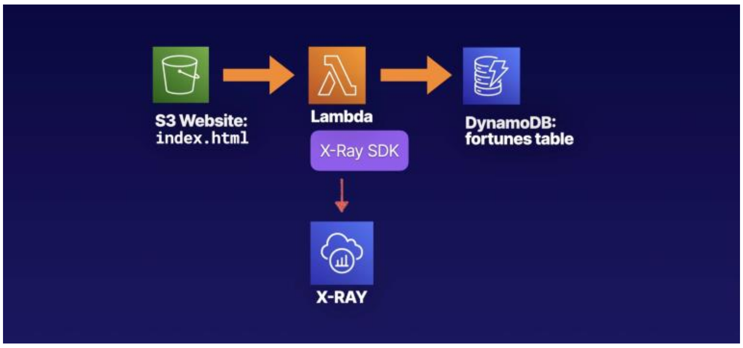
Serverless Web App (AWS Lambda + API Gateway + DynamoDB).

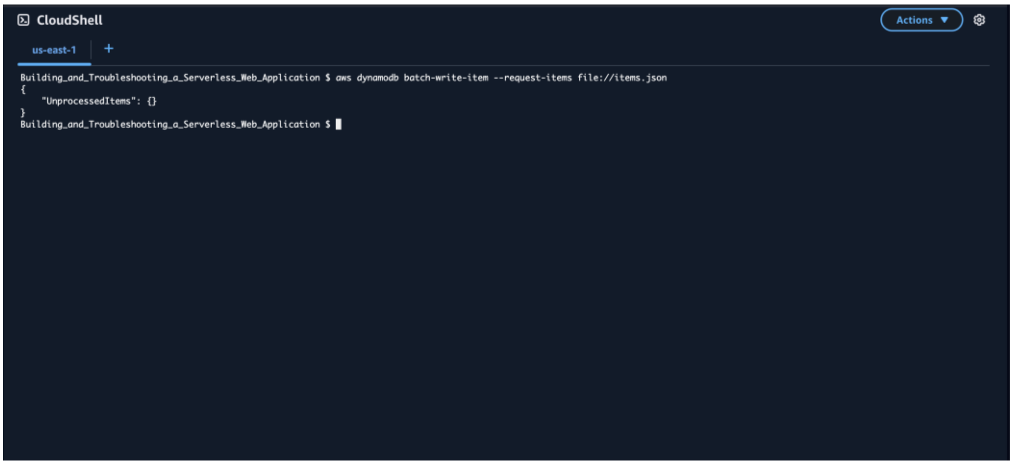
Objectives

Successfully complete this lab by achieving the following learning objectives:

Create a DynamoDB Table

1. Create the DynamoDB table and populate it using the file named items.json. All files required for this hands-on lab are located in the folder named Building_and_Troubleshooting_a_Serverless_Web_Application that you will find in the following GitHub repository.

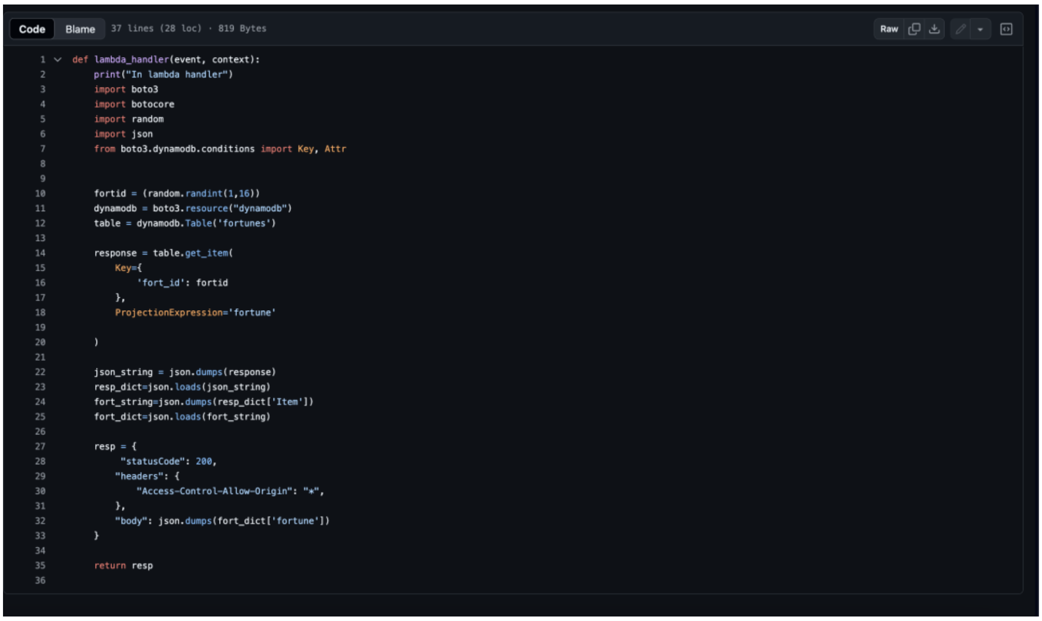




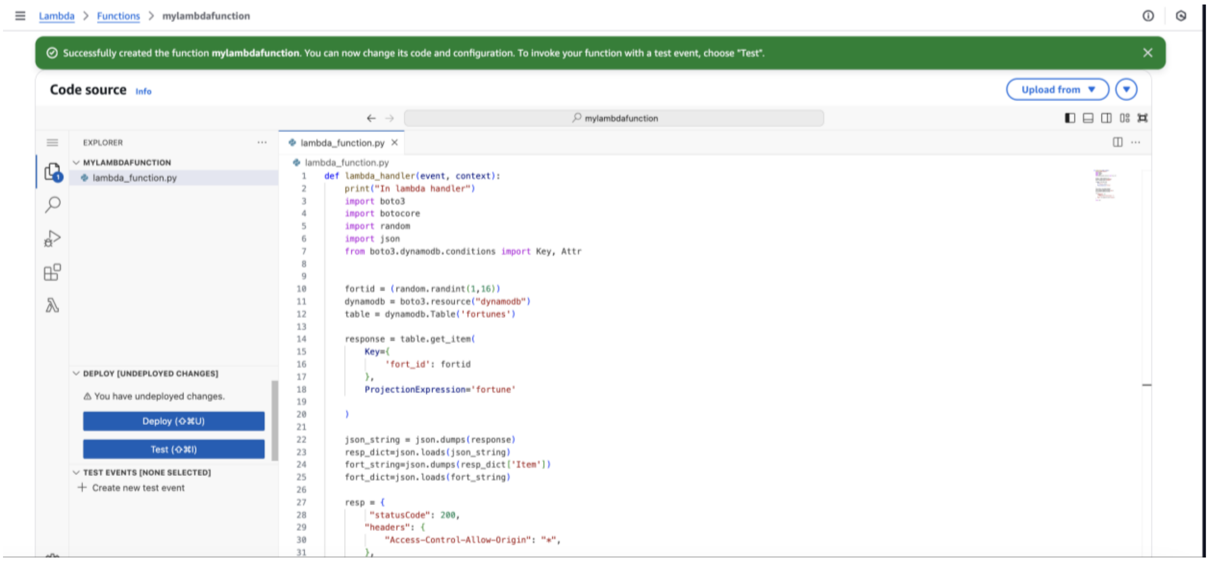
```
Building_and_Troubleshooting_a_Serverless_Web_Application $ aws dynamodb batch-write-item --request-items file://items.json
{
    "UnprocessedItems": {}
}
Building_and_Troubleshooting_a_Serverless_Web_Application $ 
```

Create a Lambda Function with a Function Endpoint

1. Create a Lambda function usinglambda_function.py.
2. Configure the Lambda to use Python 3.9 and 256 MB.
3. Test the function to see if it works.
4. Diagnose and fix an error with the function configuration.
5. Create and test a function URL endpoint once the function's error has been corrected.

