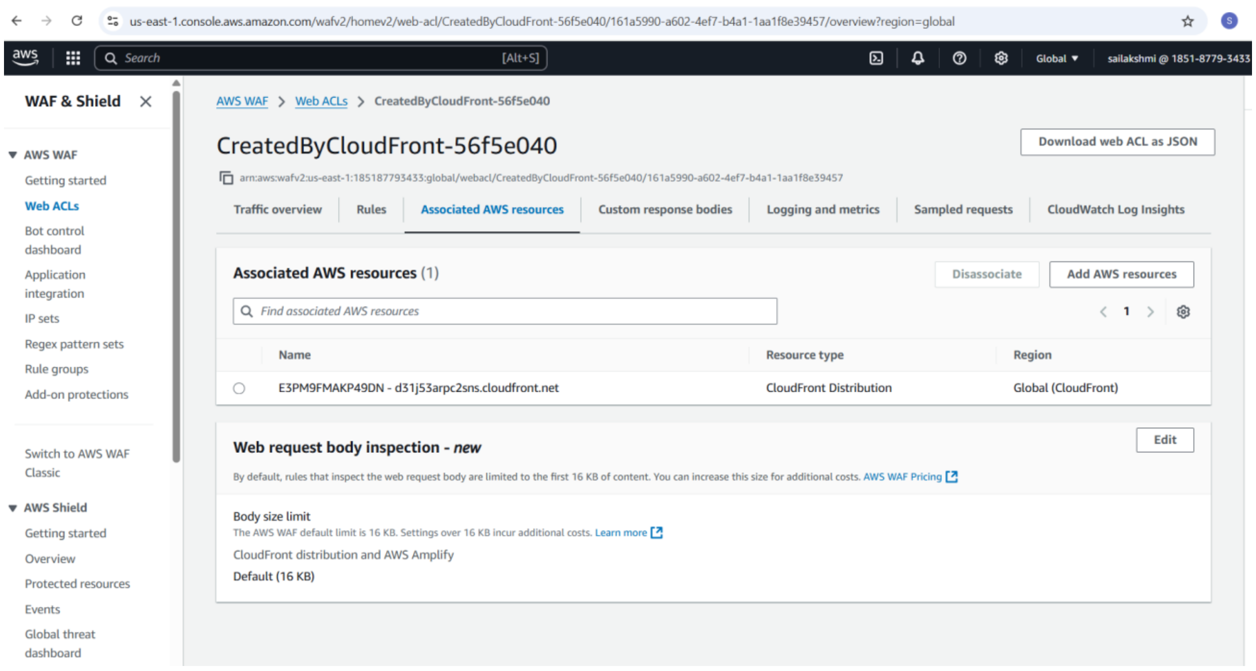
Web ACLs (1)

Web ACLs that you have defined in the selected region.

Global (CloudFront) | Delete | Create web ACL

Find web ACLs

Name	Description	ARN	ID
CreatedByCloudFront-56f5e040	-	arn:aws:wafv2:us-east-1:185187793433:global/webacl/...	161a5990-a602-4ef7-b4a1-1aa1f8e39457



us-east-1.console.aws.amazon.com/wafv2/homev2/web-acl/CreatedByCloudFront-56f5e040/161a5990-a602-4ef7-b4a1-1aa1f8e39457/overview?region=global

AWS WAF > Web ACLs > CreatedByCloudFront-56f5e040

CreatedByCloudFront-56f5e040

arn:aws:wafv2:us-east-1:185187793433:global/webacl/CreatedByCloudFront-56f5e040/161a5990-a602-4ef7-b4a1-1aa1f8e39457

Associated AWS resources (1)

Name	Resource type	Region
E3PM9FMAKP49DN - d31j53arpc2sns.cloudfront.net	CloudFront Distribution	Global (CloudFront)

