

# **Deploying Serverless Web Application on AWS Using: S3**

**API Gateway**

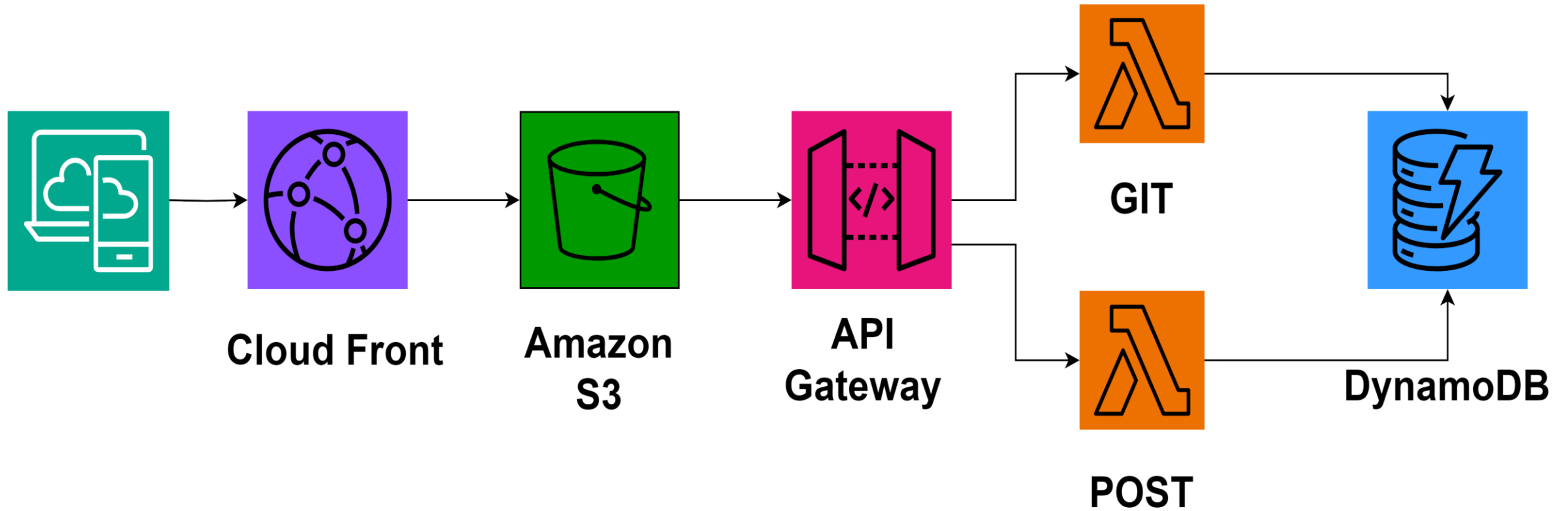
**Lambda**

**DynamoDB**

**CloudFront**

By: Salma Salah

# Project's Diagram



# First

## Create DynamoDB table

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

DynamoDB

X

Dashboard

Tables

Explore items

PartiQL editor

Backups

Exports to S3

Imports from S3

Integrations 

New

Reserved capacity

Settings

▼ DAX

Clusters

Subnet groups

Parameter groups

Events

DynamoDB > Tables

Tables (0) 

Info

Find tables by table name

Any tag key

Any tag value

< 1 >

Na...

▲

Status

Partition key

Sort key

Indexes

Deletion protection

Read capacity mode

Write capacity mode

You have no tables in this account in this AWS Region.

Create table

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

DynamoDB > Tables > Create table

Create table

Table details 

Info

DynamoDB is a schemaless database that requires only a table name and a primary key when you create the table.

Table name

This will be used to identify your table.

studentData

Between 3 and 255 characters, containing only letters, numbers, underscores (\_), hyphens (-), and periods (.).

Partition key

The partition key is part of the table's primary key. It is a hash value that is used to retrieve items from your table and allocate data across hosts for scalability and availability.

studentid

String

1 to 255 characters and case sensitive.

Sort key - optional

You can use a sort key as the second part of a table's primary key. The sort key allows you to sort or search among all items sharing the same partition key.

Enter the sort key name

String

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Table After Creation

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

DynamoDB

Dashboard

Tables

Explore items

PartiQL editor

Backups

Exports to S3

Imports from S3

Integrations

Reserved capacity

Settings

DAX

Clusters

Subnet groups

Parameter groups

Events

DynamoDB > Tables

Tables (1) Info

Find tables by table name

Any tag key

Any tag value

< 1 >

Actions

Delete

Create table

	Name	Status	Partition key	Sort key	Indexes	Deletion protection	Read capacity mode	Write capacity mode
<input type="checkbox"/>	<a href="#">studentData</a>	Active	studentid (S)	-	0	Off	Provisioned (5)	Provisioned (5)

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

# Second

## Create IAM Rule to Allow Lambda function to access DynamoDB

The screenshot displays the AWS IAM console interface. On the left, the 'Identity and Access Management (IAM)' sidebar is visible, with 'Roles' selected under 'Access management'. The main content area shows the 'Roles (18)' page, which includes a search bar and a table of existing roles. The table has columns for 'Role name', 'Trusted entities', and 'Last activity'. Below the table, the 'Create role' wizard is shown, with the 'Web identity' option selected. The 'Use case' section is also visible, showing 'Lambda' as the selected service.

**Roles (18)** Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

Role name	Trusted entities	Last activity
<a href="#">AmazonRedshift-CommandsAccessRole-20240115T044836</a>	AWS Service: redshift-serverless, <a href="#">anc</a>	-
<a href="#">AWSServiceRoleForAmazonElasticFileSystem</a>	AWS Service: elasticfilesystem (Servi	179 days ag
<a href="#">AWSServiceRoleForApplicationAutoScaling_DynamoDBTable</a>	AWS Service: dynamodb.application-	-
<a href="#">AWSServiceRoleForAutoScaling</a>	AWS Service: autoscaling (Service-Lir	174 days ag
<a href="#">AWSServiceRoleForBackup</a>	AWS Service: backup (Service-Linked	21 hours ag
<a href="#">AWSServiceRoleForECS</a>	AWS Service: ecs (Service-Linked Rol	148 days ag
<a href="#">AWSServiceRoleForElasticLoadBalancing</a>	AWS Service: elasticloadbalancing (\$	148 days ag
<a href="#">AWSServiceRoleForNetworkFirewall</a>	AWS Service: network-firewall (Servi	91 days ago

**Create role**

☐ Web identity  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ SAML 2.0 federation  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ Custom trust policy  
Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case  
Lambda

Choose a use case for the specified service.  
Use case  
☒ Lambda  
Allows Lambda functions to call AWS services on your behalf.

Cancel Next

# Policies that are Attached to the Rule

IAM > Roles > Create role

Step 1  
[Select trusted entity](#)

Step 2  
[Add permissions](#)

Step 3  
Name, review, and create

Name, review, and create

Role details

Role name

Enter a meaningful name to identify this role.