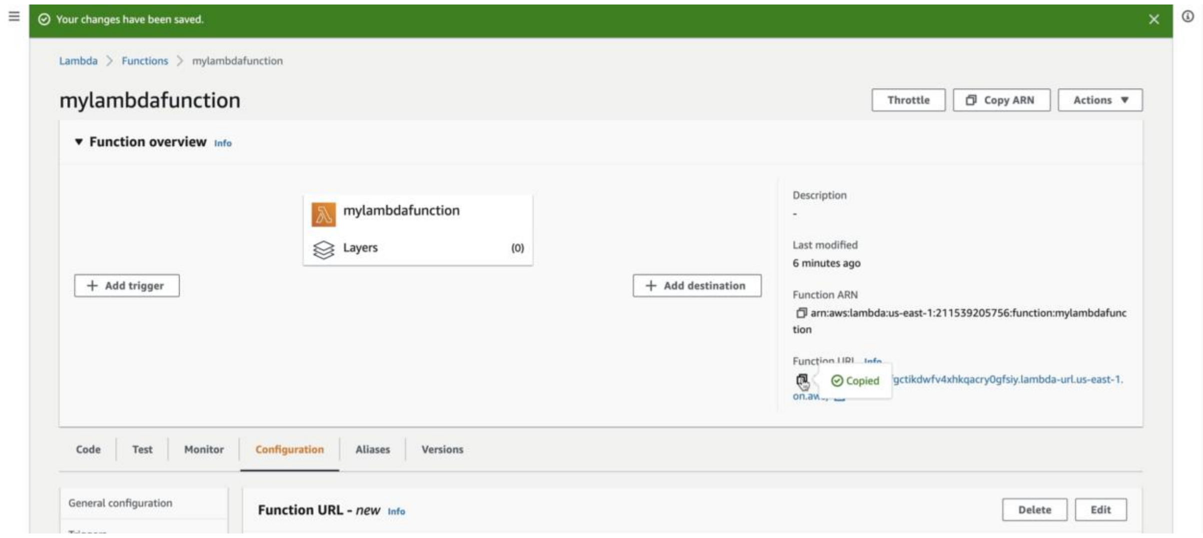


```
def lambda_handler(event, context):
    print("In lambda handler")
    import boto3
    import botocore
    import random
    import json
    from boto3.dynamodb.conditions import Key, Attr


    fortid = (random.randint(1,16))
    dynamodb = boto3.resource("dynamodb")
    table = dynamodb.Table('fortunes')

    response = table.get_item(
        Key={
            'fort_id': fortid
        },
        ProjectionExpression='fortune'

    )

    json_string = json.dumps(response)
    resp_dict=json.loads(json_string)
    fort_string=json.dumps(resp_dict['Item'])
    fort_dict=json.loads(fort_string)

    resp = {
         "statusCode": 200,
        "headers": {
            "Access-Control-Allow-Origin": "*",
        },
        "body": json.dumps(fort_dict['fortune'])
    }

    return resp

```



Successfully created the function **mylambdafunction**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

```
def lambda_handler(event, context):
    print("In lambda handler")
    import boto3
    import botocore
    import random
    import json
    from boto3.dynamodb.conditions import Key, Attr


    fortid = (random.randint(1,16))
    dynamodb = boto3.resource("dynamodb")
    table = dynamodb.Table('fortunes')

    response = table.get_item(
        Key={
            'fort_id': fortid
        },
        ProjectionExpression='fortune'

    )

    json_string = json.dumps(response)
    resp_dict=json.loads(json_string)
    fort_string=json.dumps(resp_dict['Item'])
    fort_dict=json.loads(fort_string)

    resp = {
         "statusCode": 200,
        "headers": {
            "Access-Control-Allow-Origin": "*",
        },
```

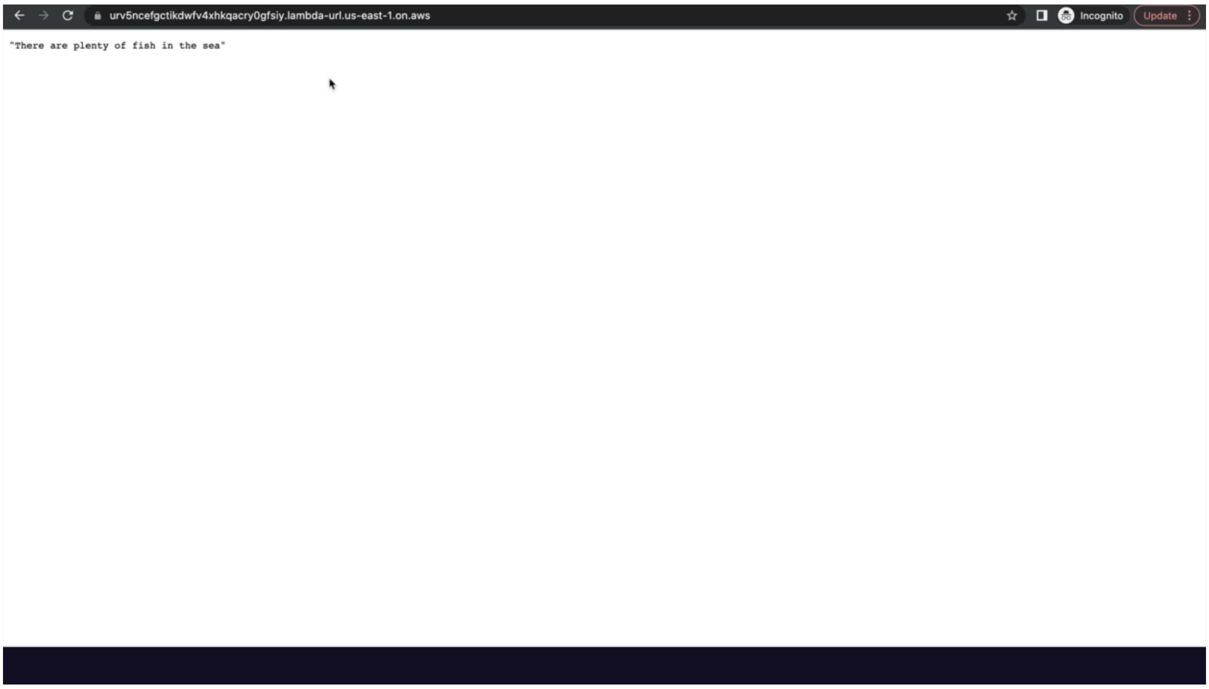

Your changes have been saved.

Description
-

Last modified
6 minutes ago

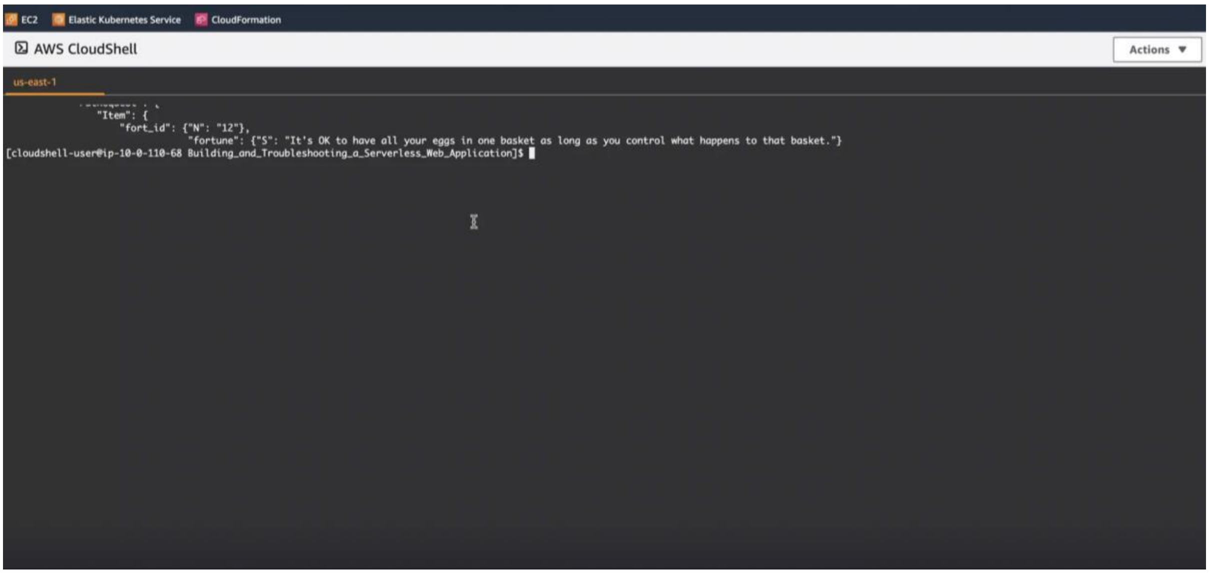
Function ARN
arn:aws:lambda:us-east-1:211539205756:function:mylambdafunction