Web request body inspection - *new*

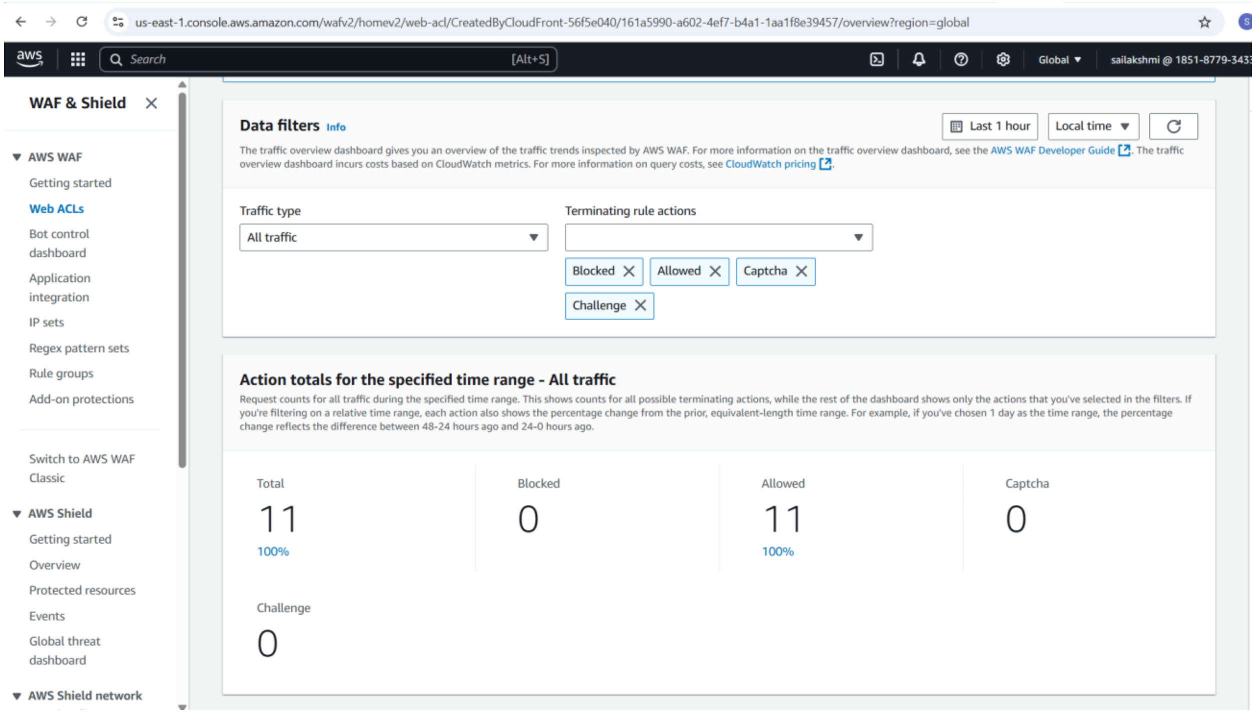
By default, rules that inspect the web request body are limited to the first 16 KB of content. You can increase this size for additional costs. AWS WAF Pricing

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)



us-east-1.console.aws.amazon.com/wafv2/homev2/web-acl/CreatedByCloudFront-56f5e040/161a5990-a602-4ef7-b4a1-1aa1f8e39457/overview?region=global

Data filters Info

The traffic overview dashboard gives you an overview of the traffic trends inspected by AWS WAF. For more information on the traffic overview dashboard, see the AWS WAF Developer Guide. The traffic overview dashboard incurs costs based on CloudWatch metrics. For more information on query costs, see CloudWatch pricing.

Traffic type: All traffic

Terminating rule actions: Blocked, Allowed, Captcha, Challenge

Action totals for the specified time range - All traffic

Request counts for all traffic during the specified time range. This shows counts for all possible terminating actions, while the rest of the dashboard shows only the actions that you've selected in the filters. If you're filtering on a relative time range, each action also shows the percentage change from the prior, equivalent-length time range. For example, if you've chosen 1 day as the time range, the percentage change reflects the difference between 48-24 hours ago and 24-0 hours ago.

Total	Blocked	Allowed	Captcha
11 (100%)	0	11 (100%)	0

Challenge: 0