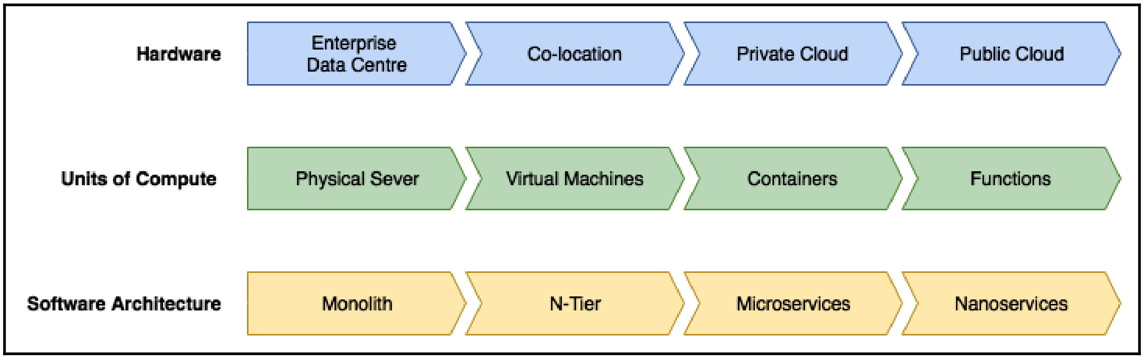
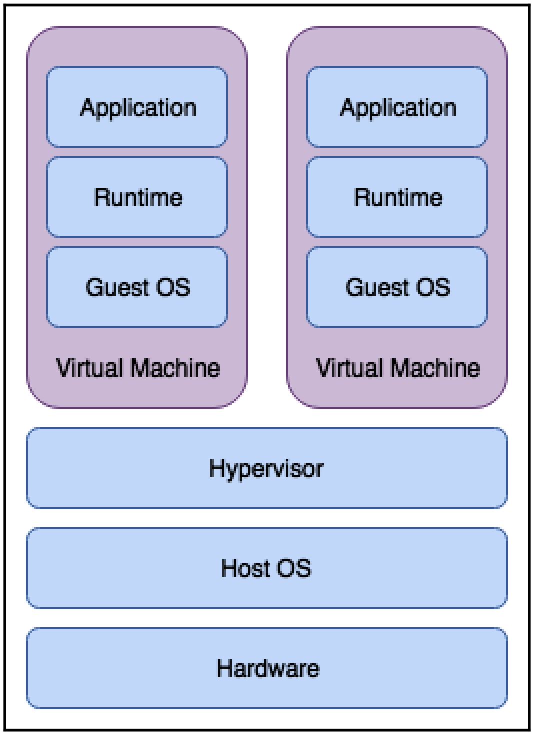
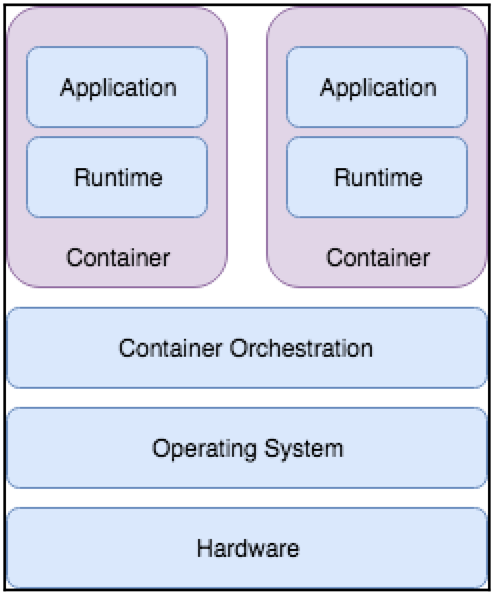
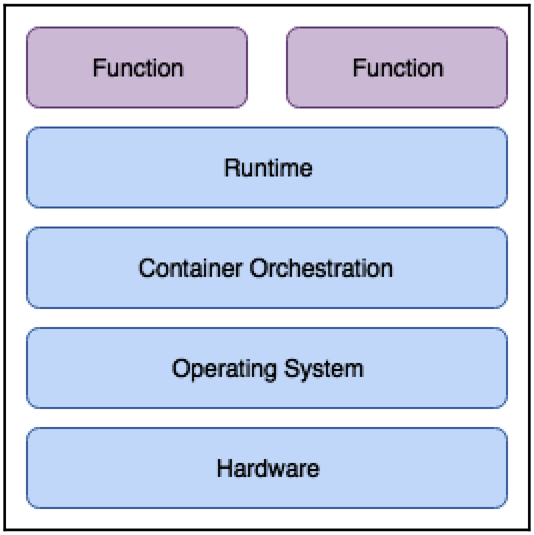
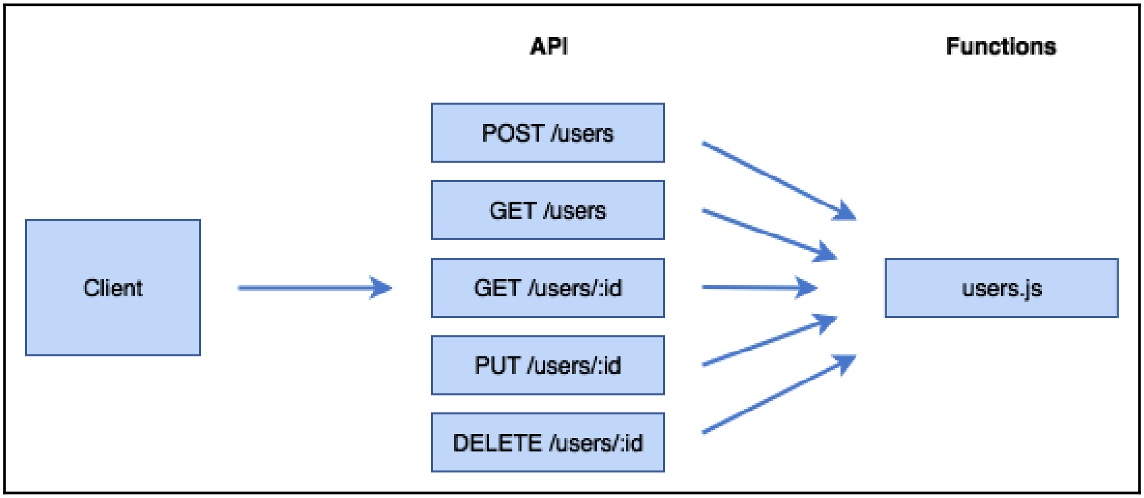
Chapter 1: The Evolution of Compute

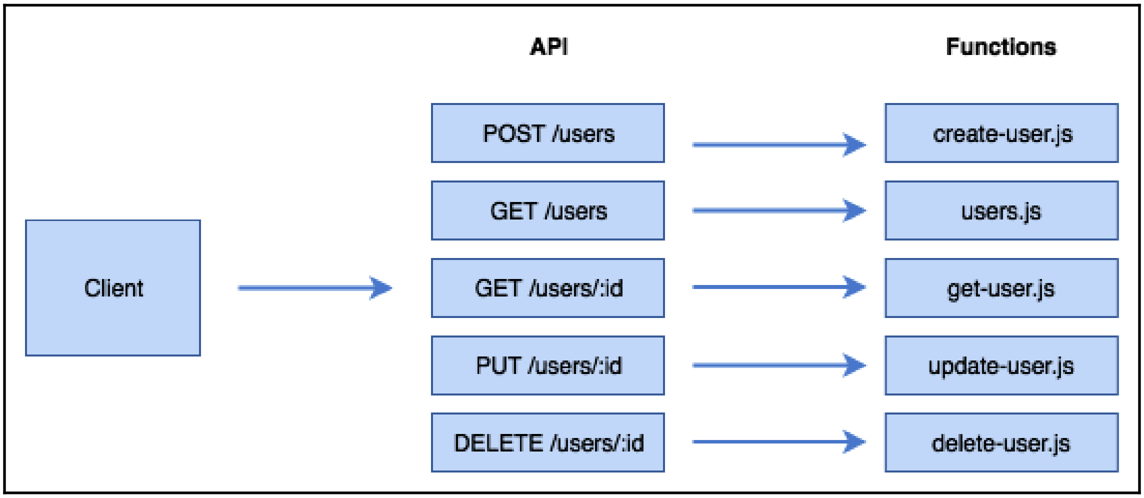




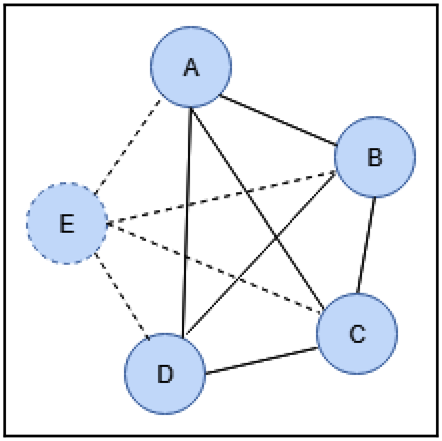

Application
Application
Runtime
Runtime
Container
Container
Container Orchestration
Operating System
Hardware

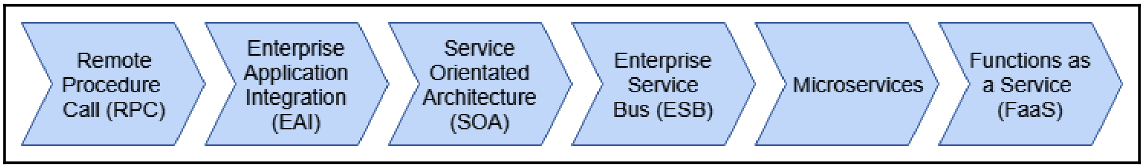

Function
Function
Runtime
Container Orchestration
Operating System
Hardware

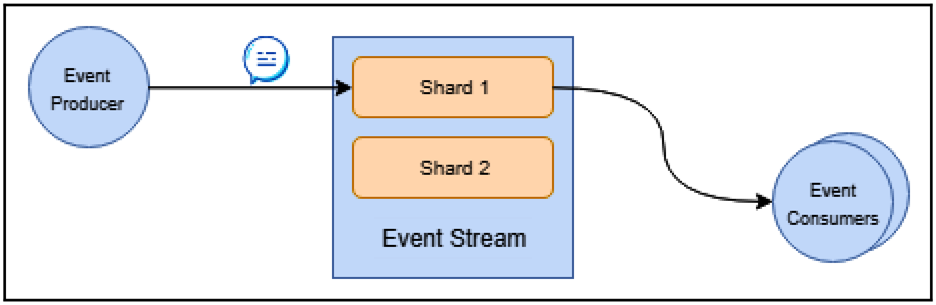

API
Functions
Client
POST /users
GET /users
GET /users/:id
PUT /users/:id
DELETE /users/:id
users.js

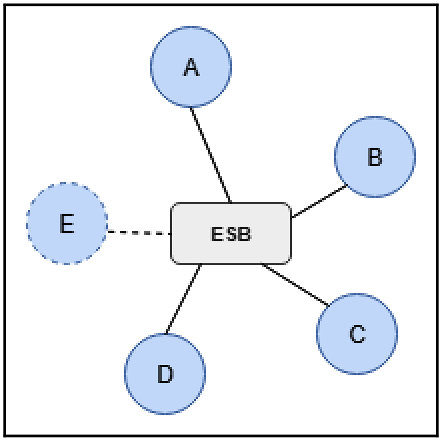

API
Functions
Client
POST /users
GET /users
GET /users/:id
PUT /users/:id
DELETE /users/:id
create-user.js
users.js
get-user.js
update-user.js
delete-user.js

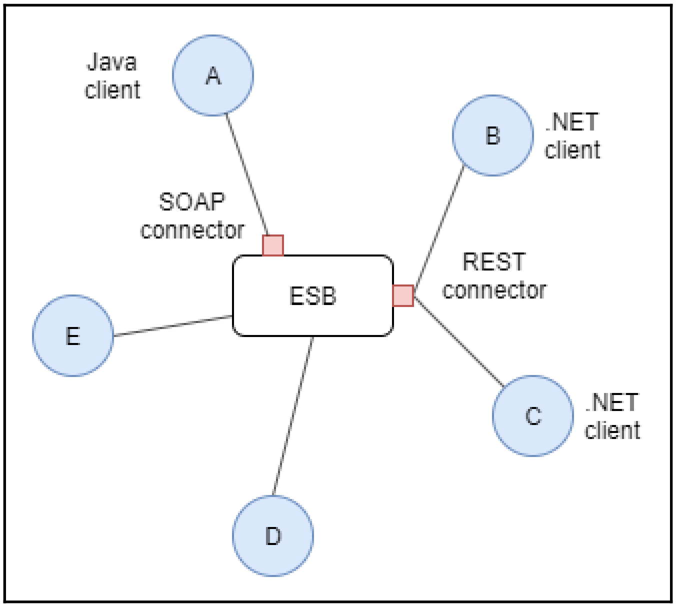
Chapter 2: Event-Driven Applications

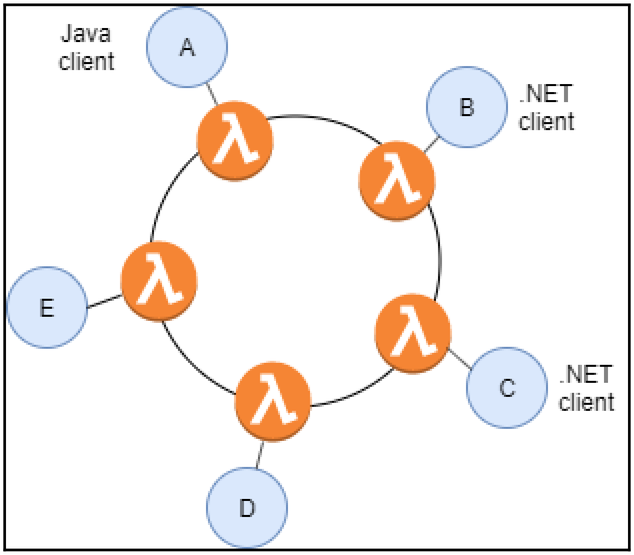


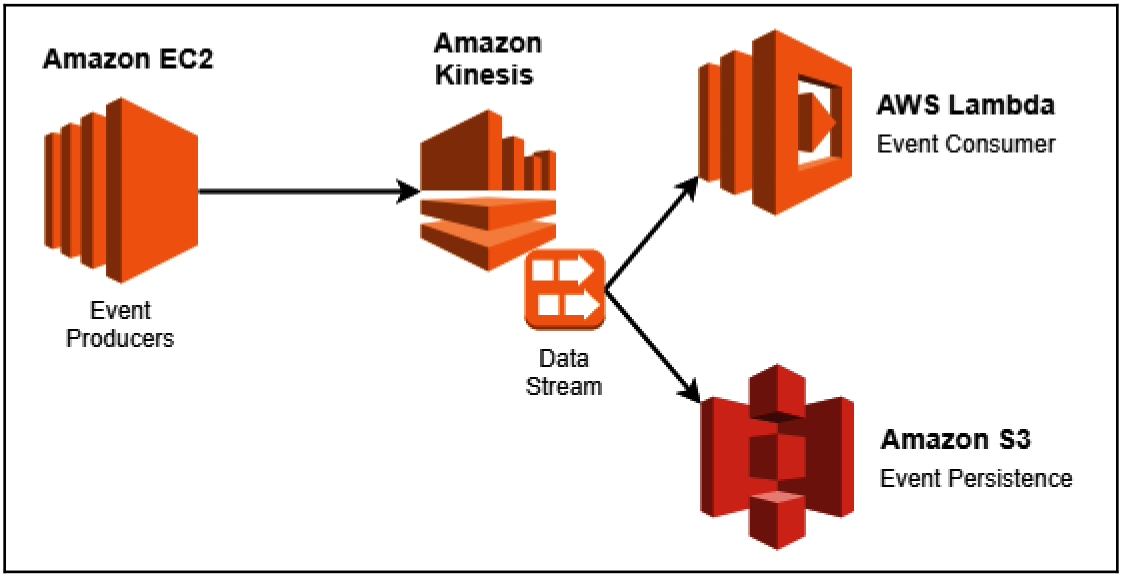

Remote Procedure Call (RPC)
Enterprise Application Integration (EAI)
Service Orientated Architecture (SOA)
Enterprise Service Bus (ESB)
Microservices
Functions as a Service (FaaS)