LambdaDynamo-rule

Maximum 64 characters. Use alphanumeric and '+-,@\_.' characters.

Description

Add a short explanation for this role.

Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: \_+-. @~ / [ ] ! \$ % ^ & \* ~ " ' , ; : .

Step 1: Select trusted entities

Edit

Trust policy

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings

Access reports

- Access Analyzer
- External access
- Unused access

Last activity  
✓ 26 minutes ago

Maximum session duration  
1 hour

Permissions

Trust relationships

Tags

Access Advisor

Revoke sessions

Permissions policies (3) Info

Refresh

Simulate

Remove

Add permissions ▼

You can attach up to 10 managed policies.

Search

Filter by Type  
All types

< 1 >

<input type="checkbox"/>	Policy name ↗	Type ▾	Attached entities ▾
<input type="checkbox"/>	+ AmazonDynamoDBFullAccess	AWS managed	1
<input type="checkbox"/>	+ AWSLambdaBasicExecutionR...	AWS managed	2
<input type="checkbox"/>	+ AWSLambdaInvocation-Dyna...	AWS managed	1

▼ Permissions boundary (not set)

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Third

## Creating Two Lambda Functions

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

Compute

AWS Lambda

lets you run code without thinking about servers.

You pay only for the compute time that you consume — there is no charge when your code is not running. With Lambda, you can run code for virtually any type of application or backend service, all with zero administration.

Get started

Author a Lambda function from scratch, or choose from one of many preconfigured examples.

Create a function

How it works

Run

Next: Lambda responds to events

.NET

Java

Node.js

Python

Ruby

Custom runtime

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

Lambda > Functions > Create function

Create function

Choose one of the following options to create your function.

☒ Author from scratch

Start with a simple Hello World example.

☐ Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

☐ Container image

Select a container image to deploy for your function.

Basic information

Function name

Enter a name that describes the purpose of your function.

getStudent

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime

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

Python 3.12

Architecture

Choose the instruction set architecture you want for your function code.

x86\_64

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# First Lambda Function (getStudent) that will get data from DynamoDB table and adding python code source

The screenshot shows the 'Create new function' wizard in the AWS Lambda console. The 'Basic information' tab is selected. The language is set to 'Python 3.12'. The architecture is set to 'x86\_64'. The permissions are set to 'Use an existing role'. The existing role is set to 'LambdaDynamo-rule'.

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

Python 3.12

Architecture [Info](#)

Choose the instruction set architecture you want for your function code.

☒ x86\_64

☐ arm64

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

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console [link](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role

Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

LambdaDynamo-rule

[View the LambdaDynamo-rule role on the IAM console.](#)

The screenshot shows the 'Code source' tab for the 'getStudent' function. The code is written in Python and uses the boto3 library to scan a DynamoDB table named 'studentData'.

Code source [Info](#)

Upload from ▼

File Edit Find View Go Tools Window Test Deploy Changes not deployed

Go to Anything (Ctrl-P)

Environment

```
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # Initialize a DynamoDB resource object for the specified region
6     dynamodb = boto3.resource('dynamodb', region_name='us-east-1')
7
8     # Select the DynamoDB table named 'studentData'
9     table = dynamodb.Table('studentData')
10
11     # Scan the table to retrieve all items
12     response = table.scan()
13     data = response['Items']
14
15     # If there are more items to scan, continue scanning until all items are retrieved
16     while 'LastEvaluatedKey' in response:
17         response = table.scan(ExclusiveStartKey=response['LastEvaluatedKey'])
18         data.extend(response['Items'])
19
20     # Return the retrieved data
21     return data
22
```



# Creating Second Lambda Function (insertStudentdata) that will PUT in DynamoDB table

The image shows two side-by-side screenshots of the AWS Lambda console. The left screenshot displays the 'Create function' page, and the right screenshot displays the 'Code source' editor.

**Left Screenshot: Create function**

The 'Create function' page is shown with the following details:

- Options to create function:**
  - ☒ Author from scratch: Start with a simple Hello World example.
  - ☐ Use a blueprint: Build a Lambda application from sample code and configuration presets for common use cases.
  - ☐ Container image: Select a container image to deploy for your function.
- Basic information:**
  - Function name:** insertStudentData (Note: Use only letters, numbers, hyphens, or underscores with no spaces.)
  - Runtime:** Python 3.12 (Note: Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.)
  - Architecture:** x86\_64 (Note: Choose the instruction set architecture you want for your function code.)

**Right Screenshot: Code source**

The 'Code source' editor shows the following Python code:

```
1 import json
2 import boto3
3
4 # Create a DynamoDB object using the AWS SDK
5 dynamodb = boto3.resource('dynamodb')
6 # Use the DynamoDB object to select our table
7 table = dynamodb.Table('studentData')
8
9 # Define the handler function that the Lambda service will use as an entry point
10 def lambda_handler(event, context):
11     # Extract values from the event object we got from the Lambda service and store in variables
12     student_id = event['studentid']
13     name = event['name']
14     student_class = event['class']
15     age = event['age']
16
17     # Write student data to the DynamoDB table and save the response in a variable
18     response = table.put_item(
19         Item={
20             'studentid': student_id,
21             'name': name,
22             'class': student_class,
23             'age': age
24         }
25     )
26
27     # Return a properly formatted JSON object
28     return {
29         'statusCode': 200,
30         'body': json.dumps('Student data saved successfully!')
31     }
```

# Test the Execution of Second (PUT) lambda function by adding data to DynamoDB

The image displays two side-by-side screenshots of the AWS Lambda console interface, illustrating the process of testing a Lambda function by adding data to DynamoDB.

**Left Screenshot (Event Creation):**

- Event name:** test
- Event sharing settings:** Private (selected)
- Template - optional:** hello-world
- Event JSON:**

```
1 {  
2   "studentid": "id",  
3   "name": "Salma",  
4   "class": "A",  
5   "age": "24"  
6 }
```
- Buttons:** Cancel, Invoke, Save

**Right Screenshot (Execution Results):**

- Message:** The test event test was successfully saved.
- Code source:** Info
- Buttons:** Test, Deploy
- Execution results:** Status: Succeeded, Max memory used: 77 MB, Time: 558.86 ms
- Test Event Name:** test
- Response:**

```
{  
  "statusCode": 200,  
  "body": "\\Student data saved successfully!\\n"  
}
```
- Function Logs:**

```
START RequestId: f6515977-76b0-4299-82ce-d8182ad2a3c4 Version: $LATEST  
END RequestId: f6515977-76b0-4299-82ce-d8182ad2a3c4  
REPORT RequestId: f6515977-76b0-4299-82ce-d8182ad2a3c4 Duration: 558.86 ms Billed Duration: 559 ms Memory Size: 128 MB Max Memory Used: 77 MB
```
- Request ID:** f6515977-76b0-4299-82ce-d8182ad2a3c4

# Test the Execution of First (GET) lambda function by retrieving data from DynamoDB

The screenshot shows the AWS DynamoDB console interface. The left sidebar contains navigation links: Dashboard, Tables, Explore items, PartiQL editor, Backups, Exports to S3, Imports from S3, Integrations, Reserved capacity, and Settings. The main content area is titled 'studentData' and includes a 'Tables (1)' section with a search bar and a list of tables. Below this, there is a 'Scan or query items' section with a status message: 'Completed. Read capacity units consumed: 0.5'. The 'Items returned (1)' section displays a table with one item:

studentid (String)	age	class	name
1	24	A	Salma

The bottom of the console shows the footer with 'CloudShell', 'Feedback', and copyright information for Amazon Web Services.