Function URL Info
Copied gctikdwfv4xhkqacry0gfsiy.lambda-url.us-east-1.on.aw

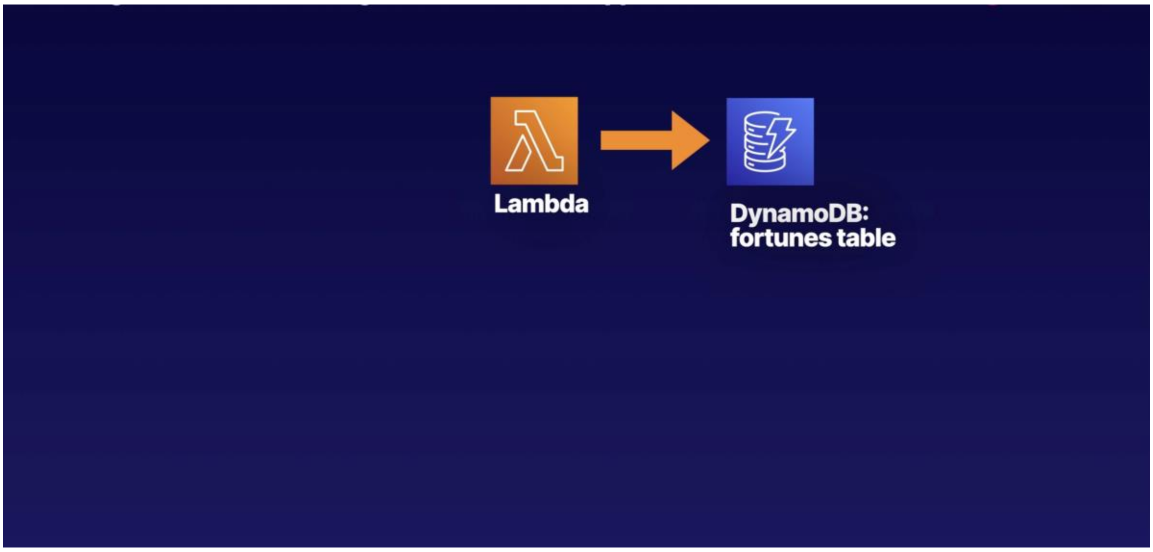


urv5ncefgctikdwfv4xhkqacry0gfsiy.lambda-url.us-east-1.on.aws
Incognito
Update

```
"There are plenty of fish in the sea"
```



```
            "Item": {
                "fort_id": {"N": "12"},
                                "fortune": {"S": "It's OK to have all your eggs in one basket as long as you control what happens to that basket."}
[cloudshell-user@ip-10-0-110-68 Building_and_Troubleshooting_a_Serverless_Web_Application]$ 
```



Create an S3 Bucket and Upload the Website Files

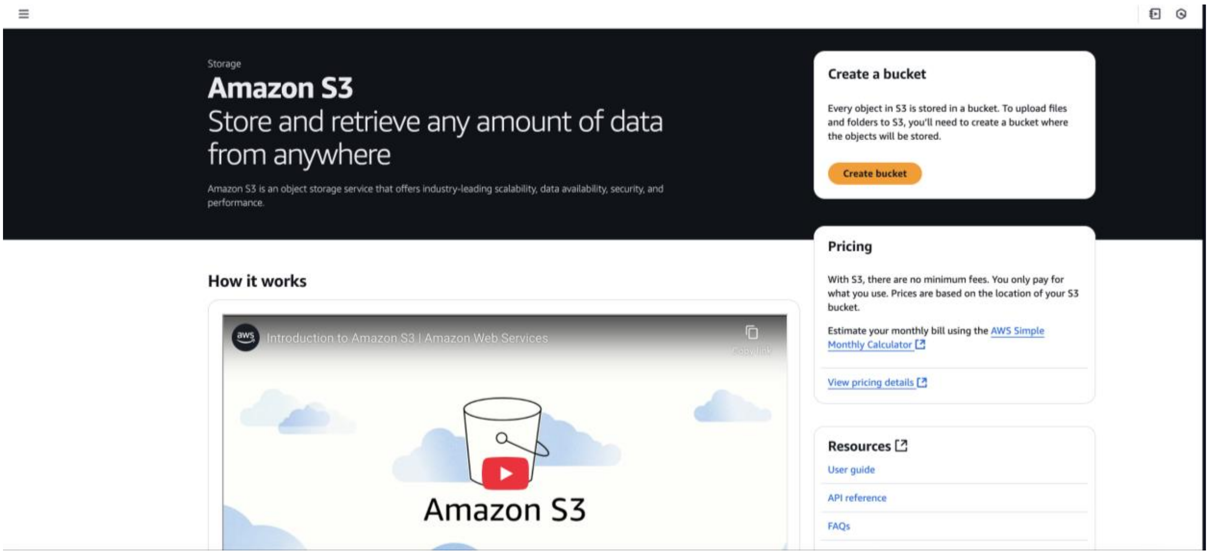
1. Modify index.html to add the invoke URL of your API Gateway endpoint.

2. Create an S3 bucket with public access enabled.

3. Upload the following website files to your bucket. Make sure they have public-read enabled: index.html, error.html, and cookie.html

4. Configure S3 static website hosting on your S3 bucket.

5. Check you can access everything from the S3 website-hosting URL.



Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

How it works

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the AWS Simple Monthly Calculator