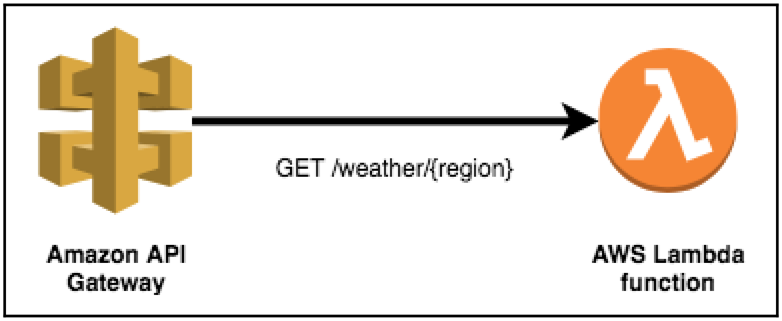

A
B
E
ESB
C
D

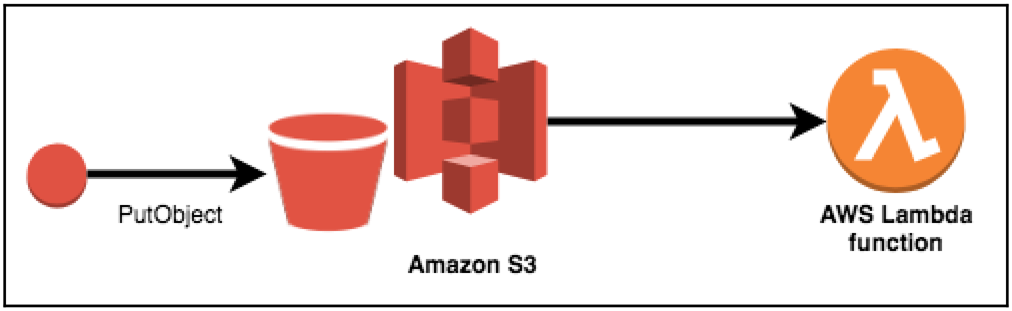

Java client
A
B
.NET client
SOAP connector
REST connector
E
ESB
C
.NET client
D

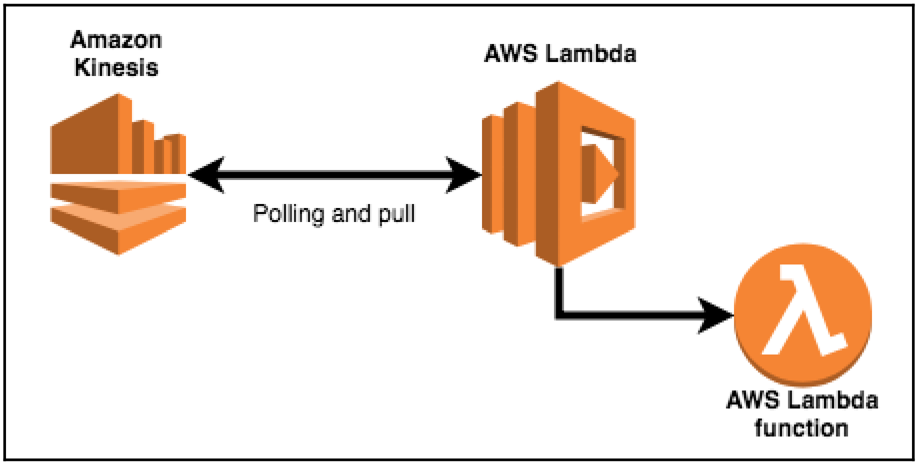

Java client
A
.NET client
B
.NET client
C
D
E

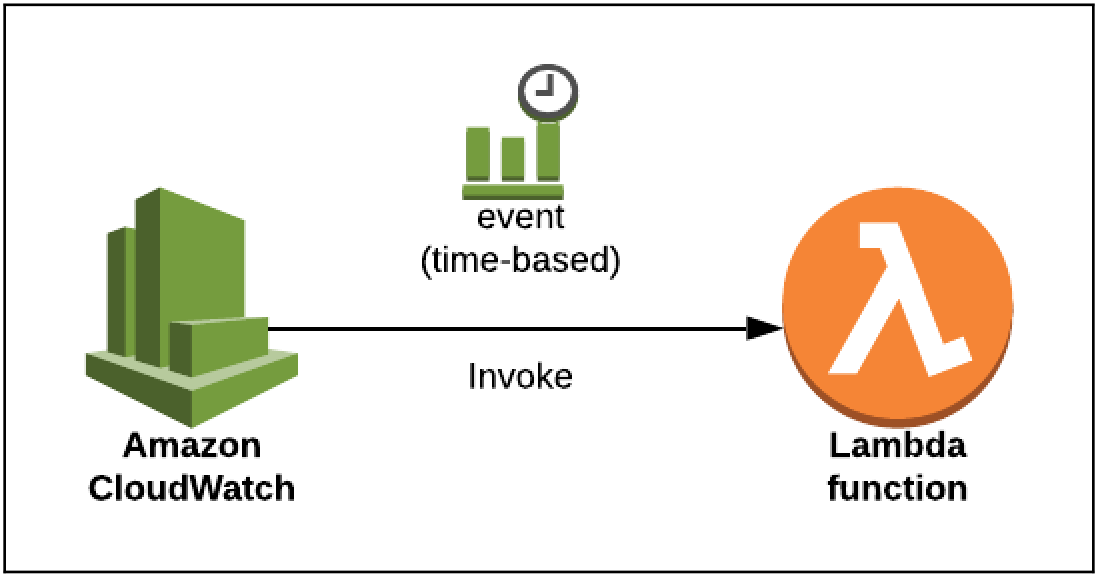

Amazon EC2
Event Producers
Amazon Kinesis
Data Stream
AWS Lambda
Event Consumer
Amazon S3
Event Persistence

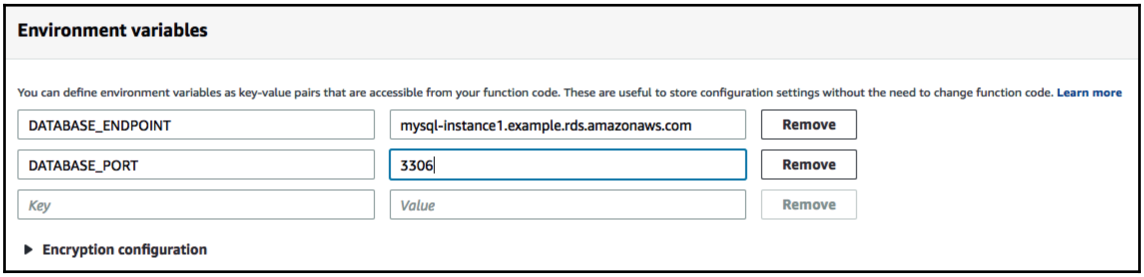
Chapter 3: The Foundations of a Function in AWS





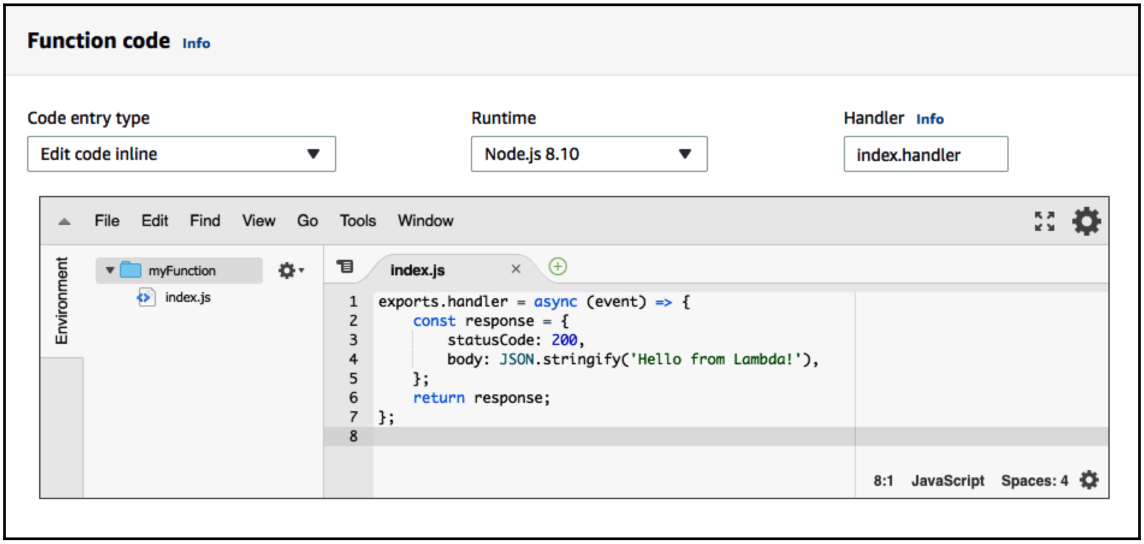





Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. Learn more

DATABASE_ENDPOINT	mysql-instance1.example.rds.amazonaws.com	Remove
DATABASE_PORT	3306	Remove
Key	Value	Remove



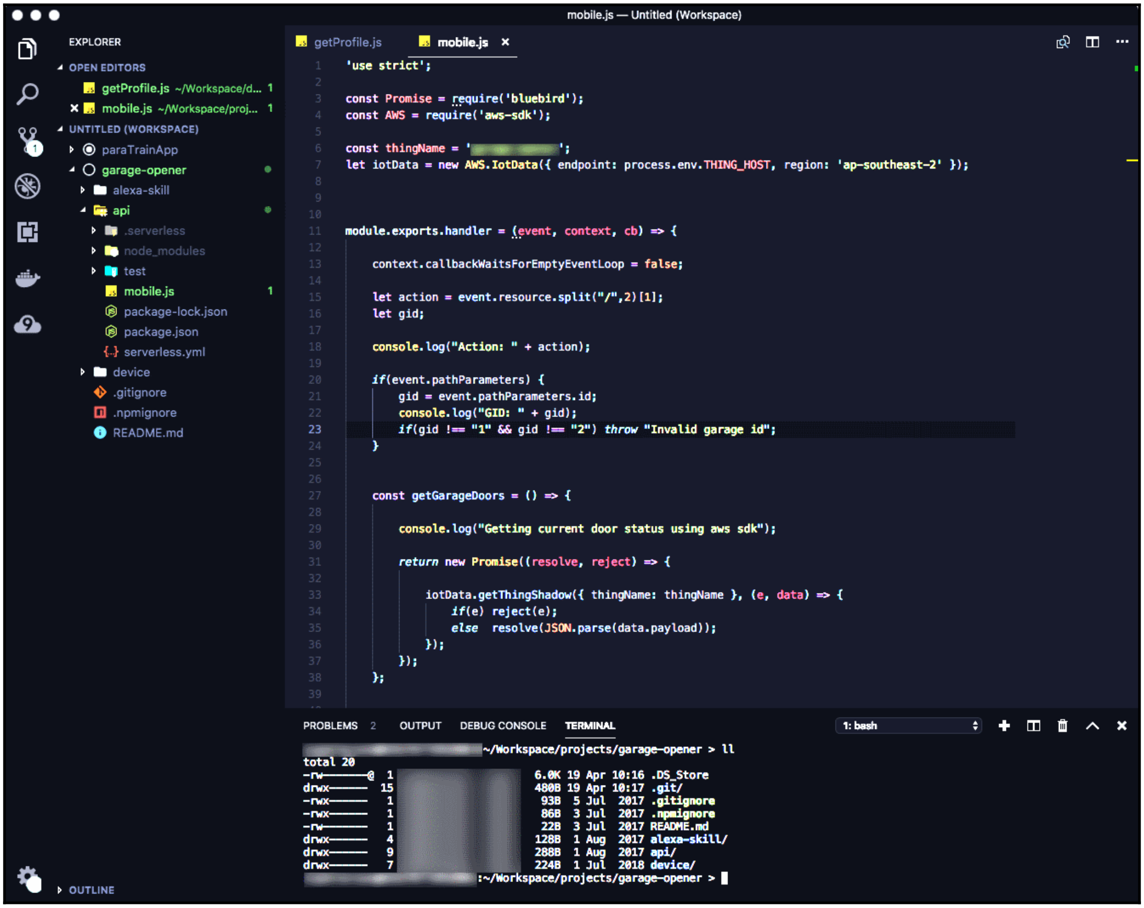
Function code Info

Code entry type: Edit code inline

Runtime: Node.js 8.10

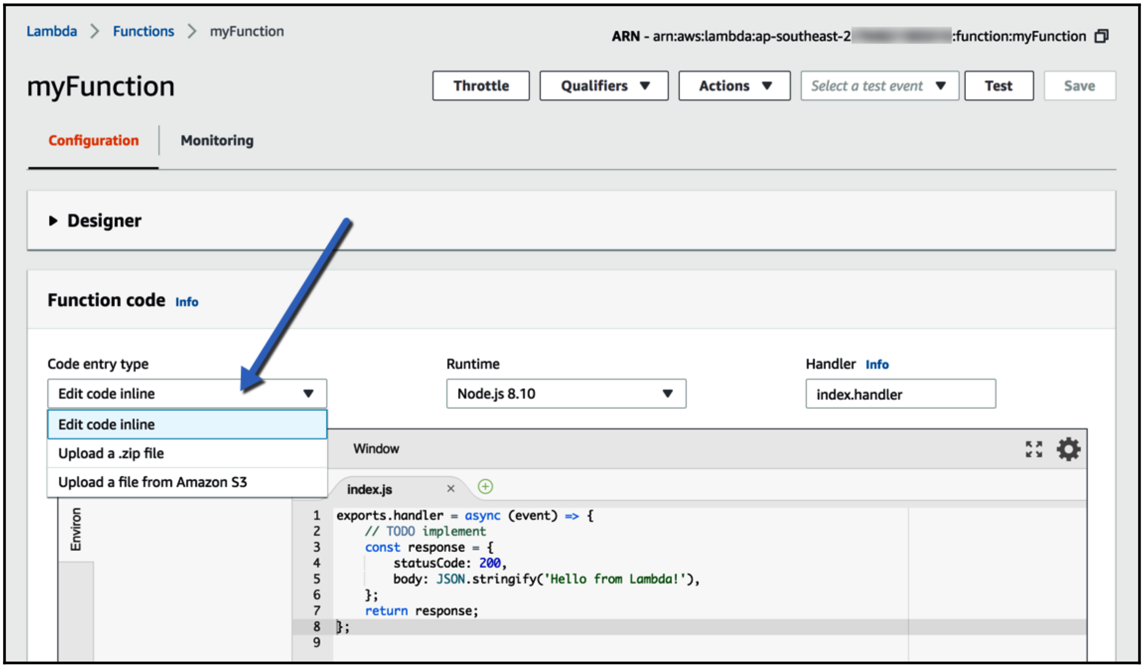
Handler Info: index.handler

```
exports.handler = async (event) => {
    const response = {
        statusCode: 200,
        body: JSON.stringify('Hello from Lambda!'),
    };
    return response;
};
```



```
'use strict';

const Promise = require('bluebird');
const AWS = require('aws-sdk');

const thingName = '[redacted]';
let iotData = new AWS.IotData({ endpoint: process.env.THING_HOST, region: 'ap-southeast-2' });



module.exports.handler = (event, context, cb) => {

    context.callbackWaitsForEmptyEventLoop = false;

    let action = event.resource.split("/",2)[1];
    let gid;

    console.log("Action: " + action);

    if(event.pathParameters) {
        gid = event.pathParameters.id;
        console.log("GID: " + gid);
        if(gid !== "1" && gid !== "2") throw "Invalid garage id";
    }


    const getGarageDoors = () => {

        console.log("Getting current door status using aws sdk");

        return new Promise((resolve, reject) => {

            iotData.getThingShadow({ thingName: thingName }, (e, data) => {
                if(e) reject(e);
                else  resolve(JSON.parse(data.payload));
            });
        });
    };