The screenshot shows the AWS Lambda console interface. The left sidebar contains navigation links: Dashboard, Functions, Layers, Triggers, Environment, and Settings. The main content area is titled 'Code source' and includes a 'Test' button. Below this, there is an 'Execution results' section with a status message: 'Status: Succeeded'. The 'Test Event Name' is 'test'. The 'Response' section displays a JSON object:

```
{  "studentid": "1",  "name": "Salma",  "class": "A",  "age": "24"}
```

The 'Function Logs' section shows the execution details, including the Request ID, Duration, Billed Duration, Memory Size, and Max Memory Used.

# Fourth

## Creating REST API using API Gateway

WebSocket 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

Build

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

ImportBuild

API details

New API

Create a new REST API.

Clone existing API

Create a copy of an API in this AWS account.

Import API

Import an API from an OpenAPI definition.

Example API

Learn about API Gateway with an example API.

API name

student

Description - optional

API endpoint type

Regional APIs are deployed in the current AWS Region. Edge-optimized APIs route requests to the nearest CloudFront Point of Presence. Private APIs are only accessible from VPCs.

Edge-optimized

API Gateway

Successfully created REST API 'student (ulm6amfyg)'

API Gateway

APIs

Custom domain names

VPC links

API: student

Resources

Stages

Authorizers

Gateway responses

Models

Resource policy

Documentation

Dashboard

API settings

Usage plans

Resources

API actions

Deploy API

Create resource

Resource details

Update documentation

Enable CORS

Path

Resource ID

/

zmka3G48

Methods (0)

Delete

Create method

Method type

Integration type

Authorization

API key

No methods

No methods defined.

# Creating GET Method and checking that it's working

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

GET

Integration type

Lambda function

Integrate your API with a Lambda function.

HTTP

Integrate with an existing HTTP endpoint.

Mock

Generate a response based on API Gateway mappings and transformations.

AWS service

Integrate with an AWS Service.

VPC link

Integrate with a resource that isn't accessible over the public internet.

Lambda proxy integration

Send the request to your Lambda function as a structured event.

Lambda function

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

us-east-1

Q am:aws:lambda:us-east-1:821594020462:function:getS

API Gateway

APIs

Custom domain names

VPC links

API: student

Resources

Stages

Authorizers

Gateway responses

Models

Resource policy

Documentation

Dashboard

API settings

Usage plans

Create resource

/

GET

GET method test results

Request

/

Status

200

Response body

[{"studentid": "1", "name": "Salma", "class": "A", "age": "24"}]

Response headers

{  
 "Content-Type": "application/json",  
 "X-Amzn-Trace-Id": "Root=1-6641ab69-6dea9c0ec6019ec7d5af86b5;Parent=0bf8556c4bf0d53a;Sampled=0;lineage=20ed1162:0"  
}

Logs

Execution log for request 6cbeb10e-c6df-4cf5-a0a7-109356cdabc0  
Mon May 13 05:55:53 UTC 2024 : Starting execution for request:  
6cbeb10e-c6df-4cf5-a0a7-109356cdabc0

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Creating PUT Method and Post Method

aws

Services

Search

[Alt+S]

POST

Integration type

Lambda function

Integrate your API with a Lambda function.

HTTP

Integrate with an existing HTTP endpoint.

Mock

Generate a response based on API Gateway mappings and transformations.

AWS service

Integrate with an AWS Service.

VPC link

Integrate with a resource that isn't accessible over the public internet.

Lambda proxy integration

Send the request to your Lambda function as a structured event.

Lambda function

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

us-east-1

am:aws:lambda:us-east-1:821594020462:function:inse

aws

Services

Search

[Alt+S]

API Gateway

API Gateway > APIs > Resources - student (ulm6amfjvg)

APIs

Custom domain names

VPC links

▼ API: student

Resources

Stages

Authorizers

Gateway responses

Models

Resource policy

Documentation

Dashboard

API settings

Usage plans

Resources

API actions ▼

Deploy API

Create resource

/

GET

OPTIONS

POST

PUT

ARN

am:aws:execute-api:us-east-1:821594020462:ulm6amfjvg/\*/PUT/

Resource ID

zrnka3is48

Client

Method request

Integration request

Lambda integration

Method response

Integration response

Method request

Integration request

Integration response

Method response

Test

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Fifth

## Creating S3 Bucket and Uploading code files

The screenshot shows the AWS Management Console interface. At the top, a green notification banner states: "Successfully created bucket 'serverlesswebapp01'". Below this, the "Buckets" section is visible. Under the "General purpose buckets" tab, a table lists the newly created bucket:

Name	AWS Region	IAM Access Analyzer	Creation date
serverlesswebapp01	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	May 13, 2024, 09:19:11 (UTC+03:00)

The screenshot shows the "Upload" page for the bucket "serverlesswebapp01". It includes a dashed box for dragging and dropping files, and a table of files and folders to be uploaded:

Name	Folder	Type
index.html	-	text/html
scripts.js	-	text/javascript

# Enable Public Access and Static website hosting

The screenshot shows the AWS Management Console interface. The breadcrumb navigation at the top reads: Amazon S3 > Buckets > serverlesswebapp01 > Edit Block public access (bucket settings). The main heading is 'Edit Block public access (bucket settings)' with an 'Info' link. Below this is a section titled 'Block public access (bucket settings)' with a descriptive paragraph. A list of four settings follows, each with a checkbox and a description:

- ☐ **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

The footer contains 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

The screenshot shows the AWS Management Console interface. The breadcrumb navigation at the top reads: Amazon S3 > Buckets > serverlesswebapp01 > Edit static website hosting. The main heading is 'Edit static website hosting' with an 'Info' link. Below this is a section titled 'Static website hosting' with a descriptive paragraph and a 'Learn more' link. Two radio buttons are present:

- ☐ **Disable**
- ☒ **Enable**

Below the radio buttons is a section titled 'Hosting type' with two options:

- ☒ **Host a static website**  
Use the bucket endpoint as the web address. [Learn more](#)
- ☐ **Redirect requests for an object**  
Redirect requests to another bucket or domain. [Learn more](#)

A blue information box contains the following text:

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#)

Below this is a section titled 'Index document' with a descriptive paragraph.

The footer contains 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.



# Generating Bucket Policy

A Policy is a container for permissions. The different types of policies you can create are an [IAM Policy](#), an [S3 Bucket Policy](#), an [SNS Topic Policy](#), a [VPC Endpoint Policy](#), and an [SQS Queue Policy](#).