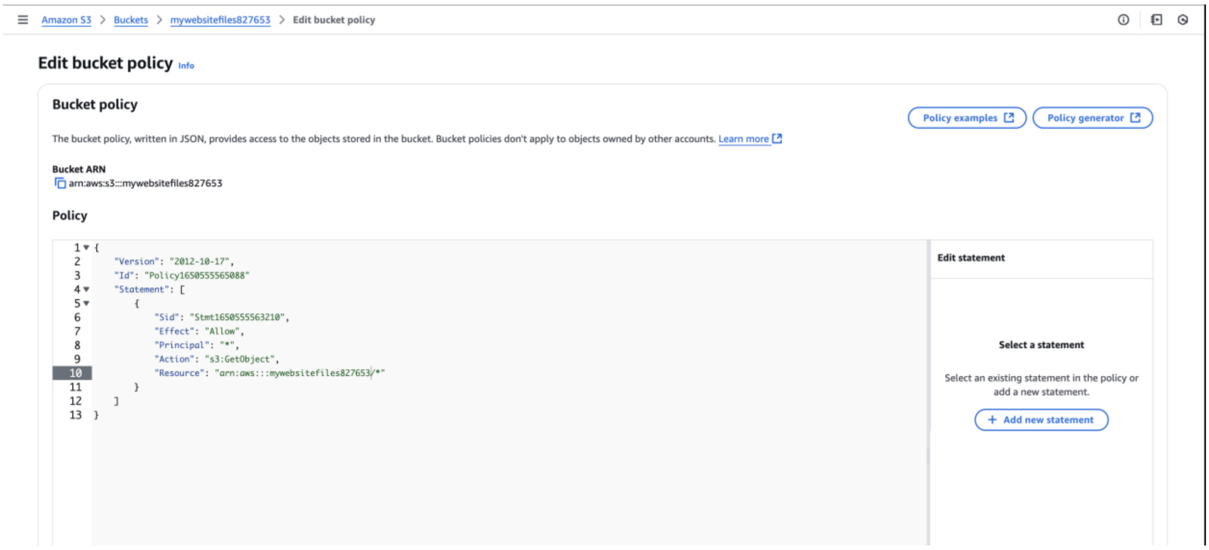
View pricing details

Resources

User guide

API reference

FAQs



Amazon S3 > Buckets > mywebsitefiles827653 > Edit bucket policy

Edit bucket policy Info

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:::mywebsitefiles827653

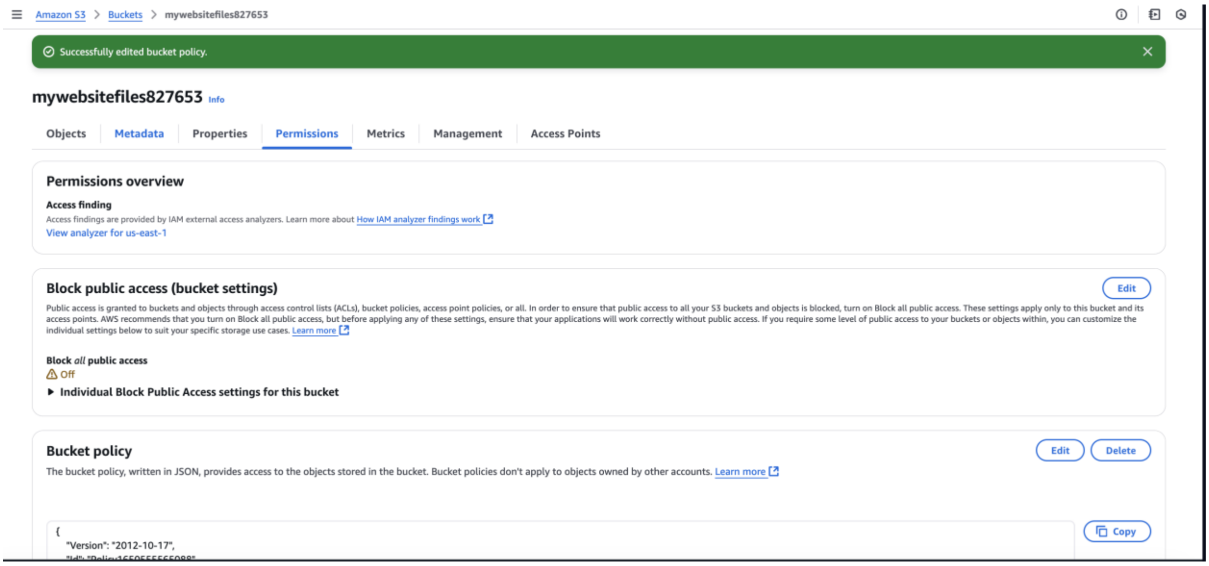
Policy

```
{
    "Version": "2012-10-17",
    "Id": "Policy1650555565088"
    "Statement": [
        {
            "Sid": "Stmt1650555563210",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:::mywebsitefiles827653/*"
        }
    ]
}
```

Edit statement

Select a statement

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



Successfully edited bucket policy.

mywebsitefiles827653 Info

Permissions overview

Access finding

Access findings are provided by IAM external access analyzers. Learn more about How IAM analyzer findings work