```

```
[redacted]~/Workspace/projects/garage-opener > ll
total 20
-rw-------@  1 [redacted]  6.0K 19 Apr 10:16 .DS_Store
drwx------  15              480B 19 Apr 10:17 .git/
-rwx------   1               93B  5 Jul  2017 .gitignore
-rwx------   1               86B  3 Jul  2017 .npmignore
-rw-------   1               22B  3 Jul  2017 README.md
drwx------   4              128B  1 Aug  2017 alexa-skill/
drwx------   9              288B  1 Aug  2017 api/
drwx------   7              224B  1 Jul  2018 device/
[redacted]:~/Workspace/projects/garage-opener > █
```



myFunction

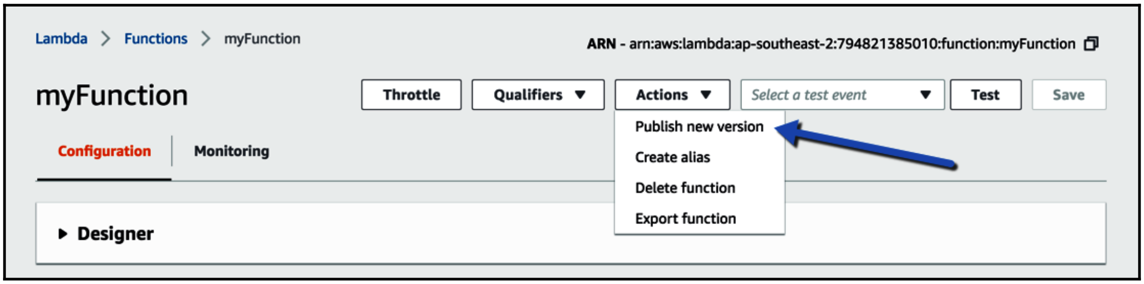
ARN - arn:aws:lambda:ap-southeast-2[redacted]:function:myFunction

Code entry type: Edit code inline

Runtime: Node.js 8.10

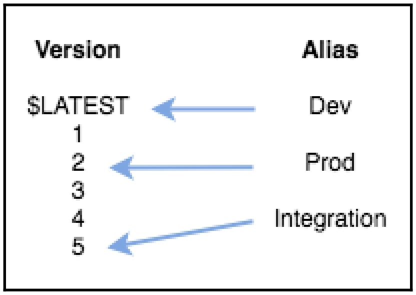
Handler Info: index.handler

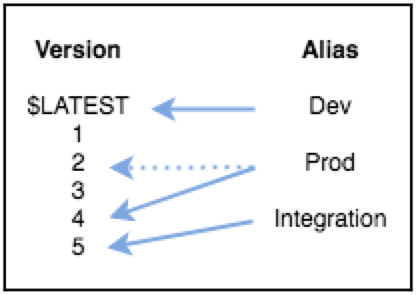
```
exports.handler = async (event) => {
    // TODO implement
    const response = {
        statusCode: 200,
        body: JSON.stringify('Hello from Lambda!'),
    };
    return response;
};
```

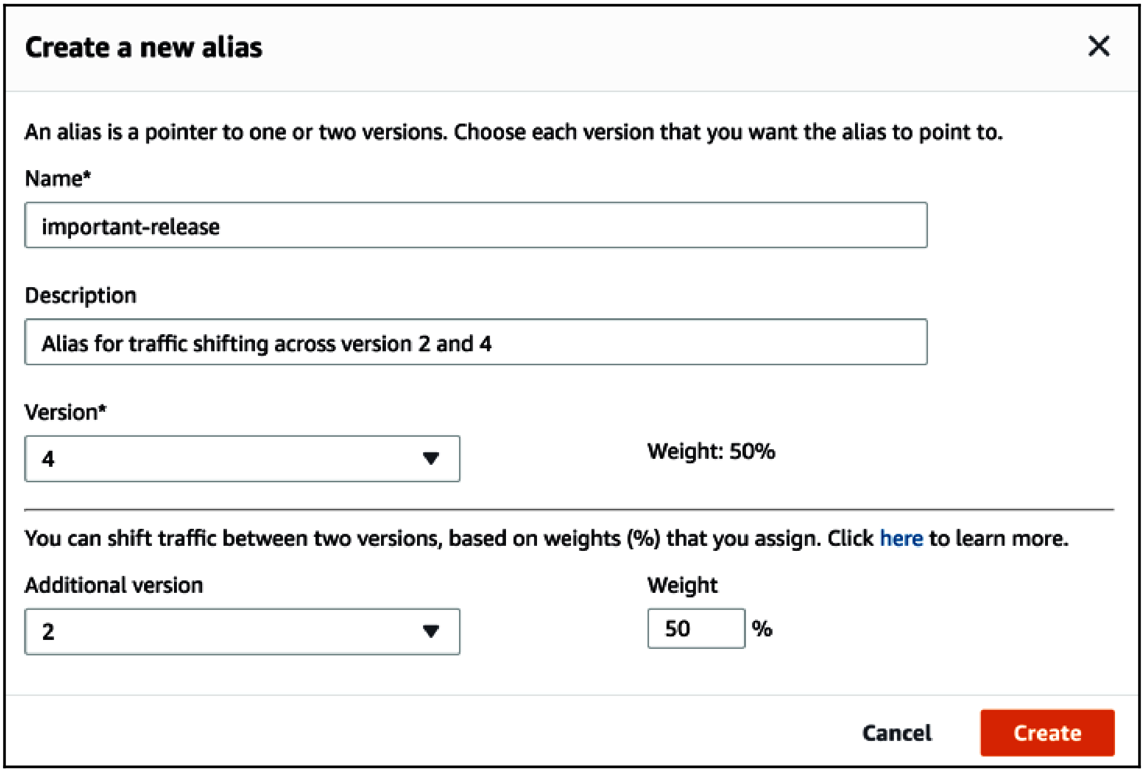


myFunction

ARN - arn:aws:lambda:ap-southeast-2:794821385010:function:myFunction







Create a new alias ✕

An alias is a pointer to one or two versions. Choose each version that you want the alias to point to.

Name*

important-release

Description

Alias for traffic shifting across version 2 and 4

Version*

4 ▼

Weight: 50%

You can shift traffic between two versions, based on weights (%) that you assign. Click here to learn more.

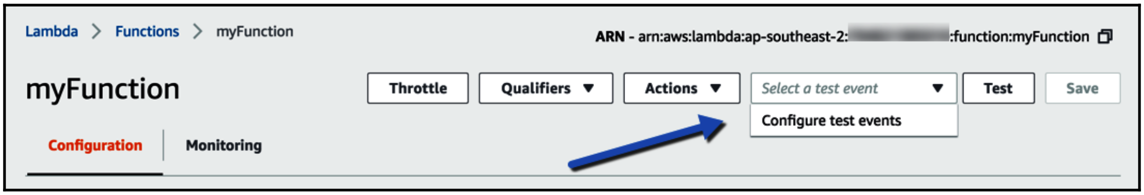
Additional version

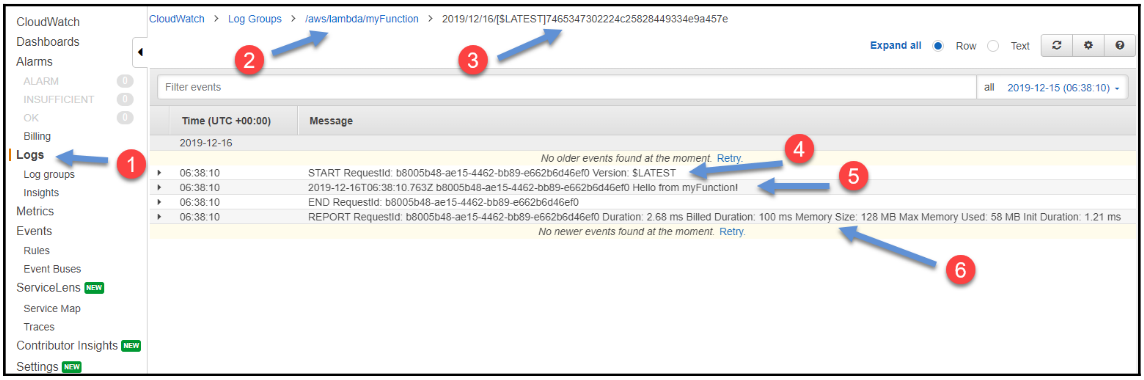
2 ▼

Weight

50 %

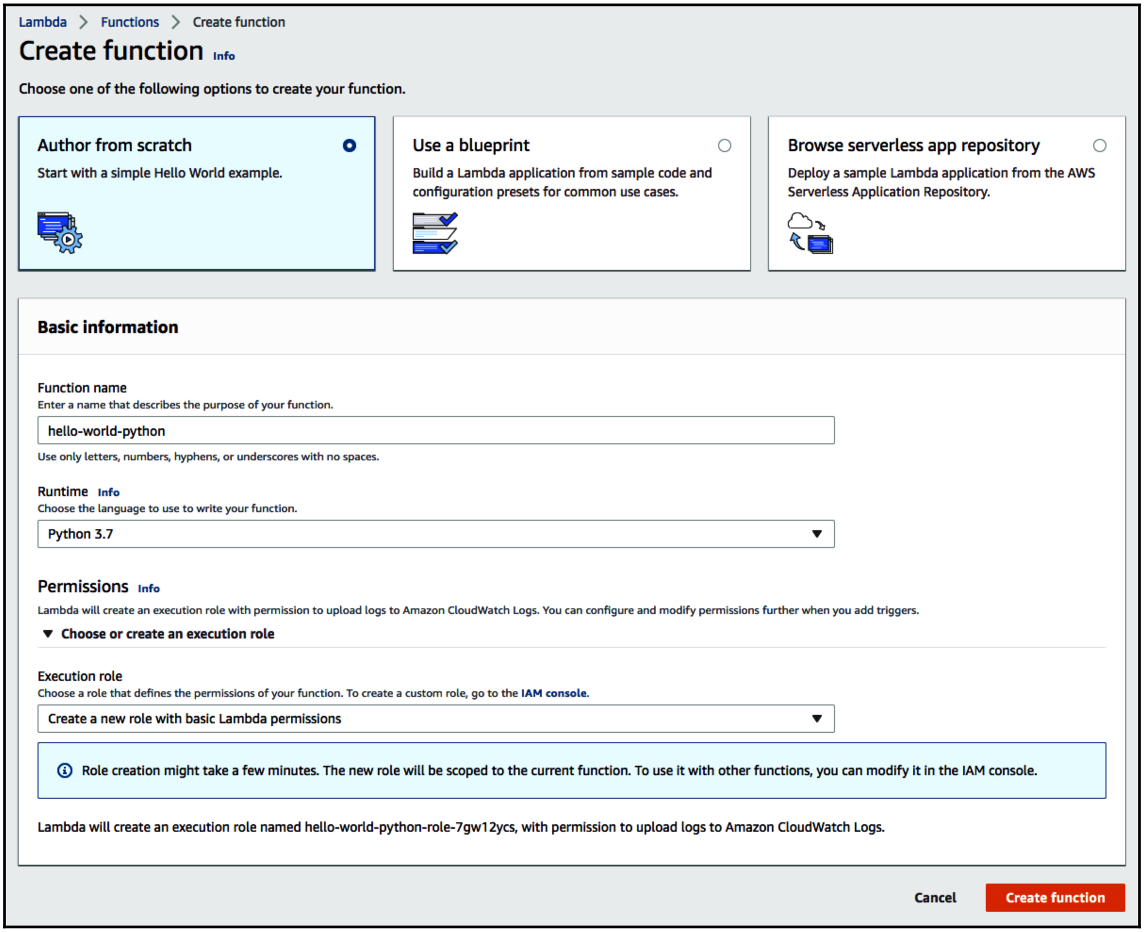
Cancel | Create





CloudWatch > Log Groups > /aws/lambda/myFunction > 2019/12/16/[$LATEST]7465347302224c25828449334e9a457e

```
Time (UTC +00:00)   Message
2019-12-16
                    No older events found at the moment. Retry.
06:38:10            START RequestId: b8005b48-ae15-4462-bb89-e662b6d46ef0 Version: $LATEST
06:38:10            2019-12-16T06:38:10.763Z b8005b48-ae15-4462-bb89-e662b6d46ef0 Hello from myFunction!
06:38:10            END RequestId: b8005b48-ae15-4462-bb89-e662b6d46ef0
06:38:10            REPORT RequestId: b8005b48-ae15-4462-bb89-e662b6d46ef0 Duration: 2.68 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 58 MB Init Duration: 1.21 ms
                    No newer events found at the moment. Retry.
```



Lambda > Functions > Create function

Create function Info

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.

Browse serverless app repository ○
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name
Enter a name that describes the purpose of your function.
hello-world-python
Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info
Choose the language to use to write your function.
Python 3.7 ▼

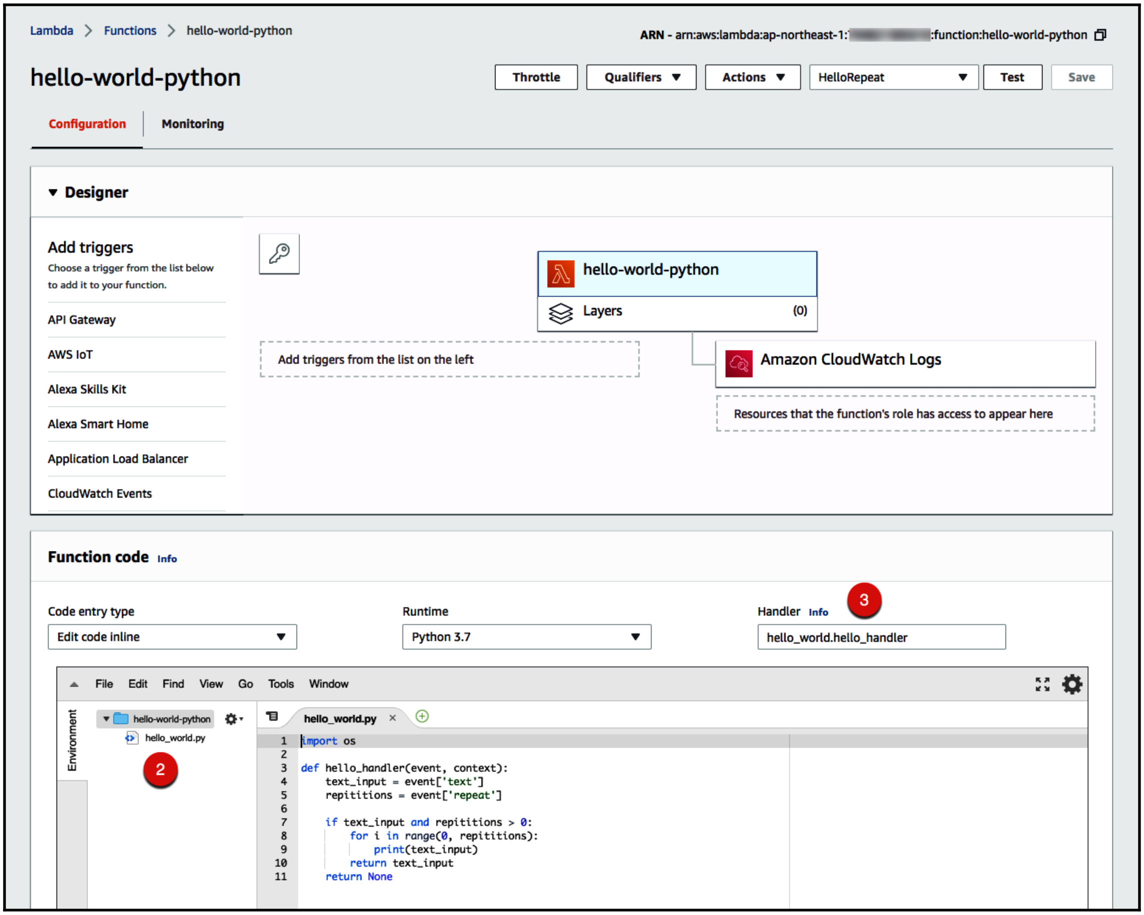
Permissions Info
Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ Choose or create an execution role

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
Create a new role with basic Lambda permissions ▼

ⓘ Role creation might take a few minutes. The new role will be scoped to the current function. To use it with other functions, you can modify it in the IAM console.