Select Type of Policy S3 Bucket Policy

## Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a [description of elements](#) that you can use in statements.

Effect ☒ Allow ☐ Deny

Principal \*

Use a comma to separate multiple values.

AWS Service Amazon S3 ☐ All Services <sup>(\*)</sup>

Use multiple statements to add permissions for more than one service.

Actions 1 Action(s) Selected ☐ All Actions <sup>(\*)</sup>

Amazon Resource Name (ARN) arn:aws:s3:::serverlessweb

ARN should follow the following format: arn:aws:s3:::{BucketName}/{Key(Names)}.  
Use a comma to separate multiple values.

[Add Conditions \(Optional\)](#)

Add Statement

## Step 3: Generate Policy

A *policy* is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

Add one or more statements above to generate a policy.

Amazon Resource Name (ARN)

ARN should follow the following format: arn:aws:s3:::{BucketName}/{Key(Names)}.  
Use a comma to separate multiple values.

[Add Conditions \(Optional\)](#)

Add Statement

You added the following statements. Click the button below to Generate a policy.

Principal(s)	Effect	Action	Resource	Conditions
• *	Allow	• s3:GetObject	arn:aws:s3:::serverlesswebapp01	None

## Step 3: Generate Policy

A *policy* is a document (written in the [Access Policy Language](#)) that acts as a container for one or more statements.

Generate Policy

[Start Over](#)

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

©2010, Amazon Web Services LLC or its affiliates. All rights reserved.

An  amazon.com company

# Adding Pucket Policy and Editing CORS in S3

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

Bucket policy

Policy examples

Policy generator

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:::serverlesswebapp01

Policy

1 {  
2 "Id": "Policy1715583686568",  
3 "Version": "2012-10-17",  
4 "Statement": [  
5 {  
6 "Sid": "Stmnt1715583628881",  
7 "Action": [  
8 "s3:GetObject"  
9 ],  
10 "Effect": "Allow",  
11 "Resource": "arn:aws:s3:::serverlesswebapp01/\*",  
12 "Principal": "\*"   
13 }  
14 ]  
15 }

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

Amazon S3 > Buckets > serverlesswebapp01 > Edit cross-origin resource sharing (CORS)

Edit cross-origin resource sharing (CORS)

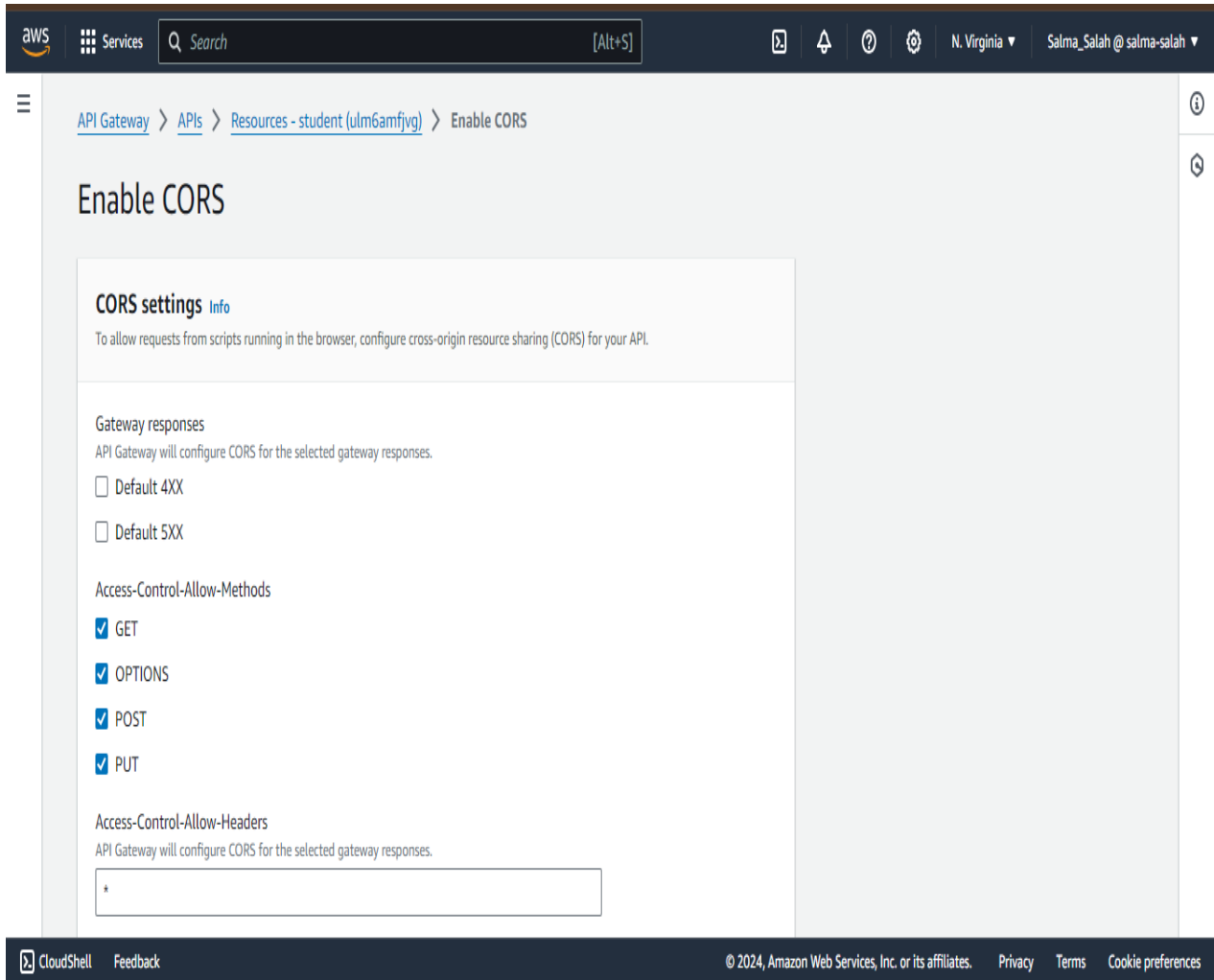
Info Info

Cross-origin resource sharing (CORS)