View analyzer for us-east-1

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

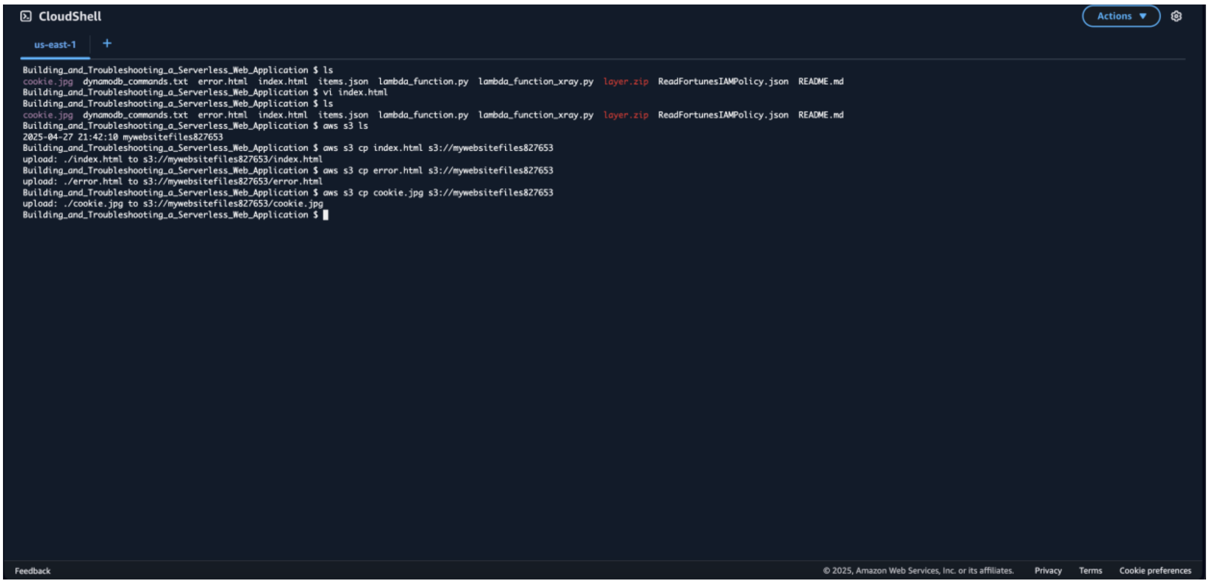
Block *all* public access
⚠ Off

▸ **Individual Block Public Access settings for this bucket**

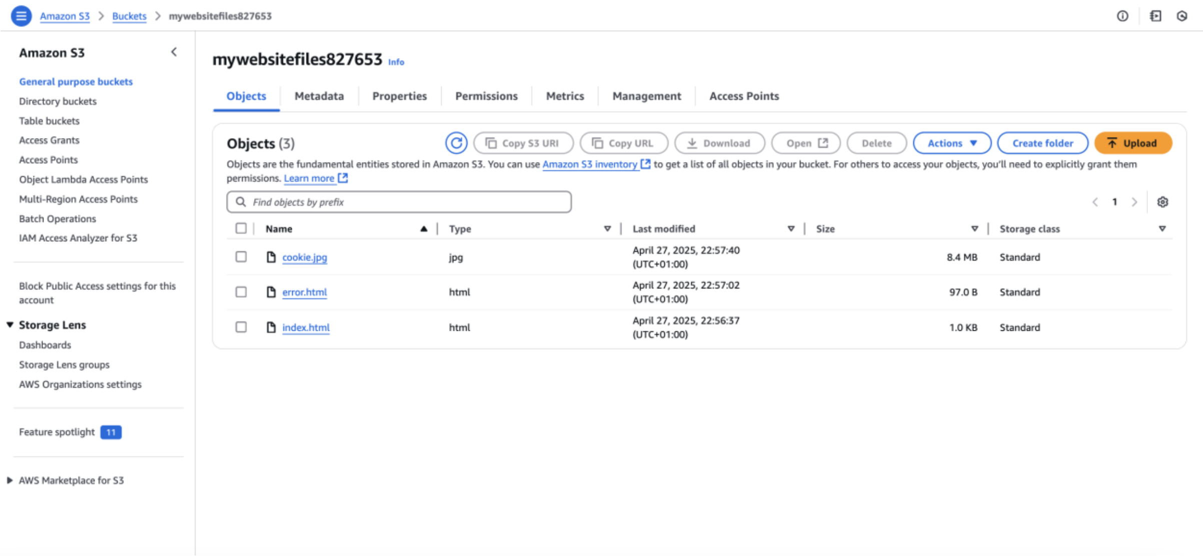
Bucket policy

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",
```



```
Building_and_Troubleshooting_a_Serverless_Web_Application $ ls
cookie.jpg  dynamodb_commands.txt  error.html  index.html  items.json  lambda_function.py  lambda_function_xray.py  layer.zip  ReadFortunesIAMPolicy.json  README.md
Building_and_Troubleshooting_a_Serverless_Web_Application $ vi index.html
Building_and_Troubleshooting_a_Serverless_Web_Application $ ls
cookie.jpg  dynamodb_commands.txt  error.html  index.html  items.json  lambda_function.py  lambda_function_xray.py  layer.zip  ReadFortunesIAMPolicy.json  README.md
Building_and_Troubleshooting_a_Serverless_Web_Application $ aws s3 ls
2025-04-27 21:42:10 mywebsitefiles827653
Building_and_Troubleshooting_a_Serverless_Web_Application $ aws s3 cp index.html s3://mywebsitefiles827653
upload: ./index.html to s3://mywebsitefiles827653/index.html
Building_and_Troubleshooting_a_Serverless_Web_Application $ aws s3 cp error.html s3://mywebsitefiles827653
upload: ./error.html to s3://mywebsitefiles827653/error.html
Building_and_Troubleshooting_a_Serverless_Web_Application $ aws s3 cp cookie.jpg s3://mywebsitefiles827653
upload: ./cookie.jpg to s3://mywebsitefiles827653/cookie.jpg
Building_and_Troubleshooting_a_Serverless_Web_Application $ █
```



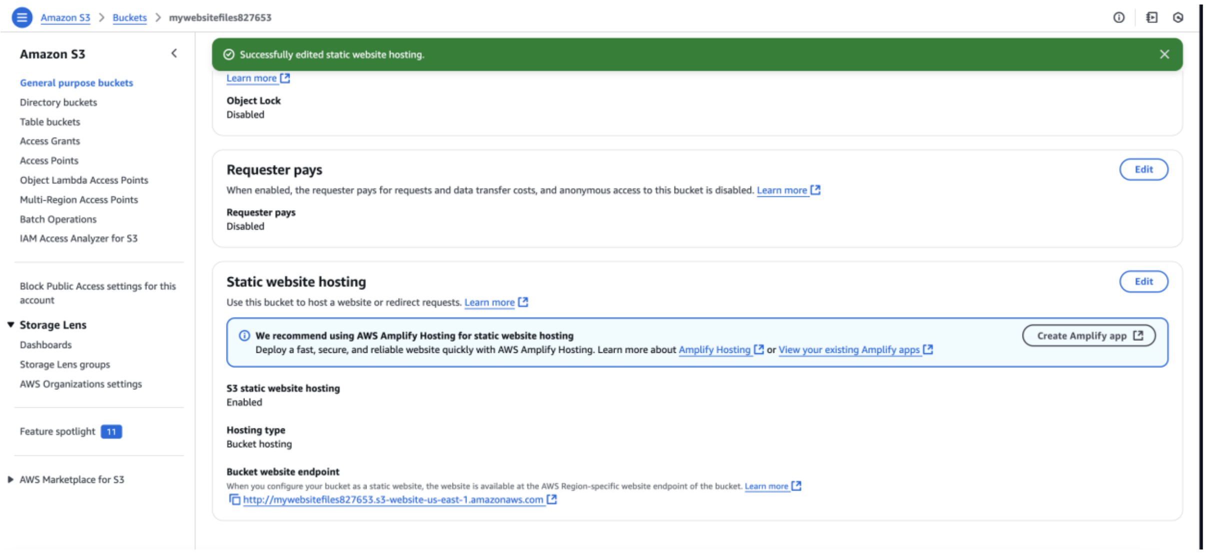
mywebsitefiles827653 Info

Objects (3)

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

Find objects by prefix

□	Name	Type	Last modified	Size	Storage class
□	cookie.jpg	jpg	April 27, 2025, 22:57:40 (UTC+01:00)	8.4 MB	Standard
□	error.html	html	April 27, 2025, 22:57:02 (UTC+01:00)	97.0 B	Standard
□	index.html	html	April 27, 2025, 22:56:37 (UTC+01:00)	1.0 KB	Standard



Amazon S3 > Buckets > mywebsitefiles827653

Successfully edited static website hosting.

Learn more

Object Lock
Disabled

Requester pays

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

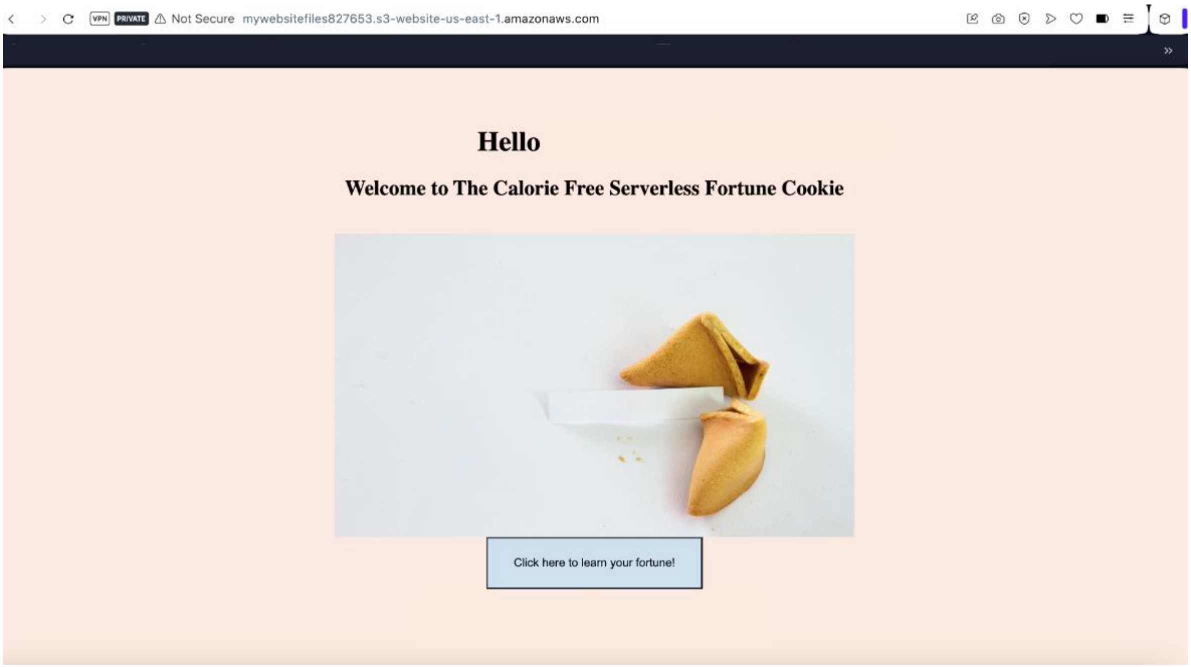
Use this bucket to host a website or redirect requests. Learn more

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

S3 static website hosting
Enabled

Hosting type
Bucket hosting

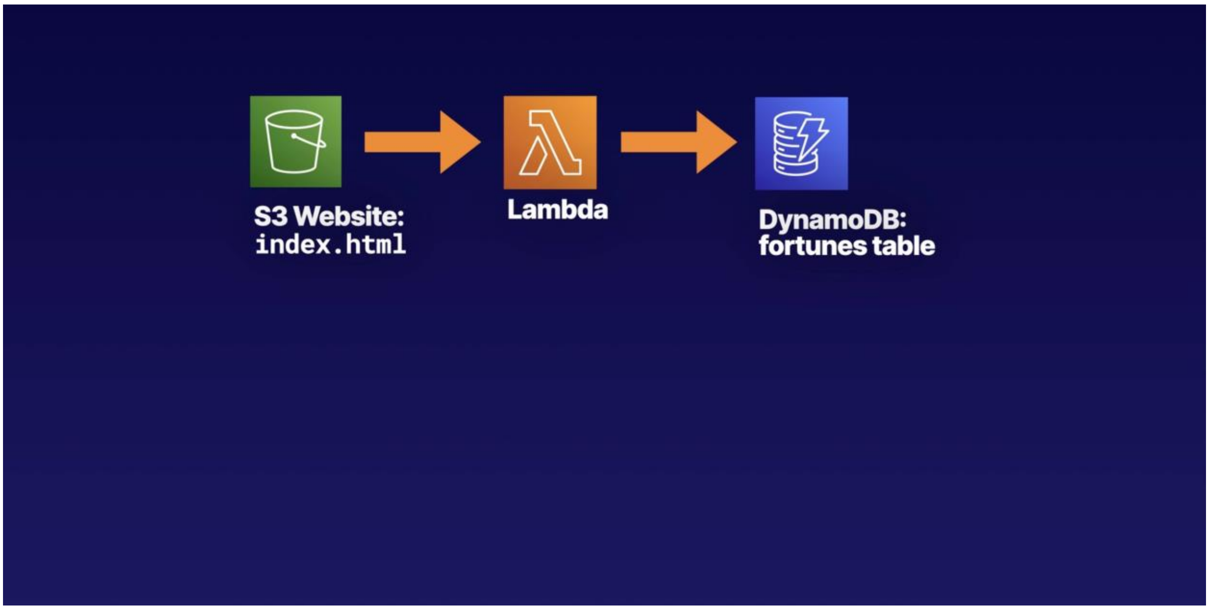