Lambda will create an execution role named hello-world-python-role-7gw12ycs, with permission to upload logs to Amazon CloudWatch Logs.



ARN - arn:aws:lambda:ap-northeast-1:[redacted]:function:hello-world-python

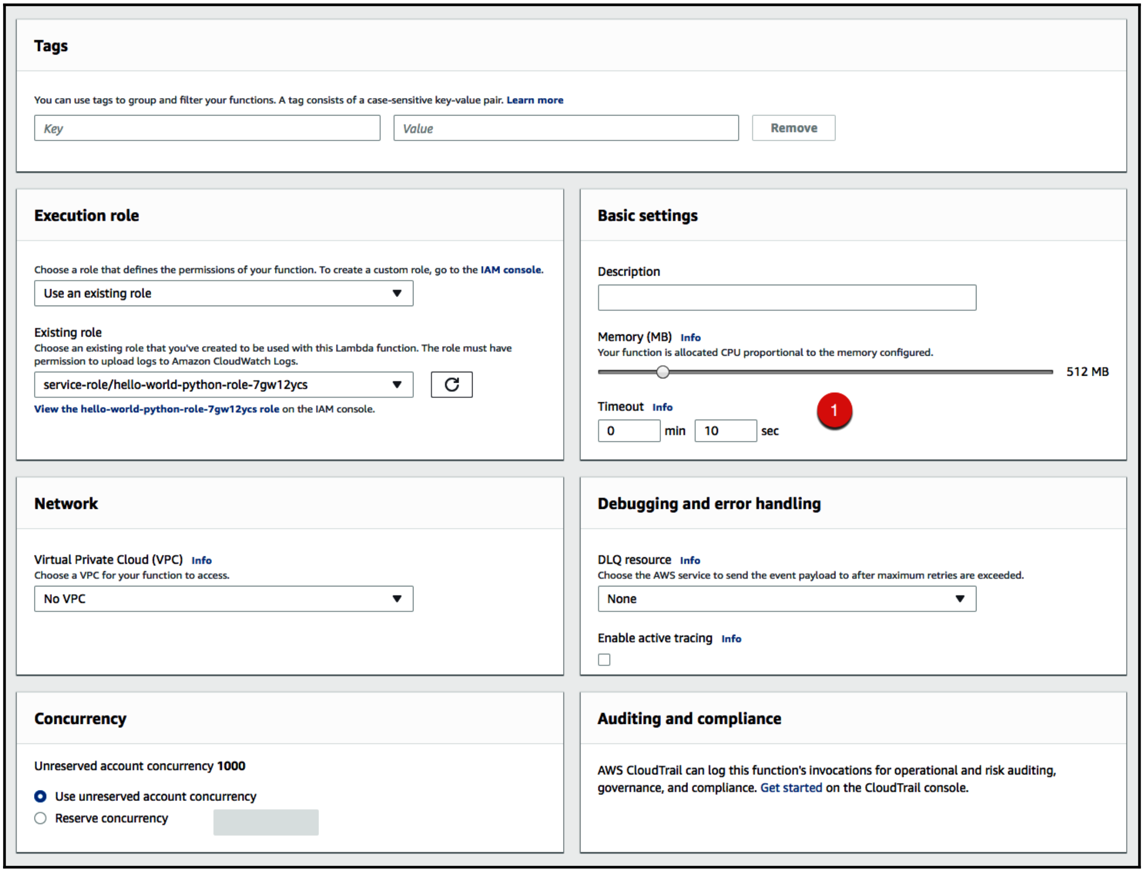
hello-world-python

Code entry type: Edit code inline

Runtime: Python 3.7

Handler: hello_world.hello_handler

```
import os

def hello_handler(event, context):
    text_input = event['text']
    repititions = event['repeat']

    if text_input and repititions > 0:
        for i in range(0, repititions):
            print(text_input)
        return text_input
    return None
```



Tags

You can use tags to group and filter your functions. A tag consists of a case-sensitive key-value pair. Learn more

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

Use an existing role ▼

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.

service-role/hello-world-python-role-7gw12ycs ▼

View the hello-world-python-role-7gw12ycs role on the IAM console.

Basic settings

Description

Memory (MB) Info

Your function is allocated CPU proportional to the memory configured.

512 MB

Timeout Info

0 min 10 sec

Network

Virtual Private Cloud (VPC) Info

Choose a VPC for your function to access.

No VPC ▼

Debugging and error handling

DLQ resource Info

Choose the AWS service to send the event payload to after maximum retries are exceeded.

None ▼

Enable active tracing Info

□

Concurrency

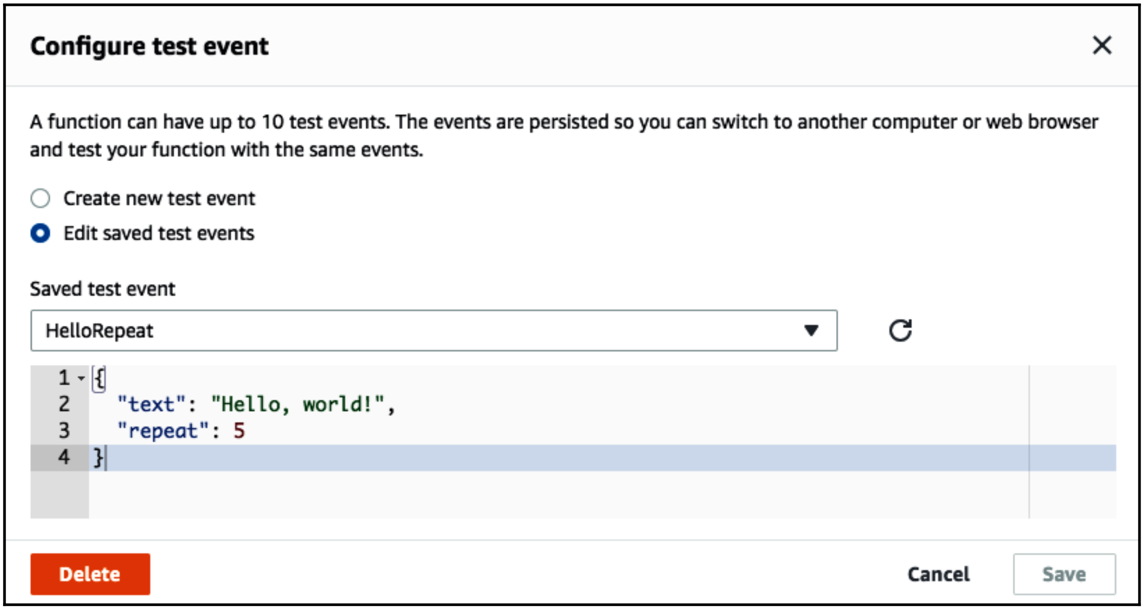
Unreserved account concurrency **1000**

● Use unreserved account concurrency

○ Reserve concurrency

Auditing and compliance

AWS CloudTrail can log this function's invocations for operational and risk auditing, governance, and compliance. Get started on the CloudTrail console.



Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

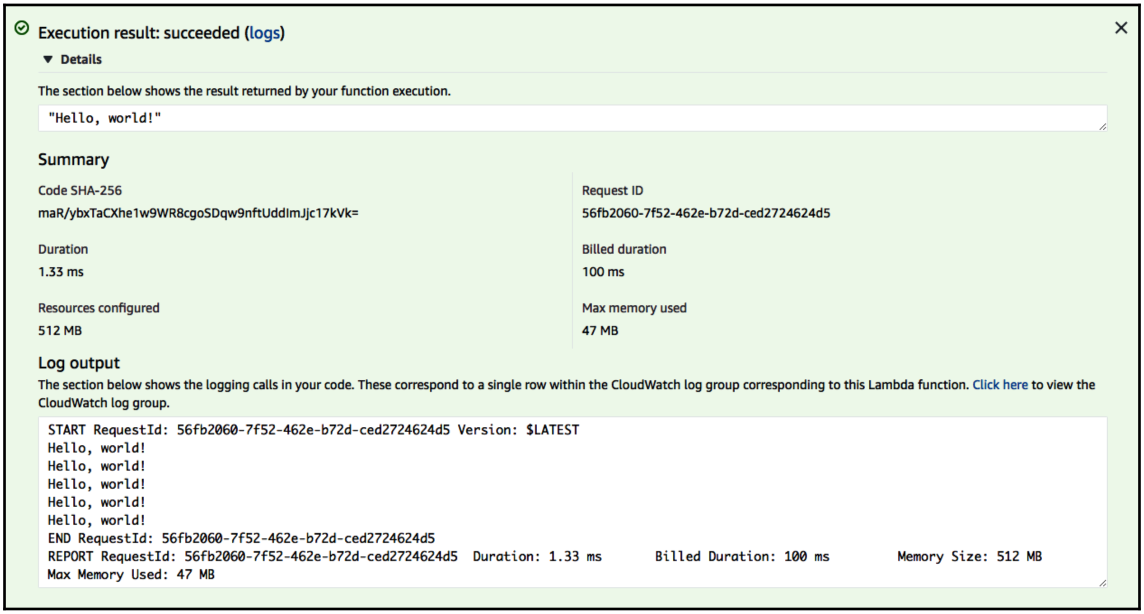
○ Create new test event

● Edit saved test events

Saved test event

HelloRepeat

```
{
  "text": "Hello, world!",
  "repeat": 5
}
```



Execution result: succeeded (logs)

▼ Details

The section below shows the result returned by your function execution.

```
"Hello, world!"
```

Summary

Code SHA-256

maR/ybxTaCXhe1w9WR8cgoSDqw9nftUddImJjc17kVk=

Request ID

56fb2060-7f52-462e-b72d-ced2724624d5

Duration

1.33 ms

Billed duration

100 ms

Resources configured

512 MB

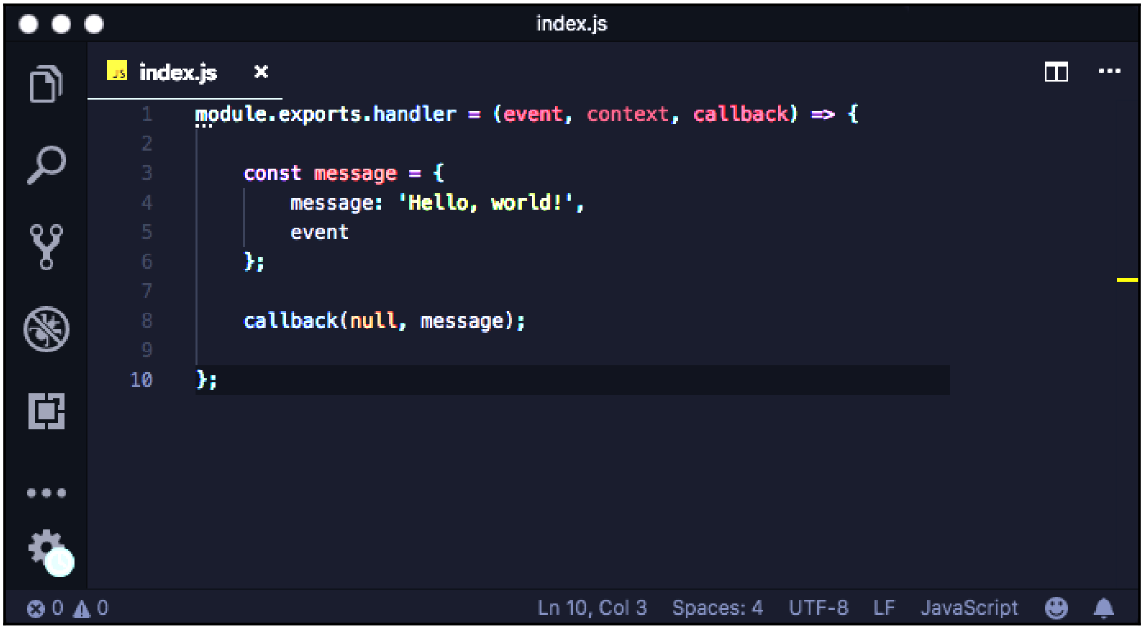
Max memory used

47 MB

Log output

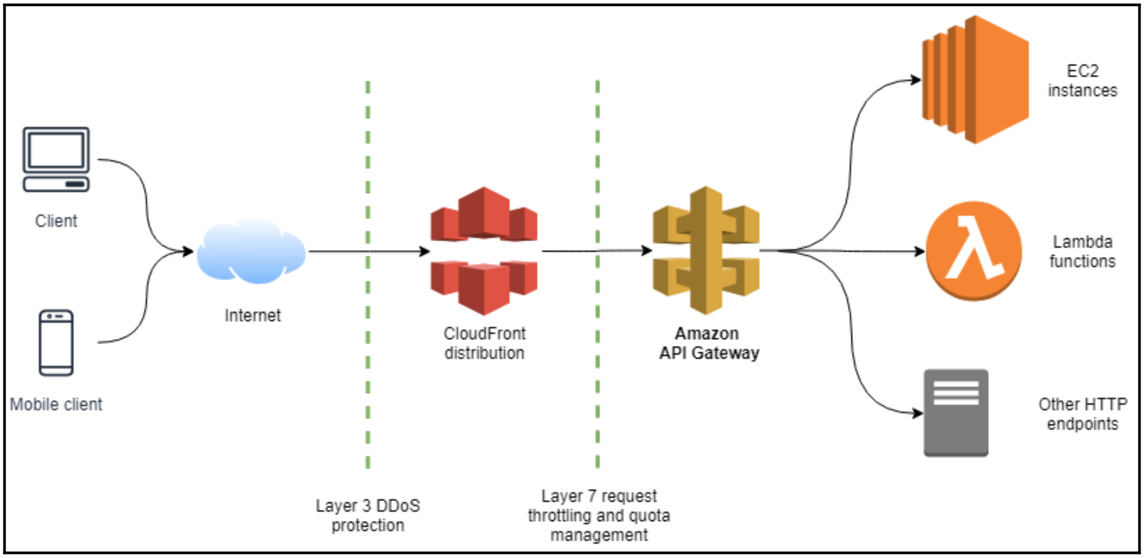
The section below shows the logging calls in your code. These correspond to a single row within the CloudWatch log group corresponding to this Lambda function. Click here to view the CloudWatch log group.