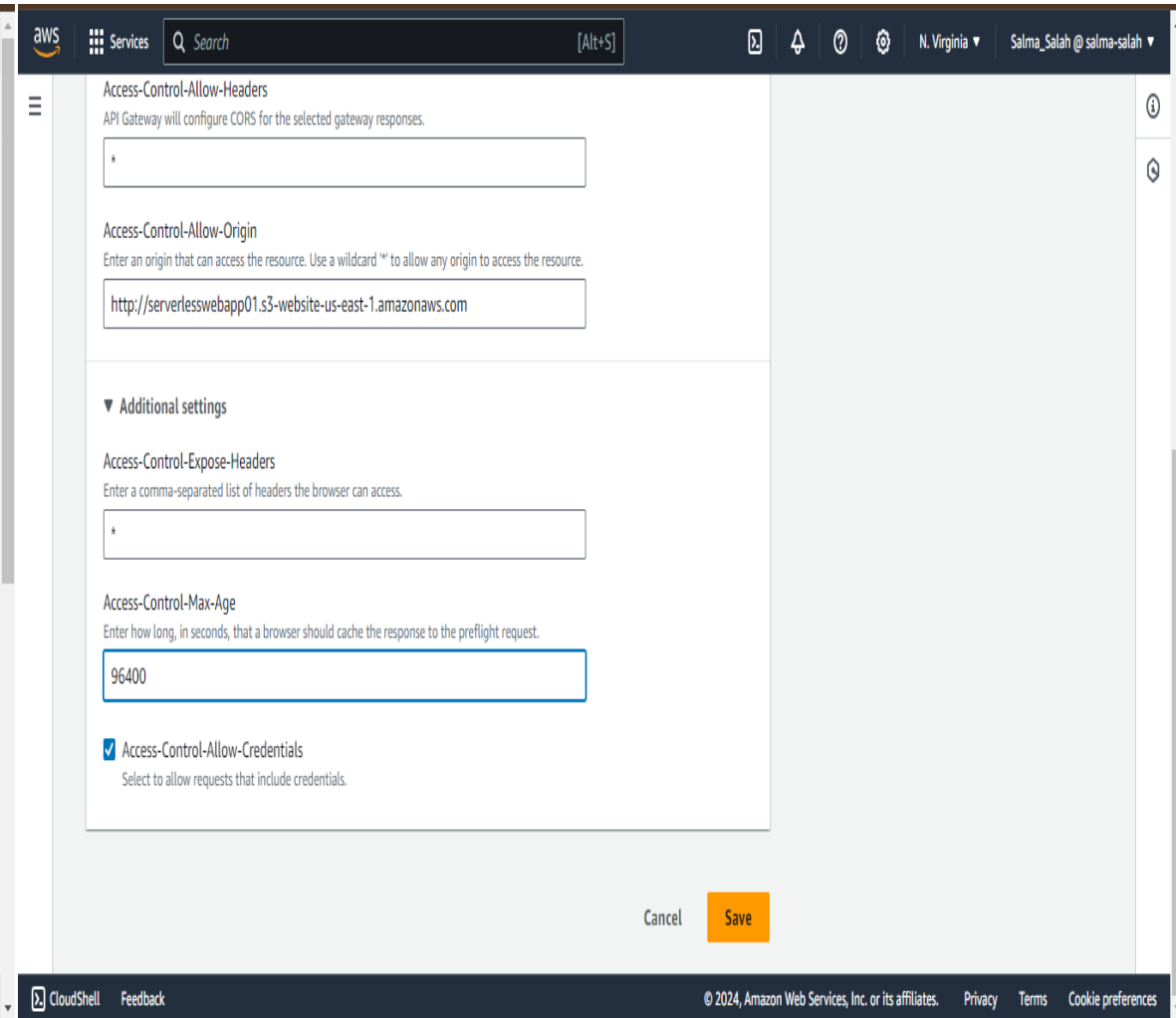
The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one domain to interact with resources in a different domain. [Learn more](#)

1 {  
2 {  
3 "AllowedHeaders": [  
4 "\*"   
5 ],  
6 "AllowedMethods": [  
7 "GET"  
8 ],  
9 "AllowedOrigins": [  
10 "\*"   
11 ],  
12 "ExposeHeaders": [  
13 ]  
14 }  
15 }