Bucket website endpoint
When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. Learn more
http://mywebsitefiles827653.s3-website-us-east-1.amazonaws.com



Not Secure mywebsitefiles827653.s3-website-us-east-1.amazonaws.com

Hello

Welcome to The Calorie Free Serverless Fortune Cookie



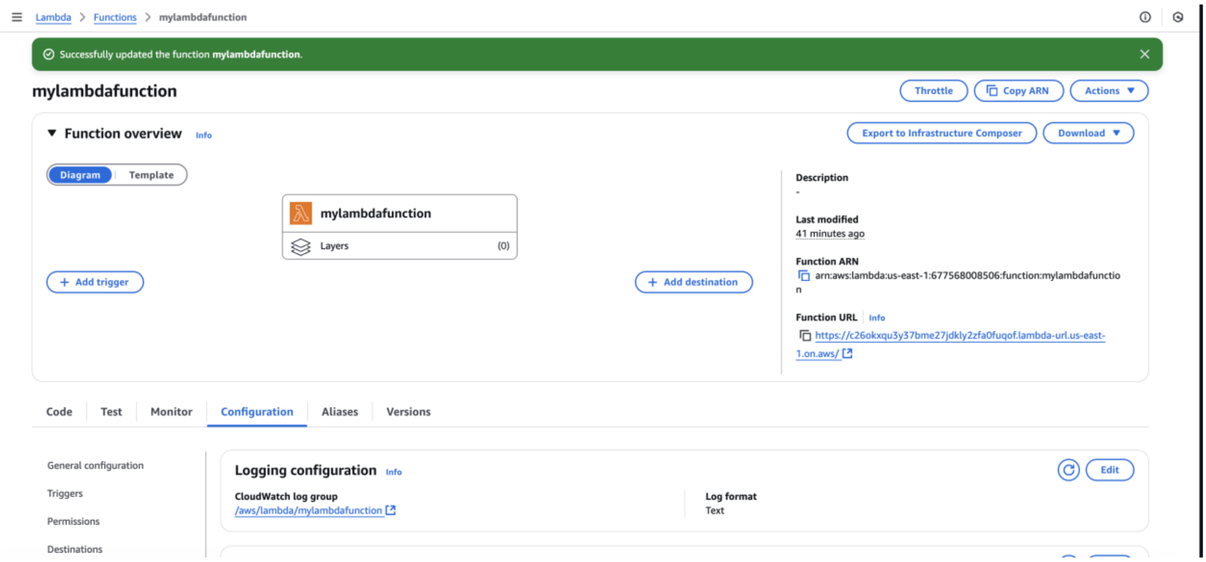
S3 Website: index.html

Lambda

DynamoDB: fortunes table

Configure X-Ray

1. Enable tracing from within the Lambda function.
2. Enable the API gateway endpoint to send traces.
3. Add the Lambda Layer to your function using the file named layer.zip Make sure to use Python 3.9.
4. Update your Lambda function code to import the X-Ray SDK to your function using the snippet provided named lambda_function_xray.py.
5. Test the function to see if it works.
6. Diagnose and fix an error with the function configuration.
7. Review The X-Ray Service Map



Successfully updated the function **mylambdafunction**.

mylambdafunction

Function overview Info

mylambdafunction

Description
-

Last modified
41 minutes ago

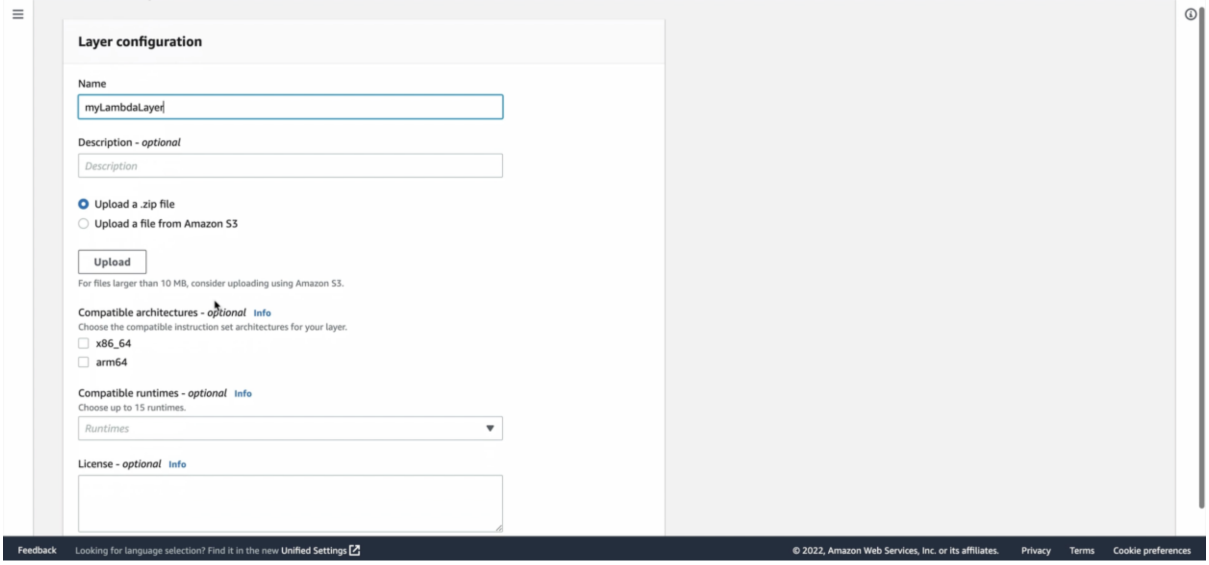
Function ARN
arn:aws:lambda:us-east-1:677568008506:function:mylambdafunction

Function URL Info
https://c26okxqu3y37bme27jdkly2zfa0fuqof.lambda-url.us-east-1.on.aws/

Logging configuration Info

CloudWatch log group: /aws/lambda/mylambdafunction

Log format: Text


Upload
Info
Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Layer configuration

Name
myLambdaLayer

Description - *optional*
Description

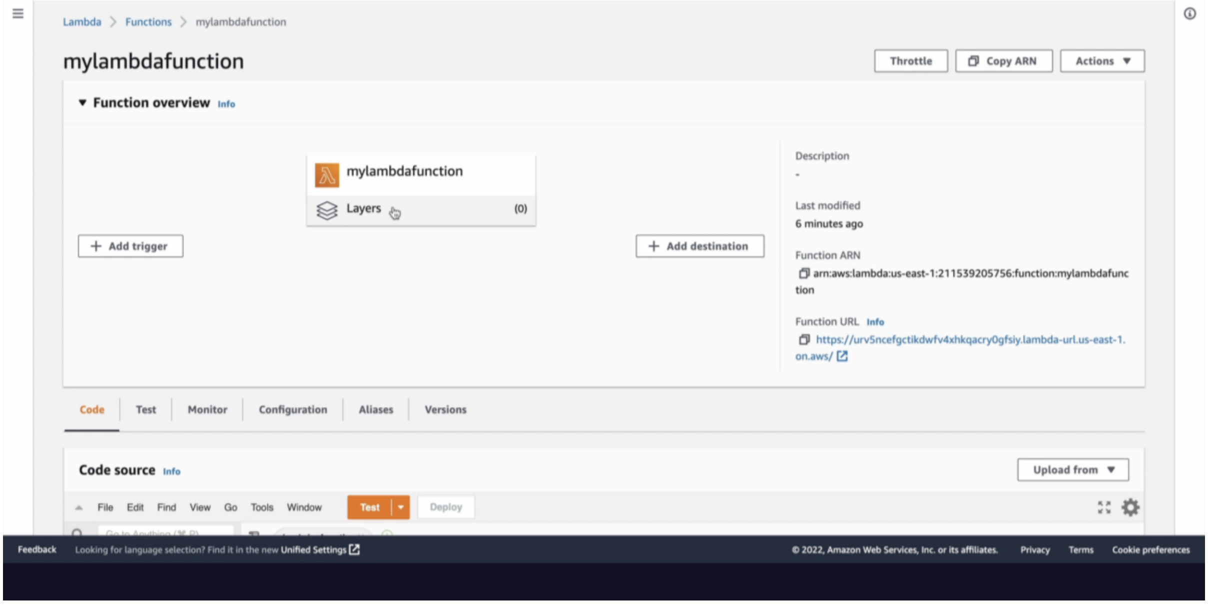
● Upload a .zip file
○ Upload a file from Amazon S3

For files larger than 10 MB, consider uploading using Amazon S3.

Compatible architectures - *optional*
Choose the compatible instruction set architectures for your layer.
□ x86_64
□ arm64

Compatible runtimes - *optional*
Choose up to 15 runtimes.
Runtimes

License - *optional*



mylambdafunction

Description
-

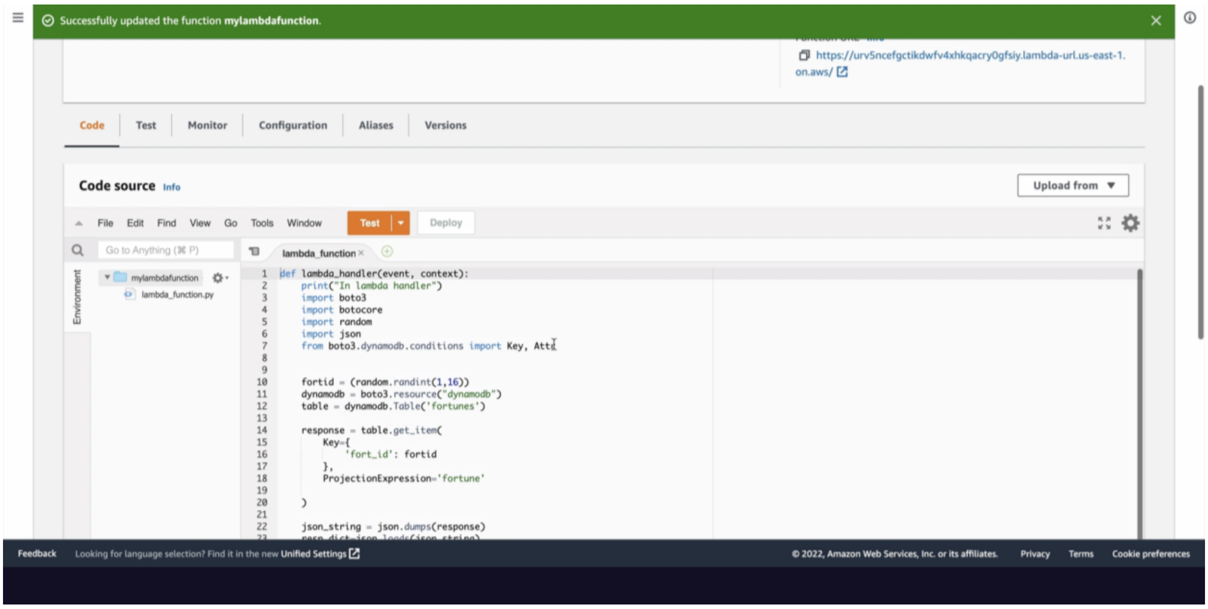
Last modified
6 minutes ago

Function ARN
arn:aws:lambda:us-east-1:211539205756:function:mylambdafunction

Function URL Info
https://urv5ncefgctikdwfv4xhkqacry0gfsiy.lambda-url.us-east-1.on.aws/

Looking for language selection? Find it in the new **Unified Settings**

© 2022, Amazon Web Services, Inc. or its affiliates.



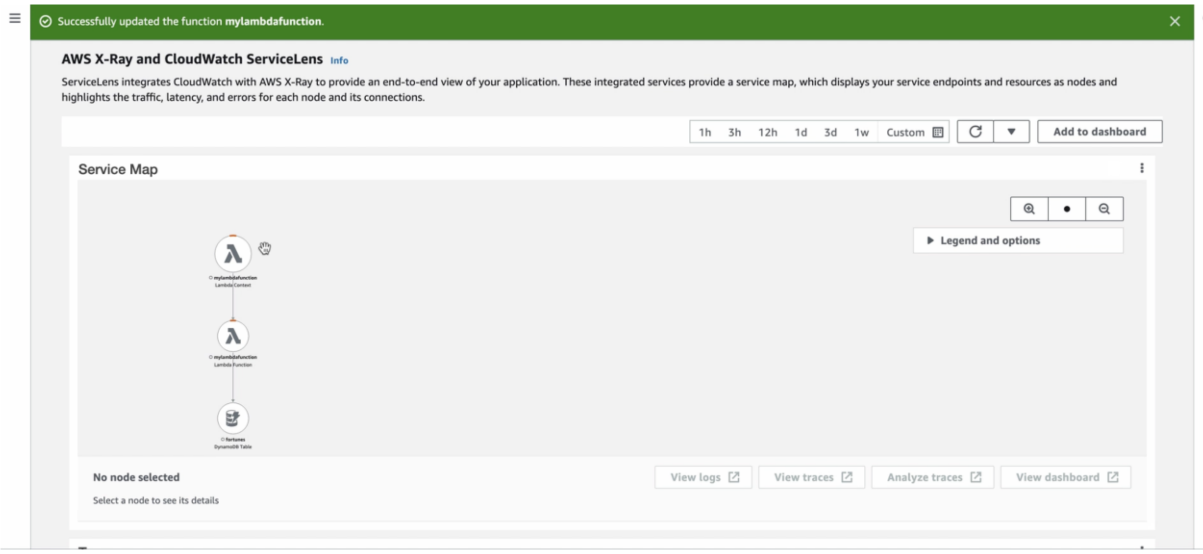
Successfully updated the function **mylambdafunction**.

https://urv5ncefgctikdwfv4xhkqacry0gfsiy.lambda-url.us-east-1.on.aws/