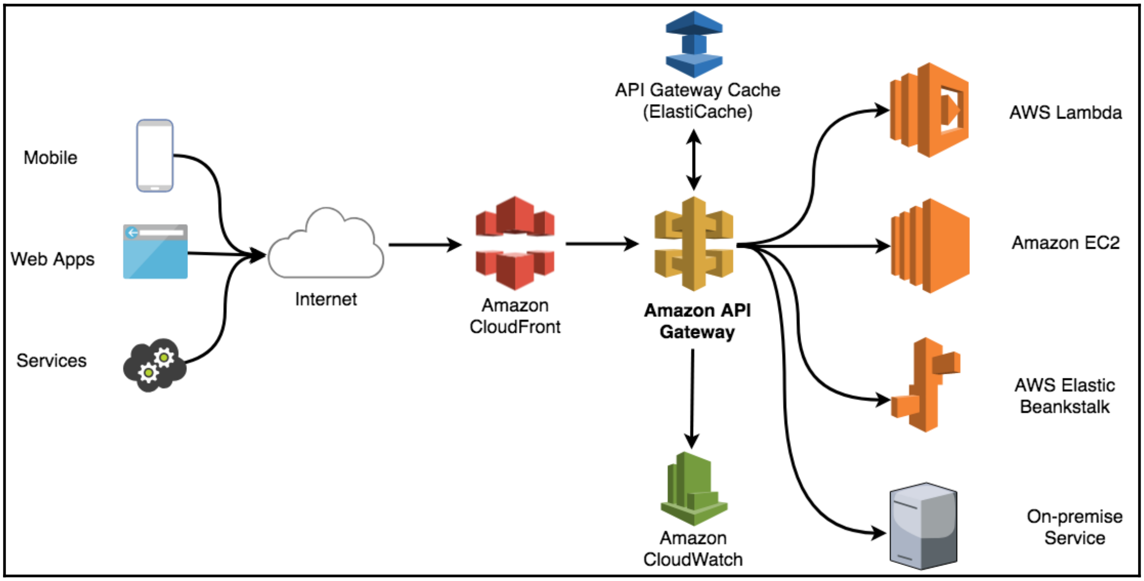
```
START RequestId: 56fb2060-7f52-462e-b72d-ced2724624d5 Version: $LATEST
Hello, world!
Hello, world!
Hello, world!
Hello, world!
Hello, world!
END RequestId: 56fb2060-7f52-462e-b72d-ced2724624d5
REPORT RequestId: 56fb2060-7f52-462e-b72d-ced2724624d5  Duration: 1.33 ms       Billed Duration: 100 ms         Memory Size: 512 MB
Max Memory Used: 47 MB
```

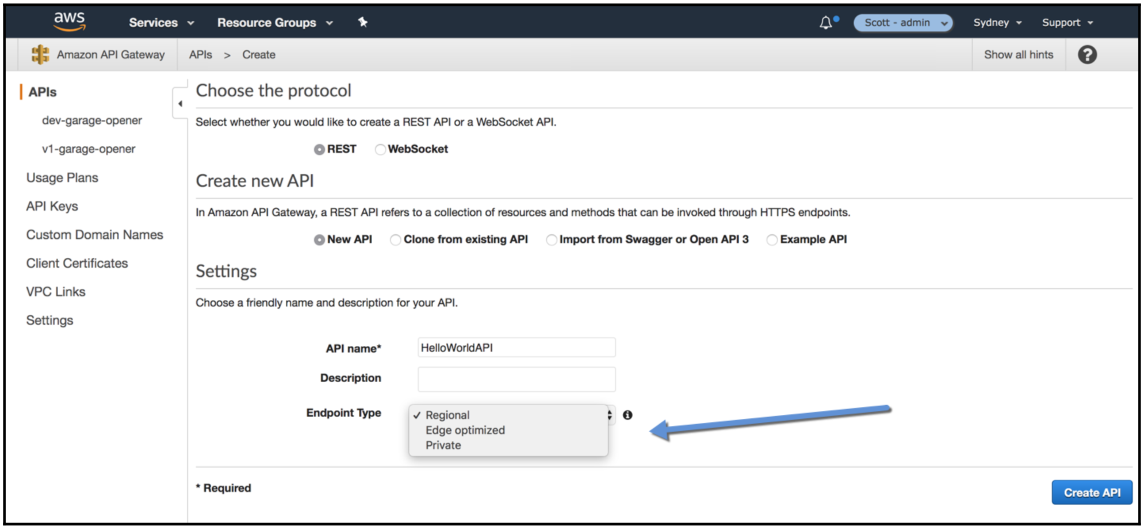


```
module.exports.handler = (event, context, callback) => {

    const message = {
        message: 'Hello, world!',
        event
    };

    callback(null, message);

};
```

Chapter 4: Adding Amazon API Gateway







Choose the protocol

Select whether you would like to create a REST API or a WebSocket API.

● REST ○ WebSocket

Create new API

In Amazon API Gateway, a REST API refers to a collection of resources and methods that can be invoked through HTTPS endpoints.

● New API ○ Clone from existing API ○ Import from Swagger or Open API 3 ○ Example API

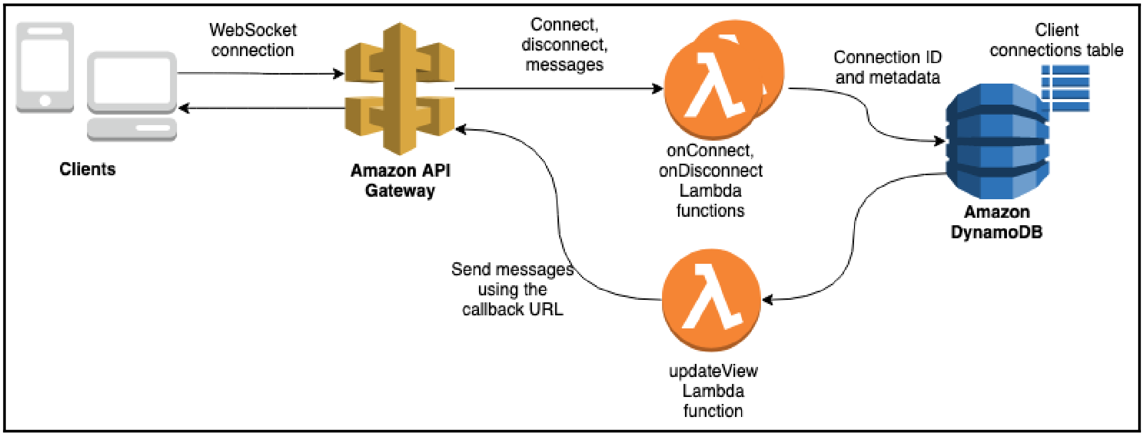
Settings

Choose a friendly name and description for your API.

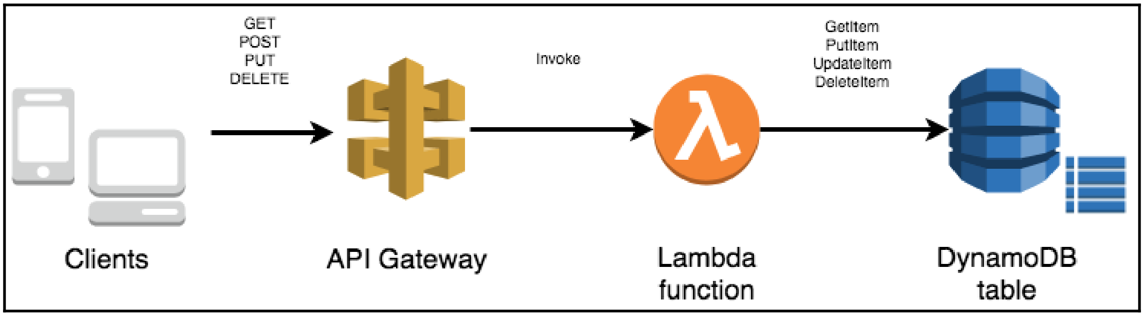
API name*	HelloWorldAPI
Description	
Endpoint Type	✓ Regional Edge optimized Private

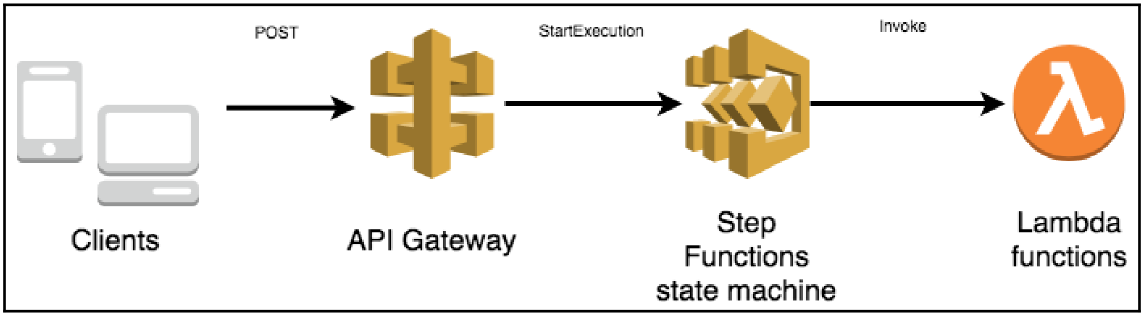
* Required

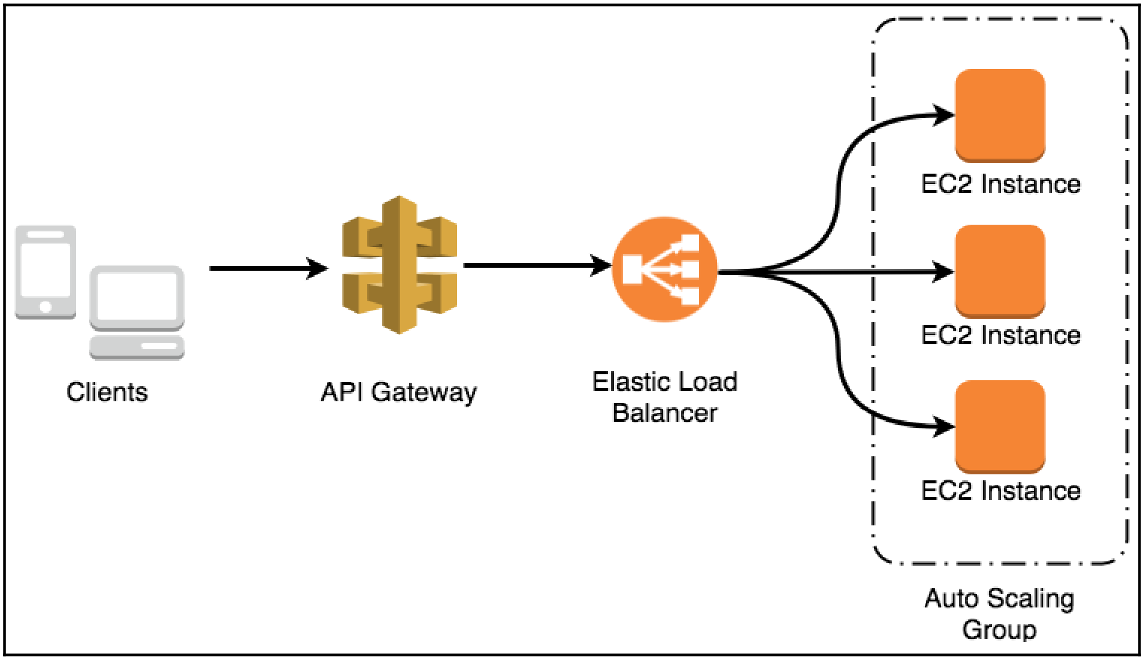
Create API

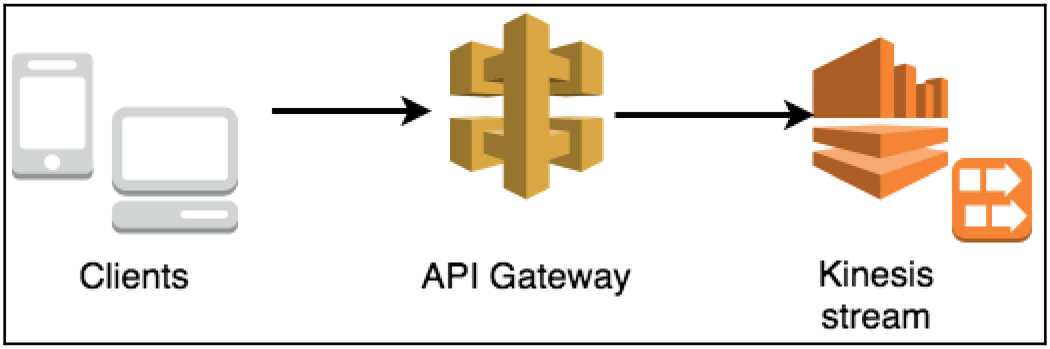


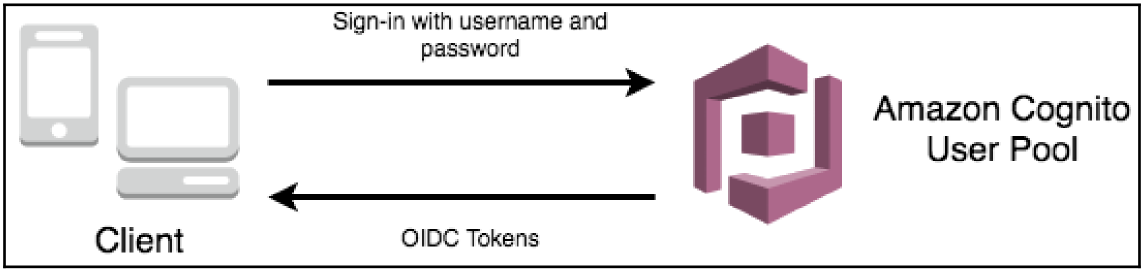
WebSocket connection
Clients
Amazon API Gateway
Connect, disconnect, messages
onConnect, onDisconnect Lambda functions
Connection ID and metadata
Client connections table
Amazon DynamoDB
Send messages using the callback URL
updateView Lambda function

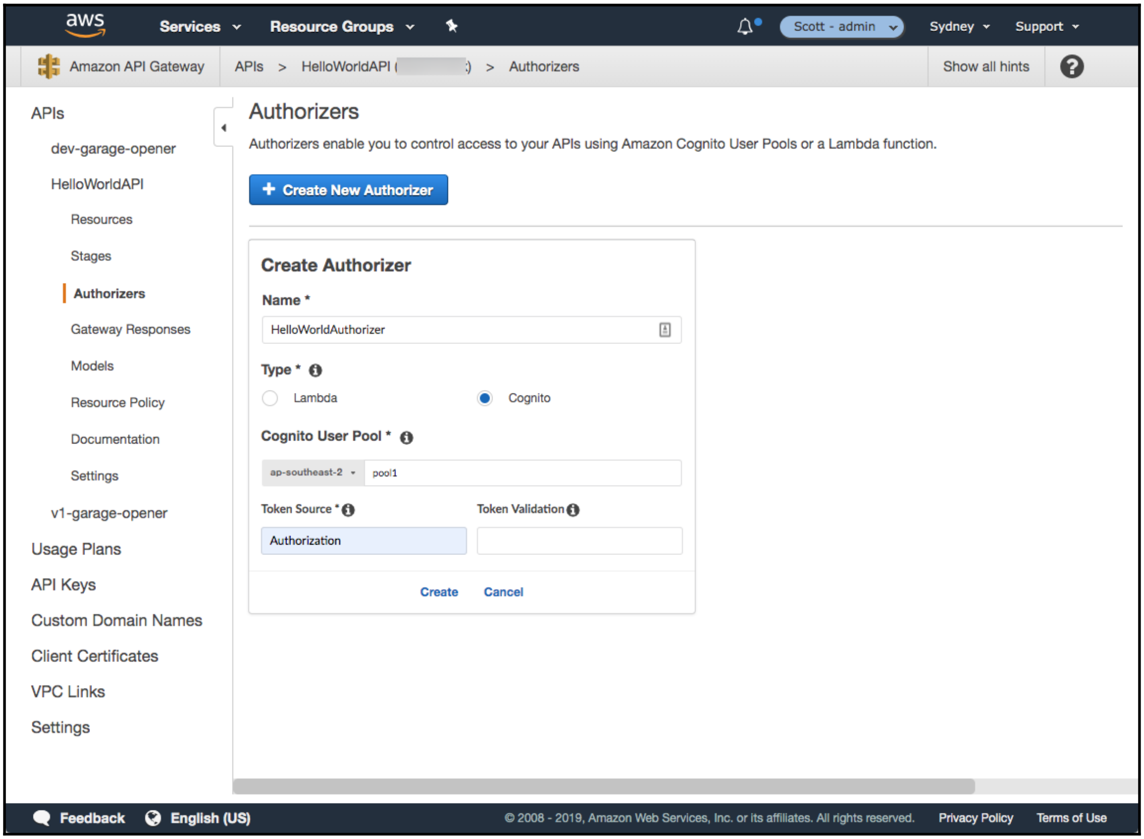



POST
StartExecution
Invoke
Clients
API Gateway
Step Functions state machine
Lambda functions


Clients
API Gateway
Elastic Load Balancer
EC2 Instance
EC2 Instance
EC2 Instance
Auto Scaling Group


Clients
API Gateway
Kinesis stream





Authorizers

Authorizers enable you to control access to your APIs using Amazon Cognito User Pools or a Lambda function.