# Enable CORS on API Gateway and adding S3 buckets's Endpoint as Origin

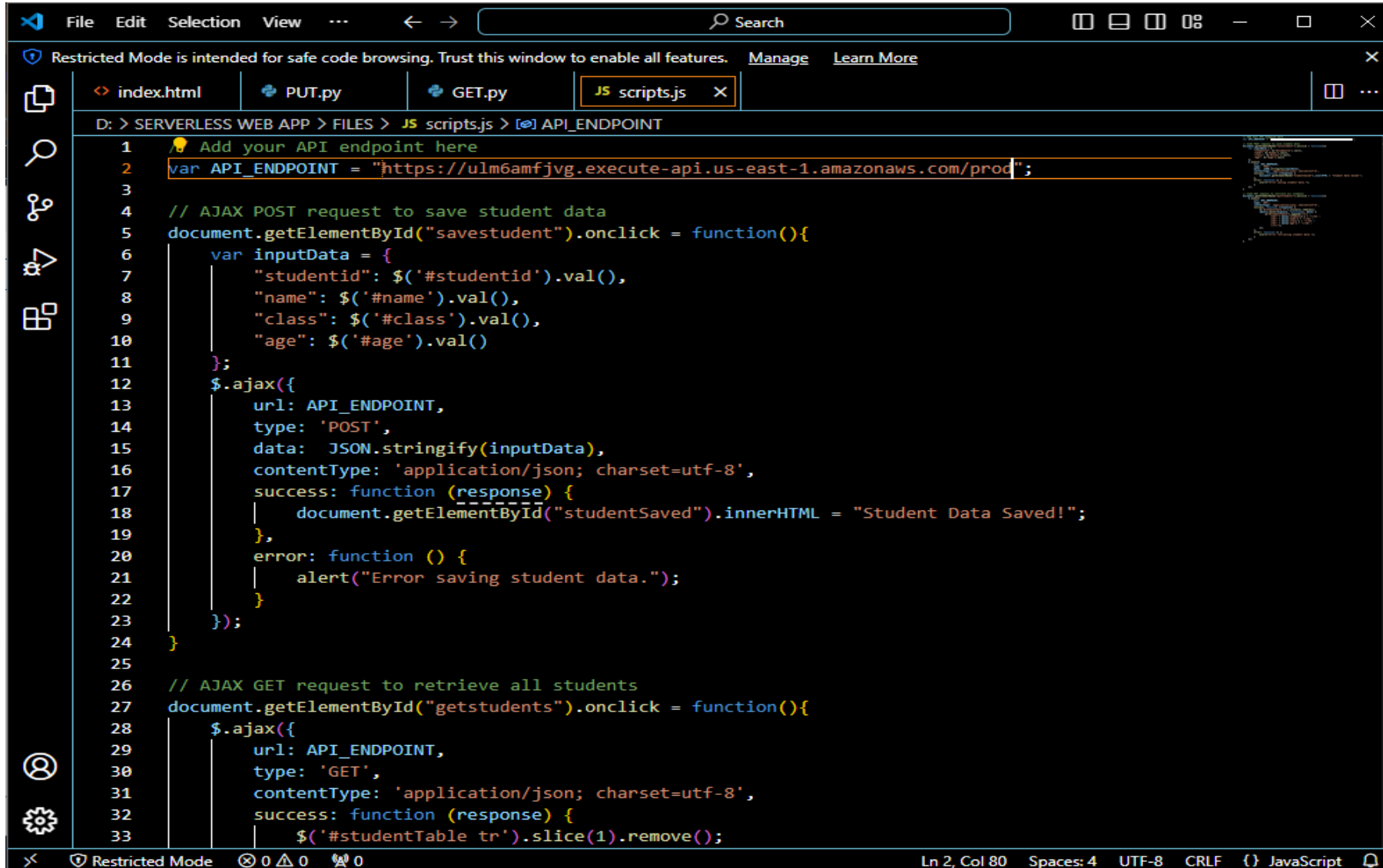


The screenshot shows the AWS API Gateway console. The breadcrumb navigation is [API Gateway](#) > [APIs](#) > [Resources - student \(ulm6amfjvg\)](#) > [Enable CORS](#). The main heading is 'Enable CORS'. Below it, there's a 'CORS settings' section with an 'Info' link. A descriptive text says: 'To allow requests from scripts running in the browser, configure cross-origin resource sharing (CORS) for your API.' Under 'Gateway responses', there's a note: 'API Gateway will configure CORS for the selected gateway responses.' There are two unchecked checkboxes: 'Default 4XX' and 'Default 5XX'. Under 'Access-Control-Allow-Methods', there are four checked checkboxes: 'GET', 'OPTIONS', 'POST', and 'PUT'. Under 'Access-Control-Allow-Headers', there's a note: 'API Gateway will configure CORS for the selected gateway responses.' and a text input field containing an asterisk (\*).



The screenshot shows the 'Additional settings' section of the AWS API Gateway console. It includes three sections: 'Access-Control-Allow-Headers' with a text input field containing an asterisk (\*); 'Access-Control-Allow-Origin' with a note 'Enter an origin that can access the resource. Use a wildcard "\*" to allow any origin to access the resource.' and a text input field containing 'http://serverlesswebapp01.s3-website-us-east-1.amazonaws.com'; and 'Access-Control-Expose-Headers' with a note 'Enter a comma-separated list of headers the browser can access.' and a text input field containing an asterisk (\*). Below these is the 'Access-Control-Max-Age' section with a note 'Enter how long, in seconds, that a browser should cache the response to the preflight request.' and a text input field containing '96400'. At the bottom, there is a checked checkbox for 'Access-Control-Allow-Credentials' with a note 'Select to allow requests that include credentials.' At the bottom right, there are 'Cancel' and 'Save' buttons.

# Adding API EndPoint in backend code



The screenshot shows a code editor window with a dark theme. The top bar includes a search bar and window management icons. Below the top bar, a notification states: "Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More". The editor has four tabs: "index.html", "PUT.py", "GET.py", and "JS scripts.js" (which is active and highlighted with a yellow border). The breadcrumb path is "D: > SERVERLESS WEB APP > FILES > JS scripts.js > API\_ENDPOINT". The code is as follows:

```
1  Add your API endpoint here
2  var API_ENDPOINT = "https://ulm6amfjvg.execute-api.us-east-1.amazonaws.com/prod";
3
4  // AJAX POST request to save student data
5  document.getElementById("savestudent").onclick = function(){
6      var inputData = {
7          "studentid": $('#studentid').val(),
8          "name": $('#name').val(),
9          "class": $('#class').val(),
10         "age": $('#age').val()
11     };
12     $.ajax({
13         url: API_ENDPOINT,
14         type: 'POST',
15         data: JSON.stringify(inputData),
16         contentType: 'application/json; charset=utf-8',
17         success: function (response) {
18             document.getElementById("studentSaved").innerHTML = "Student Data Saved!";
19         },
20         error: function () {
21             alert("Error saving student data.");
22         }
23     });
24 }
25
26 // AJAX GET request to retrieve all students
27 document.getElementById("getstudents").onclick = function(){
28     $.ajax({
29         url: API_ENDPOINT,
30         type: 'GET',
31         contentType: 'application/json; charset=utf-8',
32         success: function (response) {
33             $('#studentTable tr').slice(1).remove();
```

The status bar at the bottom shows "Ln 2, Col 80", "Spaces: 4", "UTF-8", "CRLF", and "JavaScript".

# Deploying API

The screenshot shows the 'Deploy API' modal dialog in the AWS API Gateway console. The dialog has a title bar with a close button (X). The main content area contains the following fields and controls:

- Stage:** A dropdown menu currently showing '\*New stage\*'. To its right is a 'Deploy API' button.
- Stage name:** A text input field containing the text 'prod'.
- Deployment description:** A large text area for providing a description of the deployment.
- Buttons:** At the bottom right, there are 'Cancel' and 'Deploy' buttons.
- Information:** A blue informational box states: 'A new stage will be created with the default settings. Edit your stage settings on the Stage page.'

The background of the console shows the API Gateway navigation menu on the left and the 'Resources' page for the 'student' API.

The screenshot shows the 'Resources' page for the 'student' API in the AWS API Gateway console. The page has a title bar with a close button (X) and a search bar. The main content area displays the following information:

- APIs:** A list of APIs with 'student' selected.
- Resources:** A list of resources with '/' selected.
- Method execution:** A section titled '/ - POST - Method execution' showing the execution flow. It includes a 'Create resource' button, 'Update documentation' and 'Delete' buttons, and a diagram illustrating the flow from 'Client' to 'Method request' to 'Integration request' to 'Lambda integration' and back through 'Integration response' and 'Method response'.
- ARN:** The ARN for the resource is displayed as 'arn:aws:execute-api:us-east-1:821594020462:ulm6amfjvg/\*/\*POST/'.
- Resource ID:** The Resource ID is displayed as 'zrnka3is48'.

The background of the console shows the API Gateway navigation menu on the left and the 'Resources' page for the 'student' API.

# Testing the Application and it's successfully working and adding new records and retrieving all records

API Gateway - Stages

serverlesswebapp01 - S3 bucke

Functions - Lambda

Student Data

serverlesswebapp01.s3-website-us-east-1.amazonaws.com

Save Student Data

Student Data Saved!