```python
def lambda_handler(event, context):
    print("In lambda handler")
    import boto3
    import botocore
    import random
    import json
    from boto3.dynamodb.conditions import Key, Att


    fortid = (random.randint(1,16))
    dynamodb = boto3.resource("dynamodb")
    table = dynamodb.Table('fortunes')

    response = table.get_item(
        Key={
            'fort_id': fortid
        },
        ProjectionExpression='fortune'

    )

    json_string = json.dumps(response)
    resp_dict=json.loads(json_string)
```

Looking for language selection? Find it in the new **Unified Settings**

© 2022, Amazon Web Services, Inc. or its affiliates.



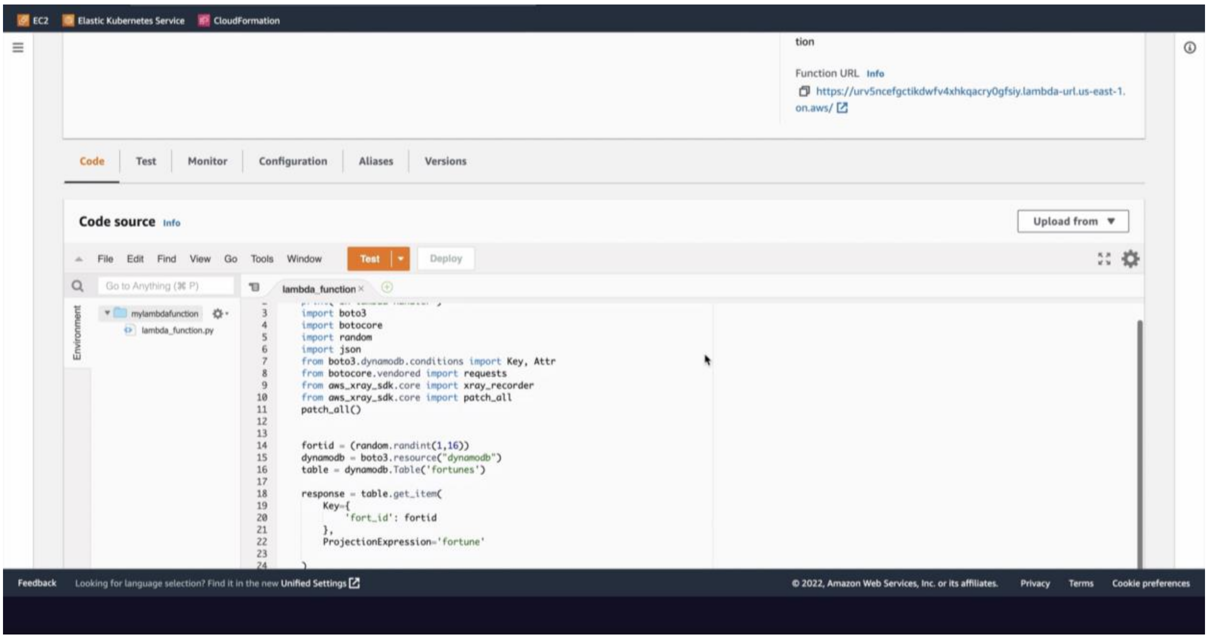
Successfully updated the function **mylambdafunction**.

AWS X-Ray and CloudWatch ServiceLens Info

ServiceLens integrates CloudWatch with AWS X-Ray to provide an end-to-end view of your application. These integrated services provide a service map, which displays your service endpoints and resources as nodes and highlights the traffic, latency, and errors for each node and its connections.

No node selected

Select a node to see its details

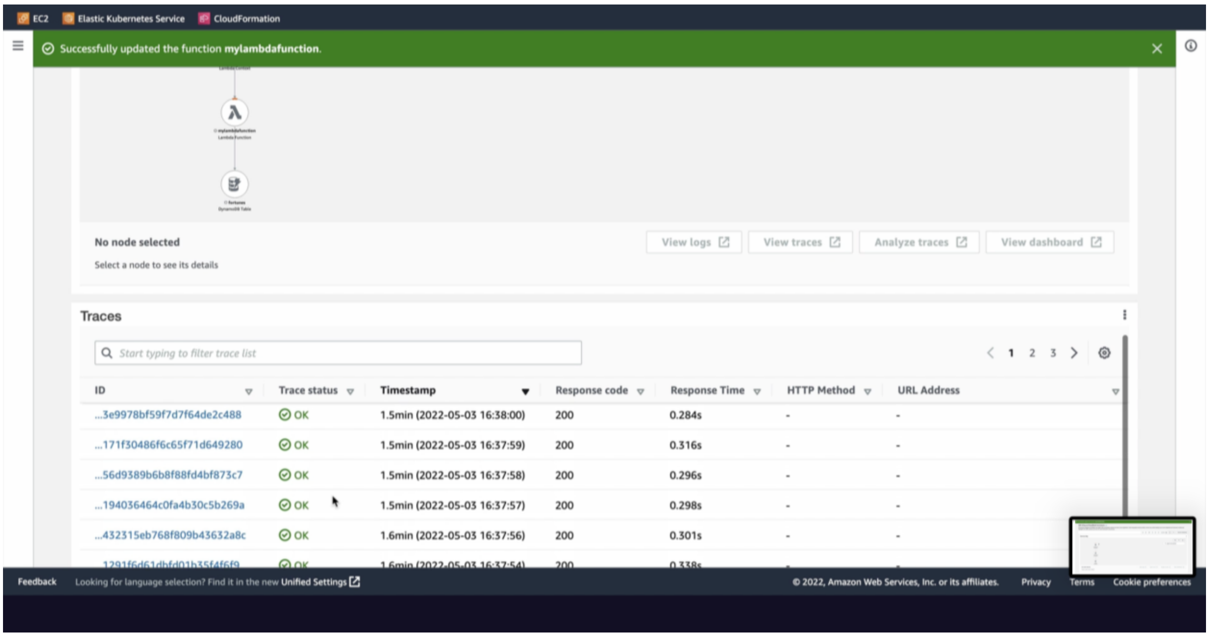


Function URL Info

https://urv5ncefgctikdwfv4xhkqacry0gfsiy.lambda-url.us-east-1.on.aws/