Create Authorizer

Name *: HelloWorldAuthorizer

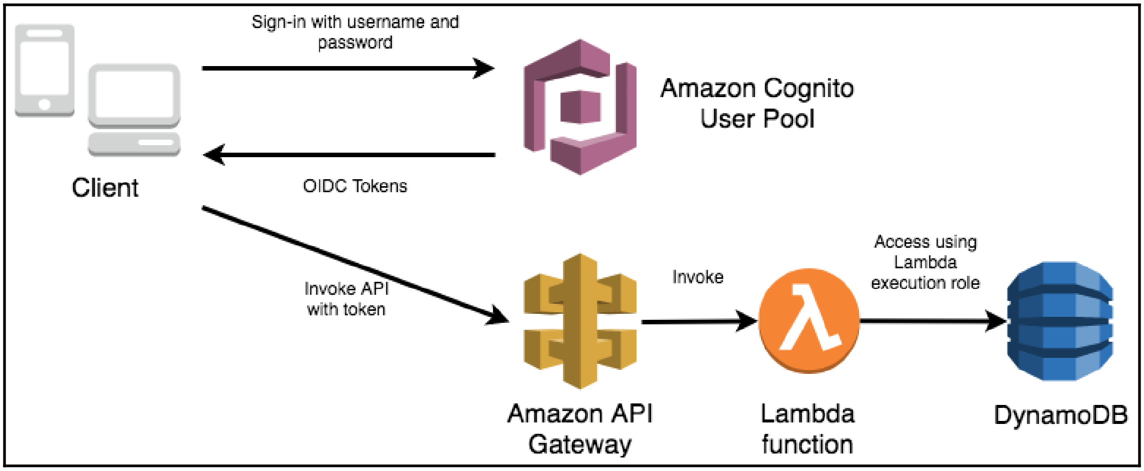
Type *: ○ Lambda ● Cognito

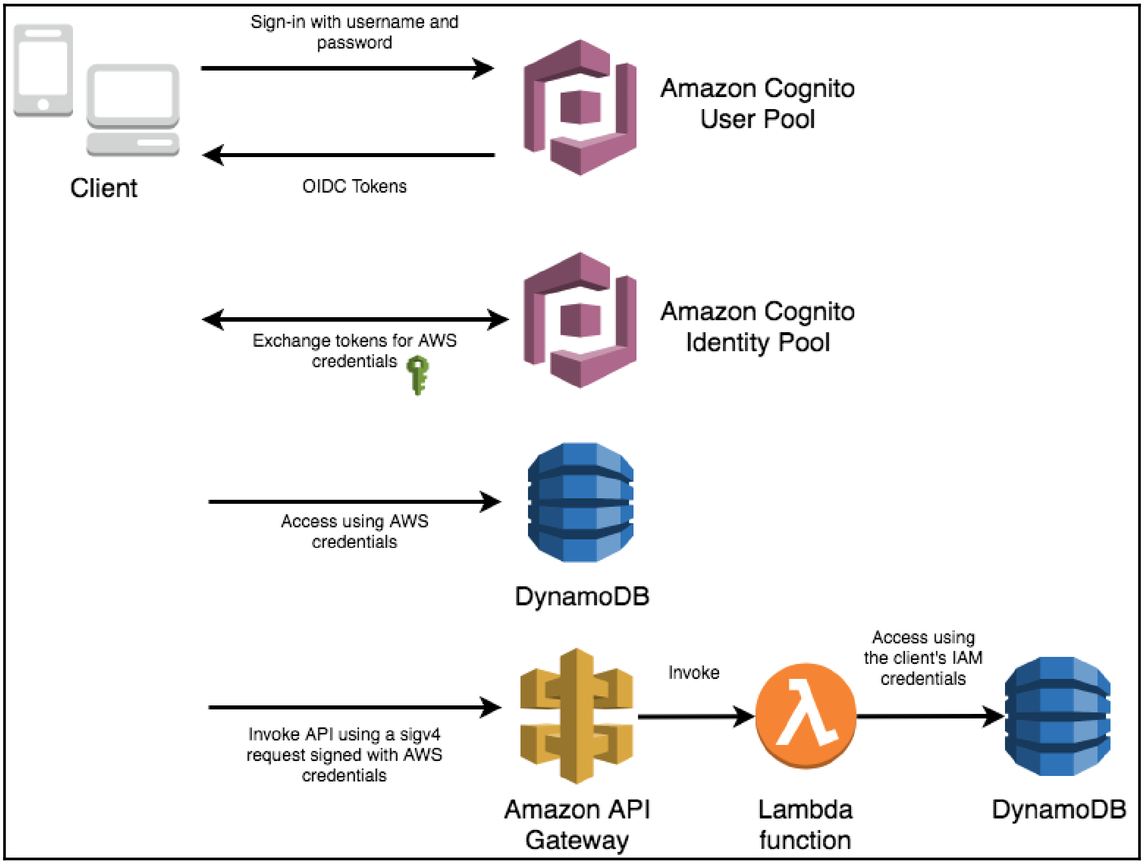
Cognito User Pool *: ap-southeast-2 pool1

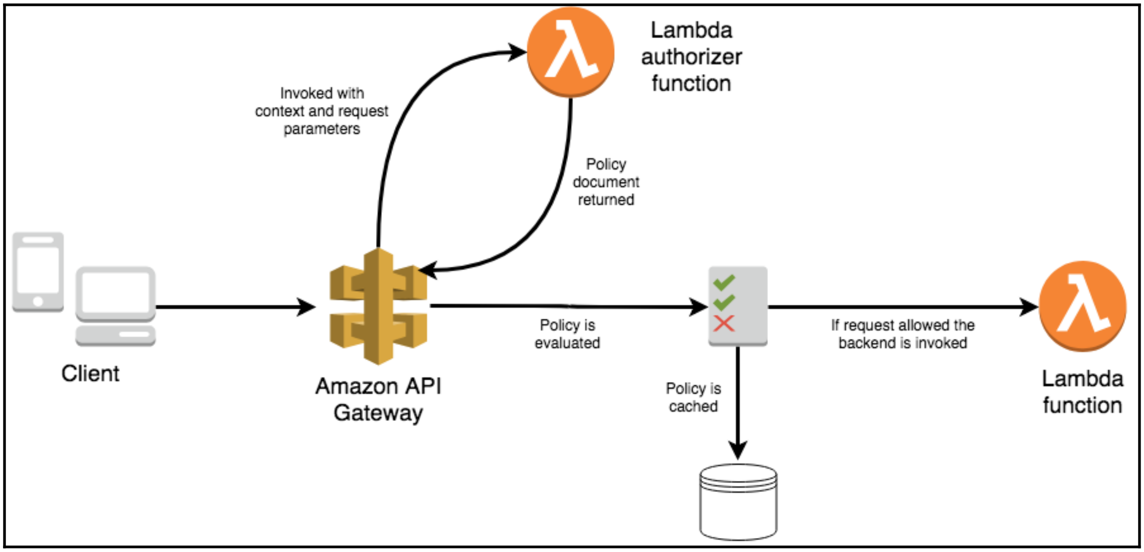
Token Source *: Authorization

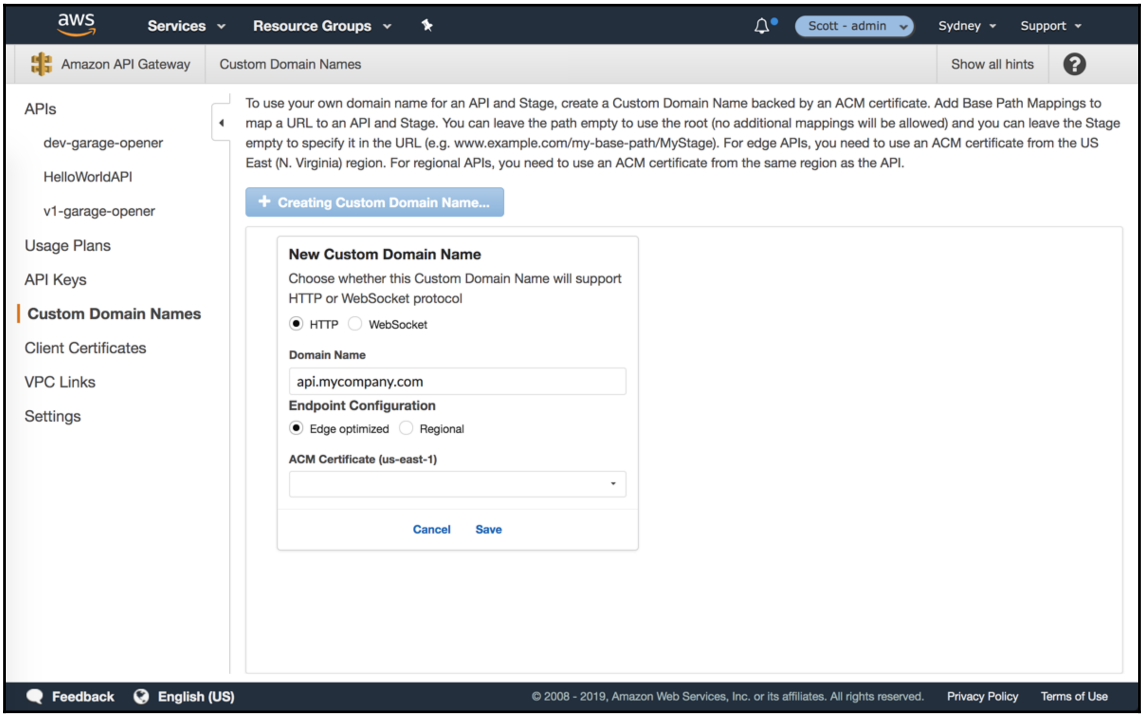
Token Validation:

© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.


Sign-in with username and password
Amazon Cognito User Pool
OIDC Tokens
Client
Invoke API with token
Invoke
Access using Lambda execution role
Amazon API Gateway
Lambda function
DynamoDB


Sign-in with username and password
Amazon Cognito User Pool
OIDC Tokens
Client
Exchange tokens for AWS credentials
Amazon Cognito Identity Pool
Access using AWS credentials
DynamoDB
Invoke API using a sigv4 request signed with AWS credentials
Invoke
Access using the client's IAM credentials
Amazon API Gateway
Lambda function
DynamoDB


Lambda authorizer function
Invoked with context and request parameters
Policy document returned
Client
Amazon API Gateway
Policy is evaluated
If request allowed the backend is invoked
Lambda function
Policy is cached



To use your own domain name for an API and Stage, create a Custom Domain Name backed by an ACM certificate. Add Base Path Mappings to map a URL to an API and Stage. You can leave the path empty to use the root (no additional mappings will be allowed) and you can leave the Stage empty to specify it in the URL (e.g. www.example.com/my-base-path/MyStage). For edge APIs, you need to use an ACM certificate from the US East (N. Virginia) region. For regional APIs, you need to use an ACM certificate from the same region as the API.

+ Creating Custom Domain Name...

New Custom Domain Name

Choose whether this Custom Domain Name will support HTTP or WebSocket protocol

● HTTP ○ WebSocket

Domain Name

api.mycompany.com

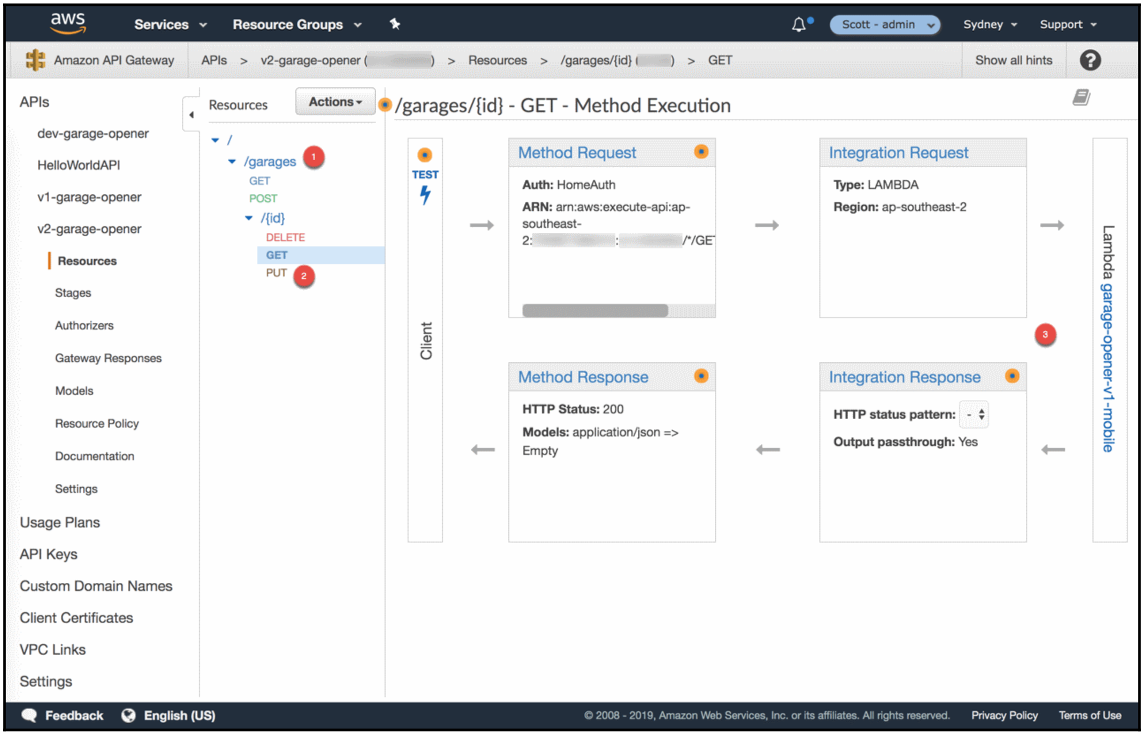
Endpoint Configuration

● Edge optimized ○ Regional

ACM Certificate (us-east-1)