View all Students

Student ID	Name	Class	Age
2	Ahmed	B	25
1	Salma	A	24

API Gateway - Enable CORS

Edit cross-origin resource shar

Functions - Lambda

Student Data

serverlesswebapp01.s3-website-us-east-1.amazonaws.com

Salma Salah to Save and View Data of Students

Student ID:

3

Name:

Nour

Class:

C

Age:

26

Save Student Data

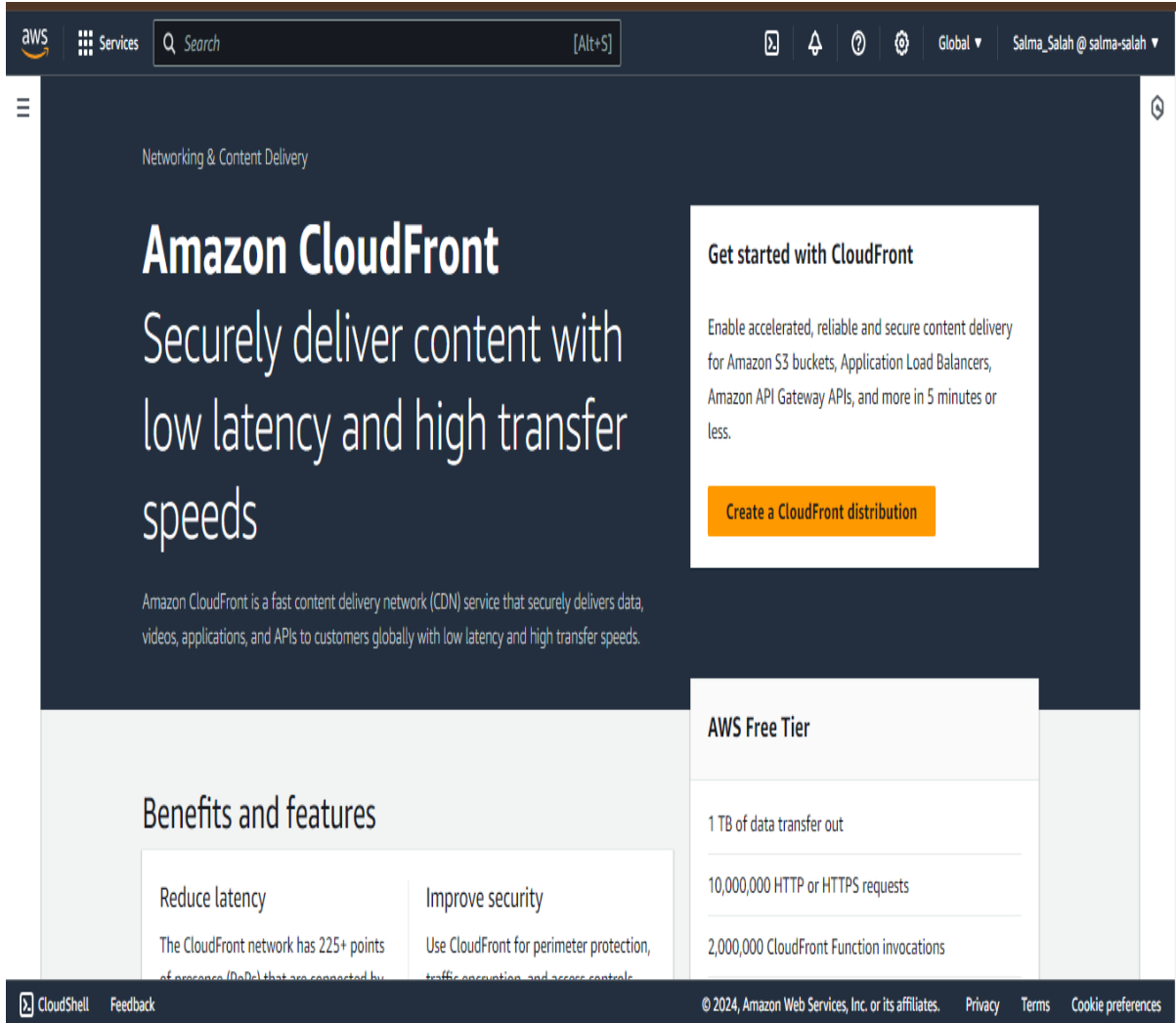
Student Data Saved!

View all Students

Student ID	Name	Class	Age
2	Ahmed	B	25
1	Salma	A	24
3	Nour	C	26

# Last Step

## Creating Cloud Front Distribution to make the Connection Secured (HTTPS)



The screenshot shows the Amazon CloudFront landing page. The header includes the AWS logo, a search bar, and navigation links. The main content area features the heading "Amazon CloudFront" and the subheading "Securely deliver content with low latency and high transfer speeds". Below this, there is a "Get started with CloudFront" section with a "Create a CloudFront distribution" button. The "AWS Free Tier" section lists benefits like "1 TB of data transfer out", "10,000,000 HTTP or HTTPS requests", and "2,000,000 CloudFront Function invocations". The "Benefits and features" section highlights "Reduce latency" and "Improve security".

Networking & Content Delivery

## Amazon CloudFront

Securely deliver content with low latency and high transfer speeds

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency and high transfer speeds.