```
import boto3
import botocore
import random
import json
from boto3.dynamodb.conditions import Key, Attr
from botocore.vendored import requests
from aws_xray_sdk.core import xray_recorder
from aws_xray_sdk.core import patch_all
patch_all()


fortid = (random.randint(1,16))
dynamodb = boto3.resource("dynamodb")
table = dynamodb.Table('fortunes')

response = table.get_item(
    Key={
        'fort_id': fortid
    },
    ProjectionExpression='fortune'

)
```

Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


EC2
Elastic Kubernetes Service
CloudFormation

Successfully updated the function **mylambdafunction**.

mylambdafunction
Lambda function
fortunes
DynamoDB Table

No node selected

Select a node to see its details

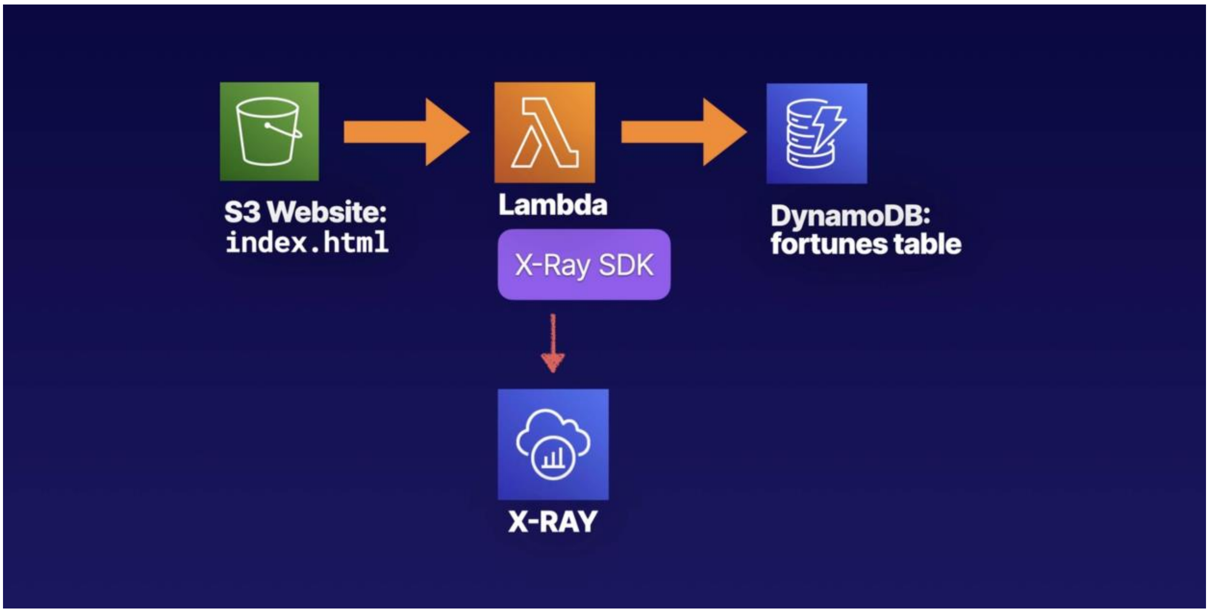
View logs
View traces
Analyze traces
View dashboard

Traces

Start typing to filter trace list
1
2
3

ID	Trace status	Timestamp	Response code	Response Time	HTTP Method	URL Address
...3e9978bf59f7d7f64de2c488	OK	1.5min (2022-05-03 16:38:00)	200	0.284s	-	-
...171f30486f6c65f71d649280	OK	1.5min (2022-05-03 16:37:59)	200	0.316s	-	-
...56d9389b6b8f88fd4bf873c7	OK	1.5min (2022-05-03 16:37:58)	200	0.296s	-	-
...194036464c0fa4b30c5b269a	OK	1.5min (2022-05-03 16:37:57)	200	0.298s	-	-
...432315eb768f809b43632a8c	OK	1.6min (2022-05-03 16:37:56)	200	0.301s	-	-
...1291f6d61dbfd01b35f4f6f9	OK	1.6min (2022-05-03 16:37:54)	200	0.338s	-	-

Feedback
Looking for language selection? Find it in the new Unified Settings
© 2022, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



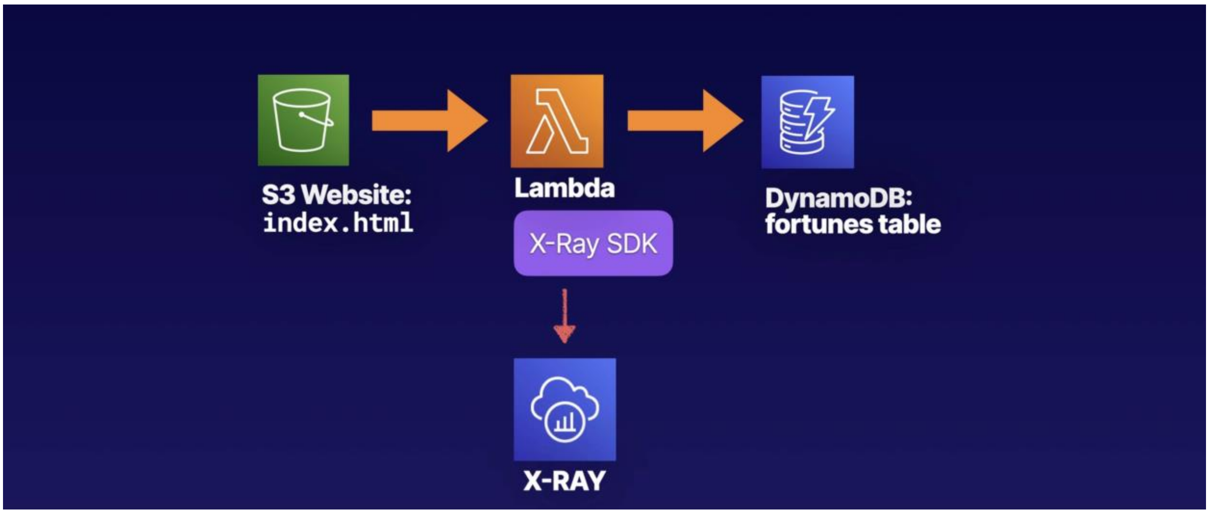
S3 Website:
`index.html`

Lambda

X-Ray SDK

DynamoDB:
fortunes table

X-RAY



S3 Website:
`index.html`

Lambda

X-Ray SDK

DynamoDB:
fortunes table

X-RAY