Cancel Save

Feedback
English (US)
© 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use



/garages/{id} - GET - Method Execution

Method Request

Auth: HomeAuth

ARN: arn:aws:execute-api:ap-southeast-2:[redacted]:[redacted]/*/GE

Integration Request

Type: LAMBDA

Region: ap-southeast-2

Method Response

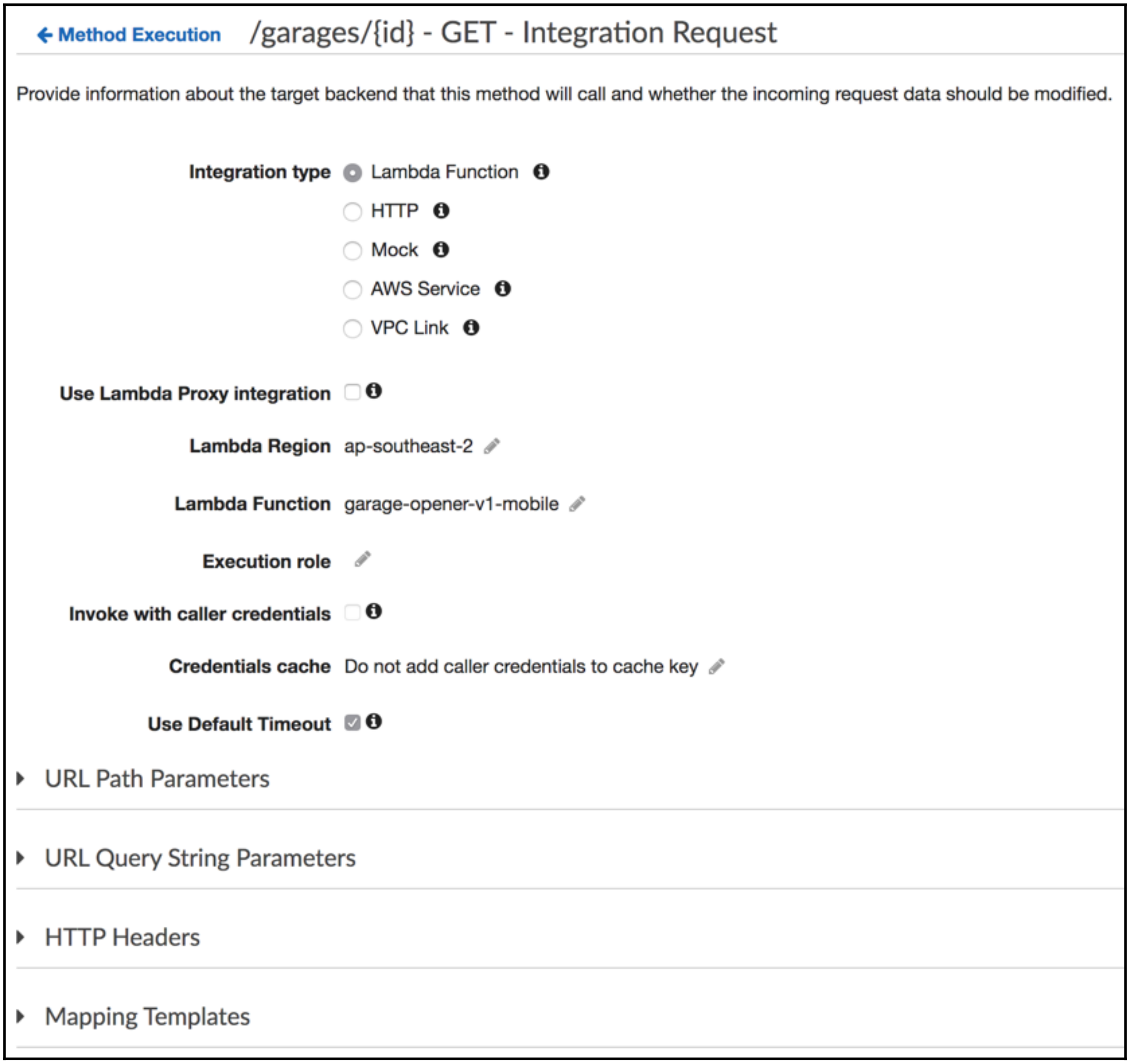
HTTP Status: 200

Models: application/json => Empty

Integration Response

HTTP status pattern: -

Output passthrough: Yes



← Method Execution /garages/{id} - GET - Integration Request

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type ● Lambda Function
○ HTTP
○ Mock
○ AWS Service
○ VPC Link

Use Lambda Proxy integration □

Lambda Region ap-southeast-2

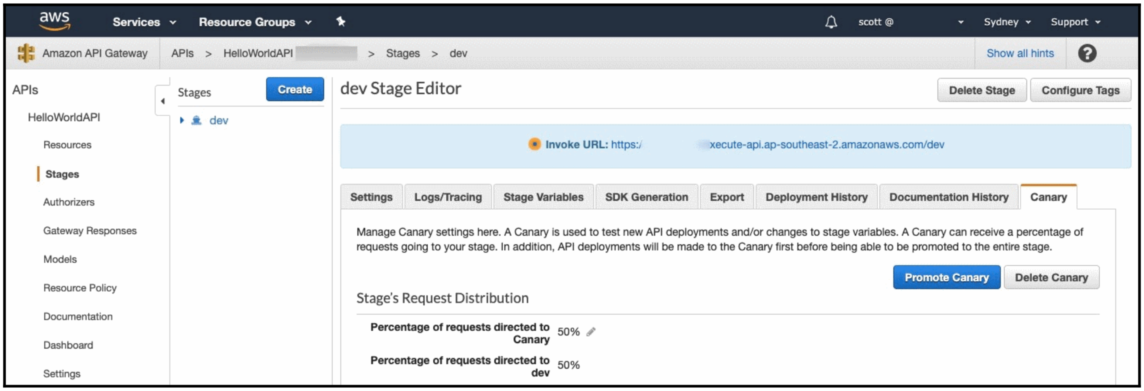
Lambda Function garage-opener-v1-mobile

Execution role

Invoke with caller credentials □

Credentials cache Do not add caller credentials to cache key

Use Default Timeout ☑



dev Stage Editor

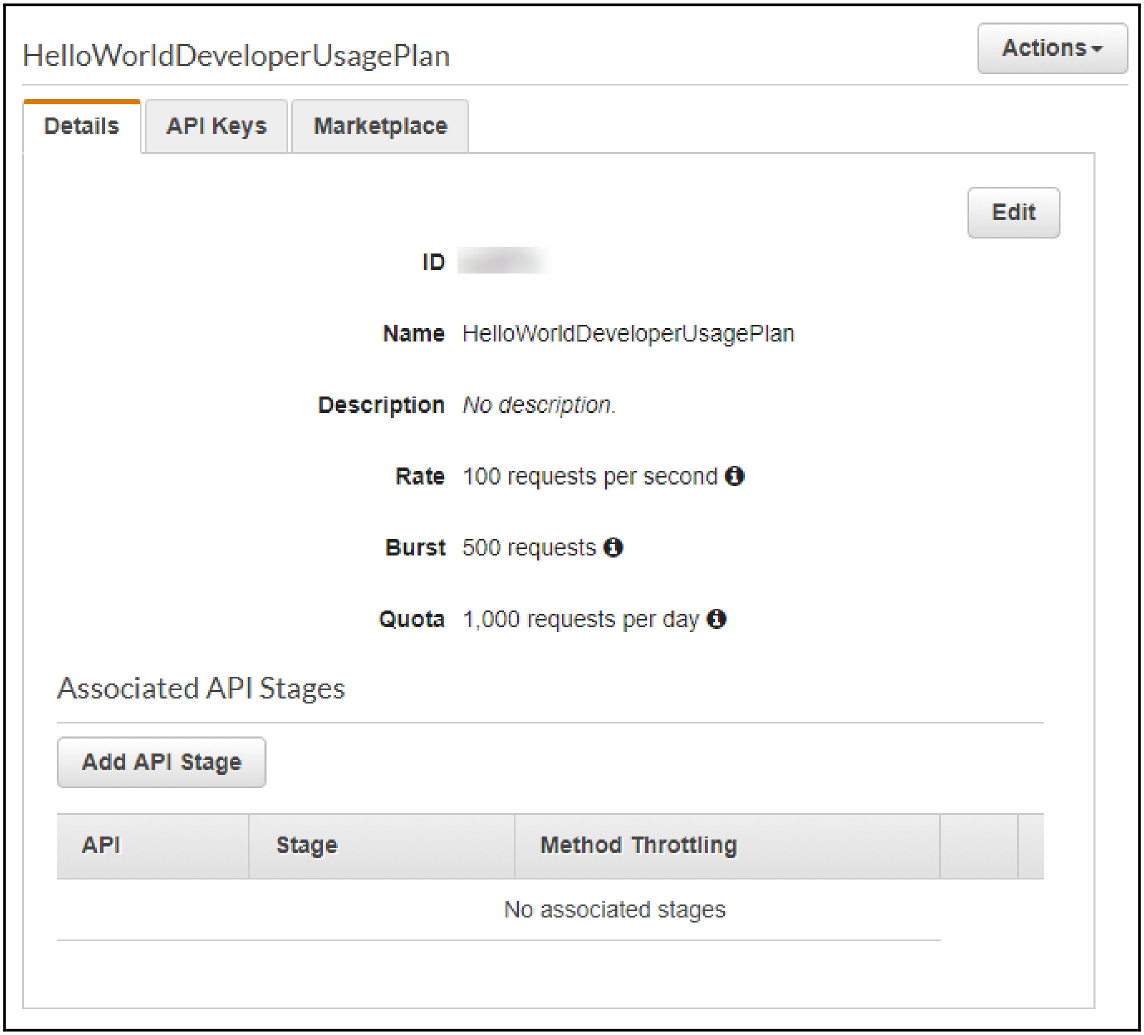
Invoke URL: https:/ xecute-api.ap-southeast-2.amazonaws.com/dev

Manage Canary settings here. A Canary is used to test new API deployments and/or changes to stage variables. A Canary can receive a percentage of requests going to your stage. In addition, API deployments will be made to the Canary first before being able to be promoted to the entire stage.

Stage's Request Distribution

Percentage of requests directed to Canary 50%

Percentage of requests directed to dev 50%



HelloWorldDeveloperUsagePlan

ID [redacted]

Name HelloWorldDeveloperUsagePlan

Description *No description.*

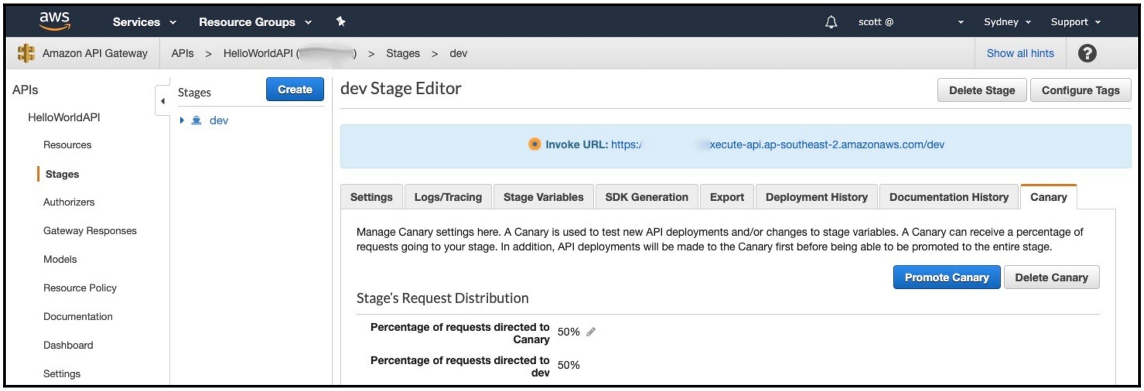
Rate 100 requests per second

Burst 500 requests

Quota 1,000 requests per day

Associated API Stages

API	Stage	Method Throttling	
	No associated stages		



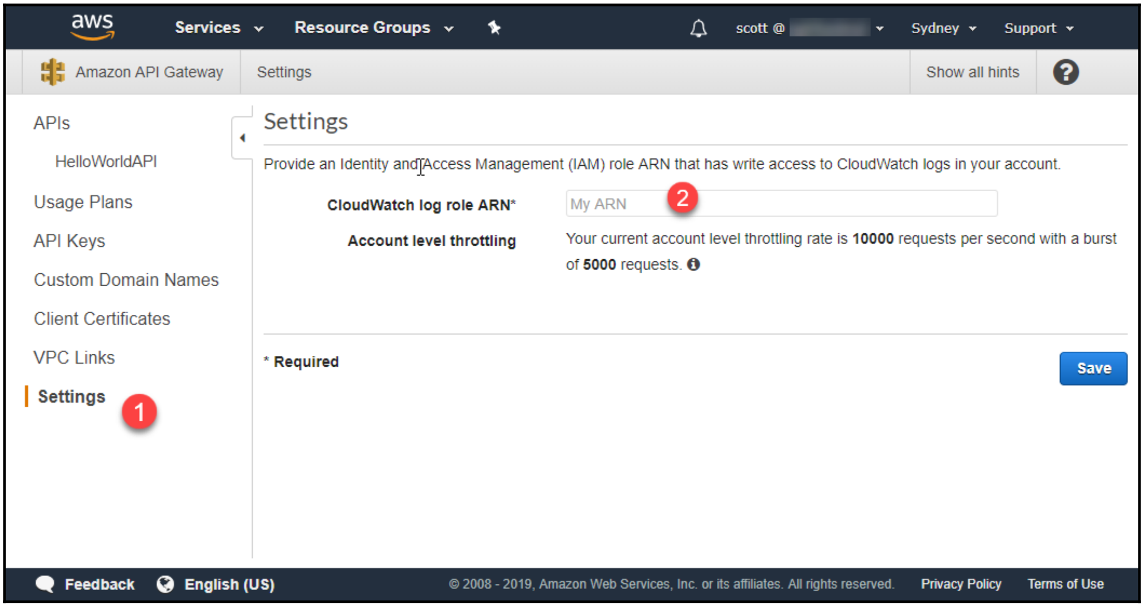
dev Stage Editor

Invoke URL: https:/ [redacted] xecute-api.ap-southeast-2.amazonaws.com/dev

Manage Canary settings here. A Canary is used to test new API deployments and/or changes to stage variables. A Canary can receive a percentage of requests going to your stage. In addition, API deployments will be made to the Canary first before being able to be promoted to the entire stage.

Stage's Request Distribution

Percentage of requests directed to Canary	50%
Percentage of requests directed to dev	50%



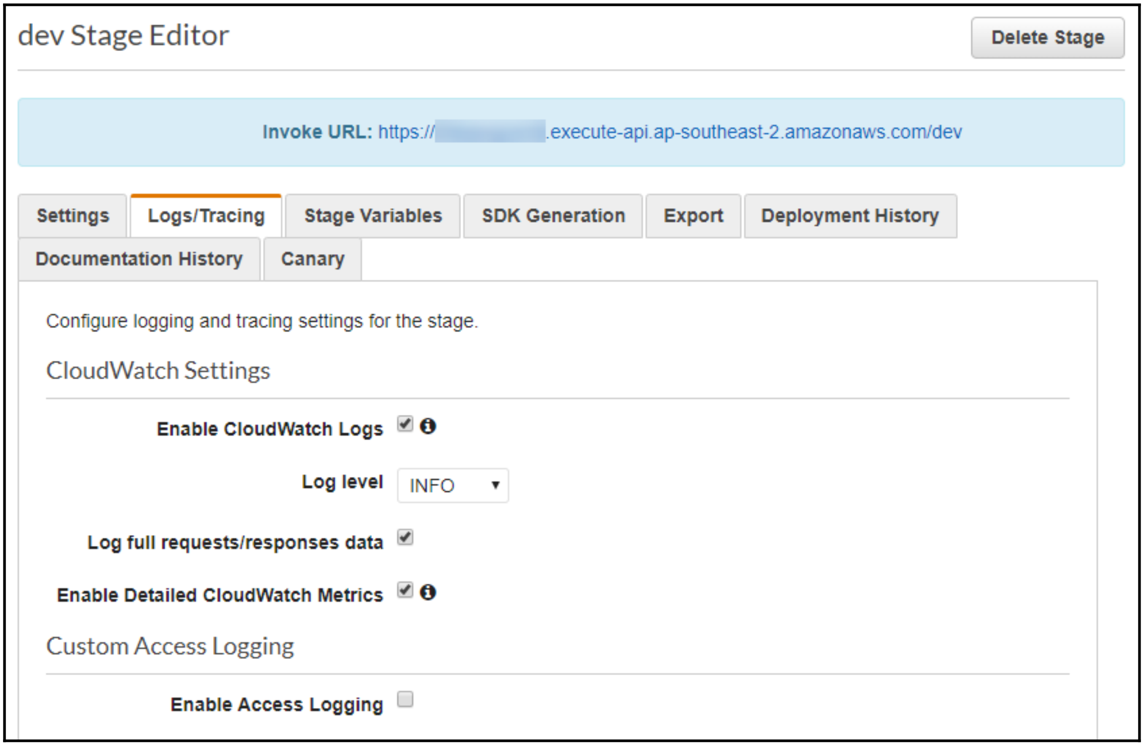
Settings

Provide an Identity and Access Management (IAM) role ARN that has write access to CloudWatch logs in your account.