### 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](#)

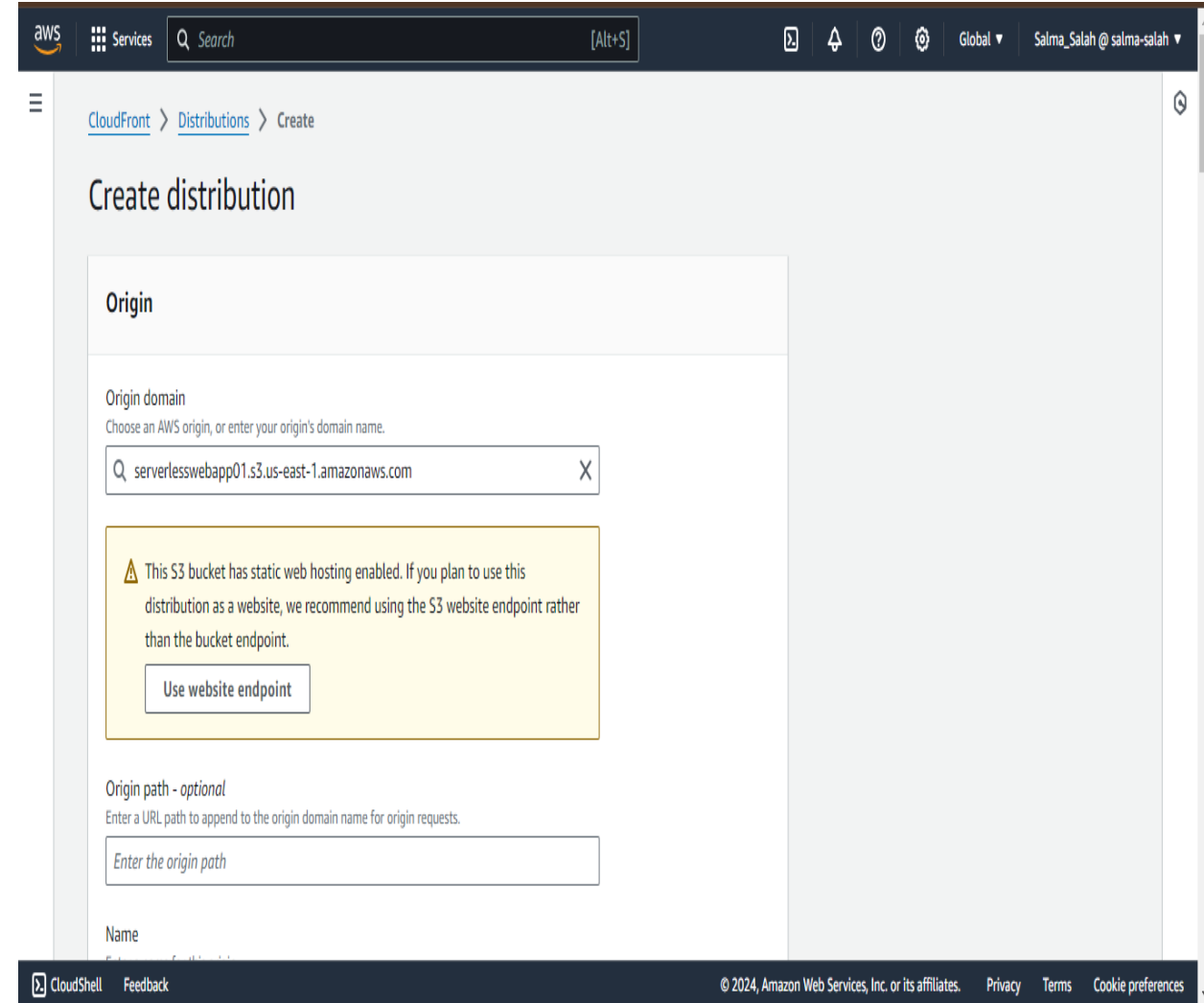
### AWS Free Tier

- 1 TB of data transfer out
- 10,000,000 HTTP or HTTPS requests
- 2,000,000 CloudFront Function invocations

### Benefits and features

- Reduce latency**  
The CloudFront network has 225+ points of presence (PoPs) that are connected by a global network of high-speed, low-latency links.
- Improve security**  
Use CloudFront for perimeter protection, traffic inspection, and access controls.

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



The screenshot shows the "Create distribution" page in the Amazon CloudFront console. The breadcrumb navigation is "CloudFront > Distributions > Create". The main heading is "Create distribution". The "Origin" section has a text input field for "Origin domain" with the value "serverlesswebapp01.s3.us-east-1.amazonaws.com". Below this, there is a yellow warning box stating: "This S3 bucket has static web hosting enabled. If you plan to use this distribution as a website, we recommend using the S3 website endpoint rather than the bucket endpoint." with a "Use website endpoint" button. The "Origin path - optional" section has a text input field with the placeholder "Enter the origin path". The "Name" section is partially visible at the bottom.

CloudFront > Distributions > Create

## Create distribution

### Origin

Origin domain

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

**Warning:** This S3 bucket has static web hosting enabled. If you plan to use this distribution as a website, we recommend using the S3 website endpoint rather than the bucket endpoint.

[Use website endpoint](#)

Origin path - optional

Enter a URL path to append to the origin domain name for origin requests.

Name

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Cloud Front Configuration

aws

Services

Search

[Alt+S]

☰

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

☐ HTTP/3

Default root object - optional

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

inde.html

Standard logging

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

☒ Off

☐ On

IPv6

☐ Off

CloudShell

Feedback

© 2024, Amazon

aws

Services

Search

[Alt+S]

☰

Origin path - optional

Enter a URL path to append to the origin domain name.

Enter the origin path

Name

Enter a name for this origin.

serverlesswebapp01.s3.us-east-1.amazonaws.com

Origin access

Info

☐ Public

Bucket must allow public access.

☒ Origin access control settings (recommended)

Bucket can restrict access to only CloudFront.

☐ Legacy access identities

Use a CloudFront origin access identity (OAI) to

Origin access control

Select an existing origin access control (recommended)

Select an origin access control

Add custom header - optional

CloudFront includes this header in all requests that

Add header

Enable Origin Shield

Origin shield is an additional caching layer that can

Create new OAC

X

Name

The name must be unique. Valid characters: letters, numbers and most special characters. Use up to 64 characters.

serverlesswebapp01.s3.us-east-1.amazonaws.com

Description - optional

The description can have up to 256 characters.

Enter description

Signing behavior

☐ Do not sign requests

☒ Sign requests (recommended)

☐ Do not override authorization header

Do not sign if incoming request has authorization header.

Origin type

S3

▼

The origin type must be the same type as origin domain.

Cancel

Create

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences



# Coping New policy after Creating Cloud Front Distribution

aws

Services

Search

[Alt+S]

Global

Salma\_Salah @ salma-salah

Introducing the CloudFront Security Dashboard

The new security tab is a unified place to configure, manage, and monitor security for your CloudFront distribution. The built-in dashboard gives you visibility into top security trends, allowed and blocked traffic, as well as visibility and controls for bots. CloudFront geographic restrictions are now part of the security dashboard.

Successfully created new distribution.

Policy statement copied

The S3 bucket policy needs to be updated

Complete distribution configuration by allowing read access to CloudFront origin access control in your policy statement. [Go to S3 bucket permissions to update policy](#)

Copy policy

CloudFront > Distributions > EFLCOMHJEQ1AW

EFLCOMHJEQ1AW

View metrics

GeneralSecurityOriginsBehaviorsError pagesInvalidationsTags

Details

Distribution domain name

d3tekucic0a239.cloudfront.net

ARN

Last modified

Deploying

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

# Block Public Access in S3 and adding new policy to make access only through CloudFront

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

Amazon S3

X

Buckets

Access Grants

Access Points

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

Amazon S3 > Buckets > serverlesswebapp01 > Edit Block public access (bucket settings)

Info Info

Edit Block public access (bucket settings)

Block public access (bucket settings)

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

☒ Block all public access

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

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

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

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

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

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

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

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

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

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

aws

Services

Search

[Alt+S]

N. Virginia

Salma\_Salah @ salma-salah

Amazon S3

X

Buckets

Access Grants

Access Points

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

Amazon S3 > Buckets > serverlesswebapp01 > Policy

Policy

```
1 {
2   "Version": "2008-10-17",
3   "Id": "PolicyForCloudFrontPrivateContent",
4   "Statement": [
5     {
6       "Sid": "AllowCloudFrontServicePrincipal",
7       "Effect": "Allow",
8       "Principal": {
9         "Service": "cloudfront.amazonaws.com"
10      },
11      "Action": "s3:GetObject",
12      "Resource": "arn:aws:s3:::serverlesswebapp01/*",
13      "Condition": {
14        "StringEquals": {
15          "AWS:SourceArn": "arn:aws:cloudfront::821594020462:distribution/EFLC"
16        }
17      }
18    }
19  ]
20 }
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

# Testing with Cloud Front End Point and now the connection is Secured (HTTPS)

Student Data

CloudFront | Global

serverlesswebapp01 - S3 bu

API Gateway - Resources

Student Data

https://d3tekucic0a239.cloudfront.net

Gmail (2602) AWS SAA-C0... AWS Certified Soluti... Console Home | Co... AZ-900: Microsoft A... Introduction - Kode... Oracle OCI Training... AZ-900 Exam - Free... All Bookmarks

## Salma Salah to Save and View Data of Students

Student ID:

Name:

Class:

Age:

Save Student Data

Student Data Saved!

View all Students

Student ID	Name	Class	Age
2	Ahmed	B	25
1	Salma	A	24
4	Yara	D	21
2	Nour	C	26