CloudWatch log role ARN* My ARN

Account level throttling Your current account level throttling rate is **10000** requests per second with a burst of **5000** requests.

* Required



dev Stage Editor

Invoke URL: https://[redacted].execute-api.ap-southeast-2.amazonaws.com/dev

Configure logging and tracing settings for the stage.

CloudWatch Settings

Enable CloudWatch Logs ☑

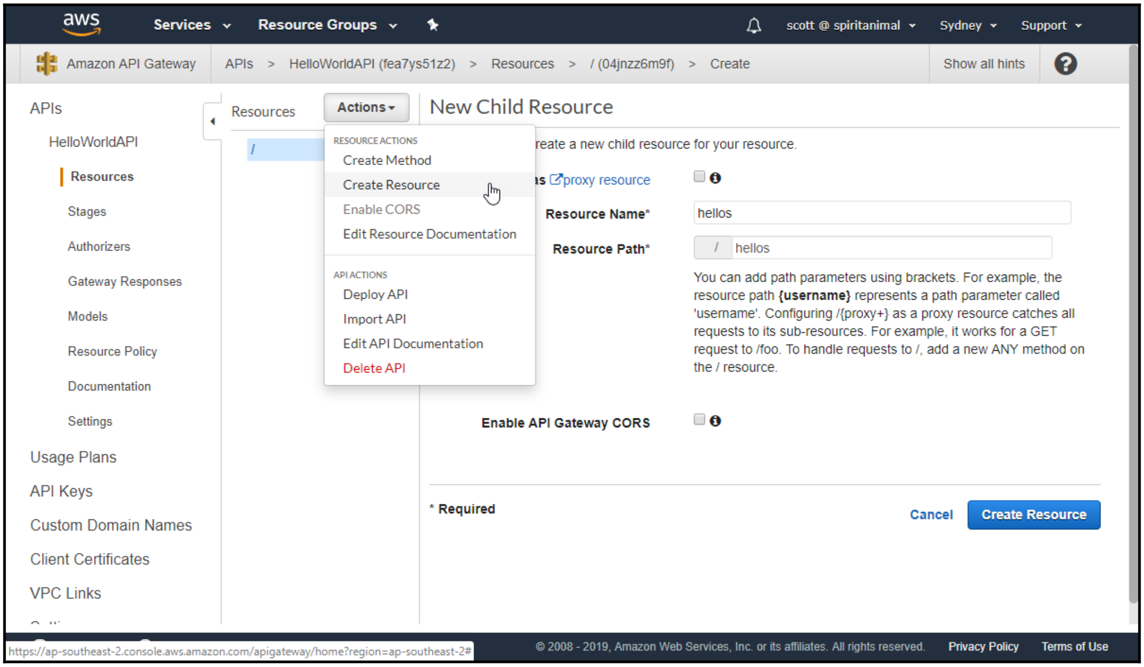
Log level INFO ▾

Log full requests/responses data ☑

Enable Detailed CloudWatch Metrics ☑

Custom Access Logging

Enable Access Logging □



New Child Resource

reate a new child resource for your resource.

s proxy resource □

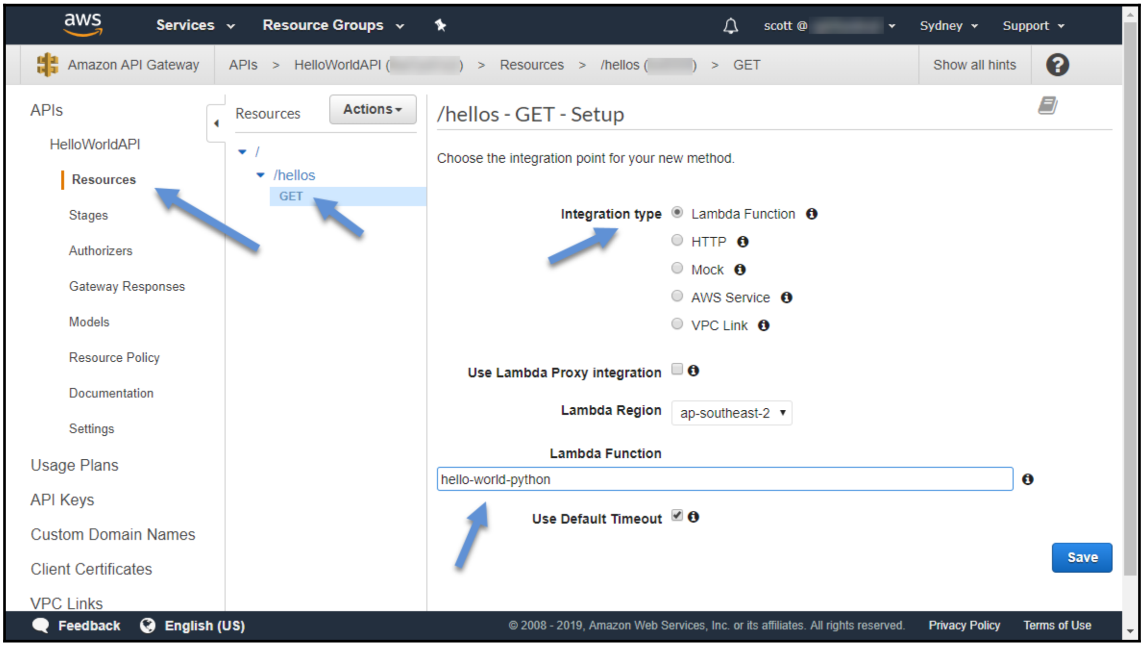
Resource Name* hellos

Resource Path* / hellos

You can add path parameters using brackets. For example, the resource path **{username}** represents a path parameter called 'username'. Configuring /{proxy+} as a proxy resource catches all requests to its sub-resources. For example, it works for a GET request to /foo. To handle requests to /, add a new ANY method on the / resource.

Enable API Gateway CORS □

* **Required**



/hellos - GET - Setup

Choose the integration point for your new method.

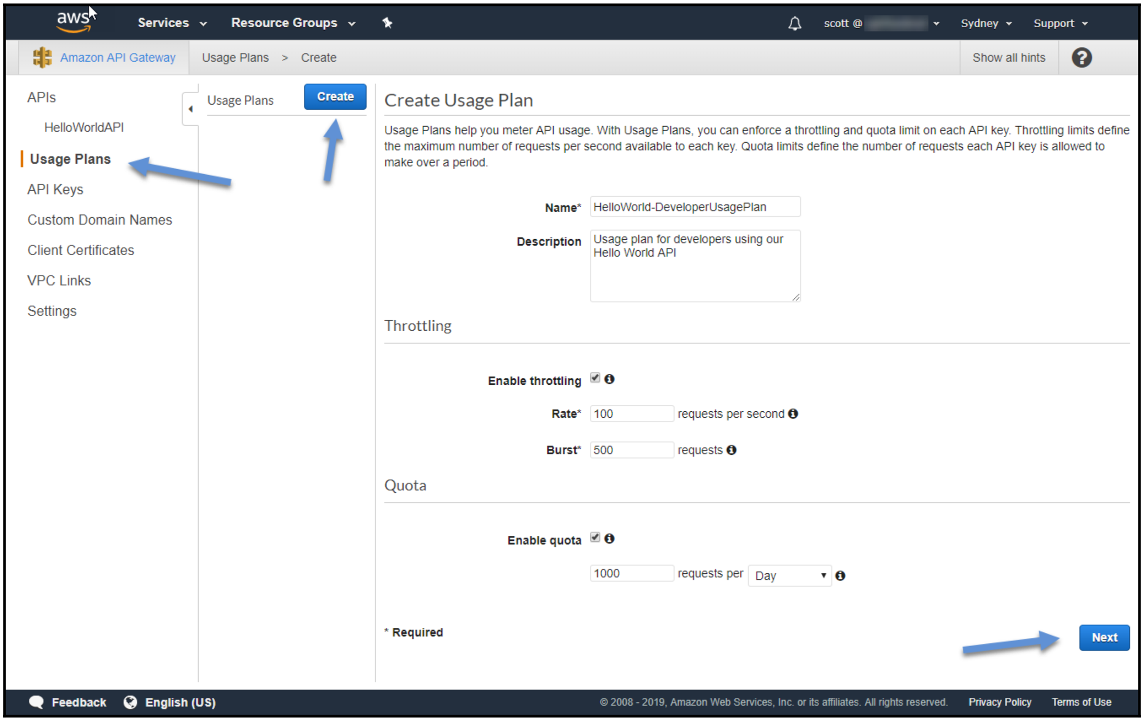
Integration type ● Lambda Function ○ HTTP ○ Mock ○ AWS Service ○ VPC Link

Use Lambda Proxy integration □

Lambda Region ap-southeast-2

Lambda Function hello-world-python

Use Default Timeout ☑



Create Usage Plan

Usage Plans help you meter API usage. With Usage Plans, you can enforce a throttling and quota limit on each API key. Throttling limits define the maximum number of requests per second available to each key. Quota limits define the number of requests each API key is allowed to make over a period.

Name* HelloWorld-DeveloperUsagePlan

Description Usage plan for developers using our Hello World API

Throttling

Enable throttling ☑

Rate* 100 requests per second

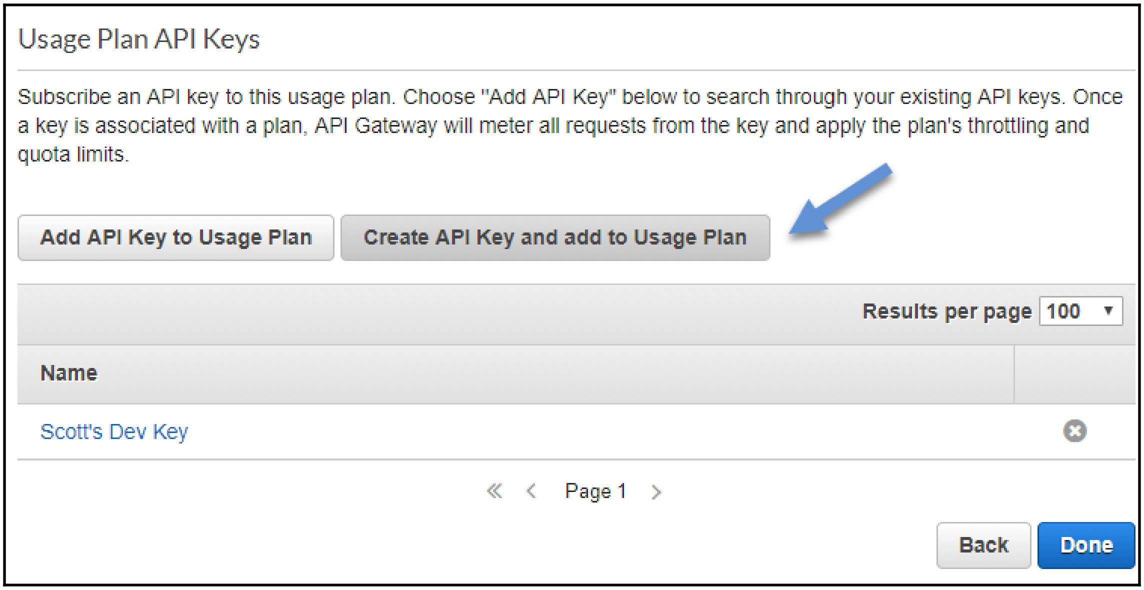
Burst* 500 requests

Quota

Enable quota ☑

1000 requests per Day

* Required

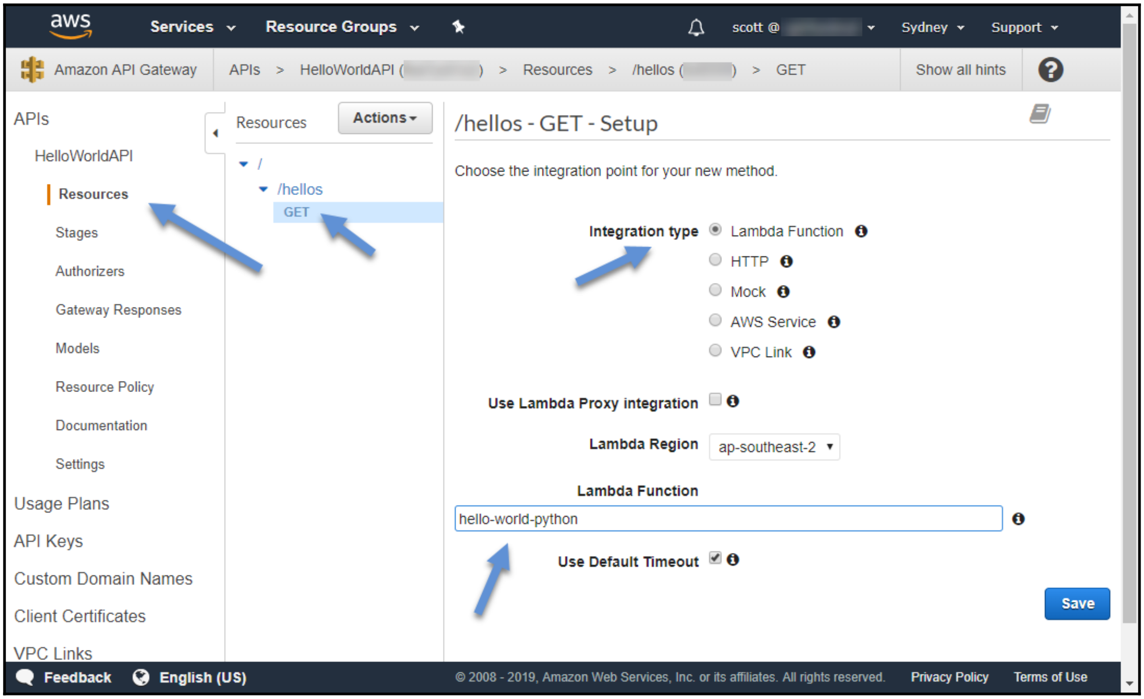


Usage Plan API Keys

Subscribe an API key to this usage plan. Choose "Add API Key" below to search through your existing API keys. Once a key is associated with a plan, API Gateway will meter all requests from the key and apply the plan's throttling and quota limits.

Add API Key to Usage Plan
Create API Key and add to Usage Plan
Results per page 100
Page 1
Back
Done

Name
Scott's Dev Key



/hellos - GET - Setup

Choose the integration point for your new method.

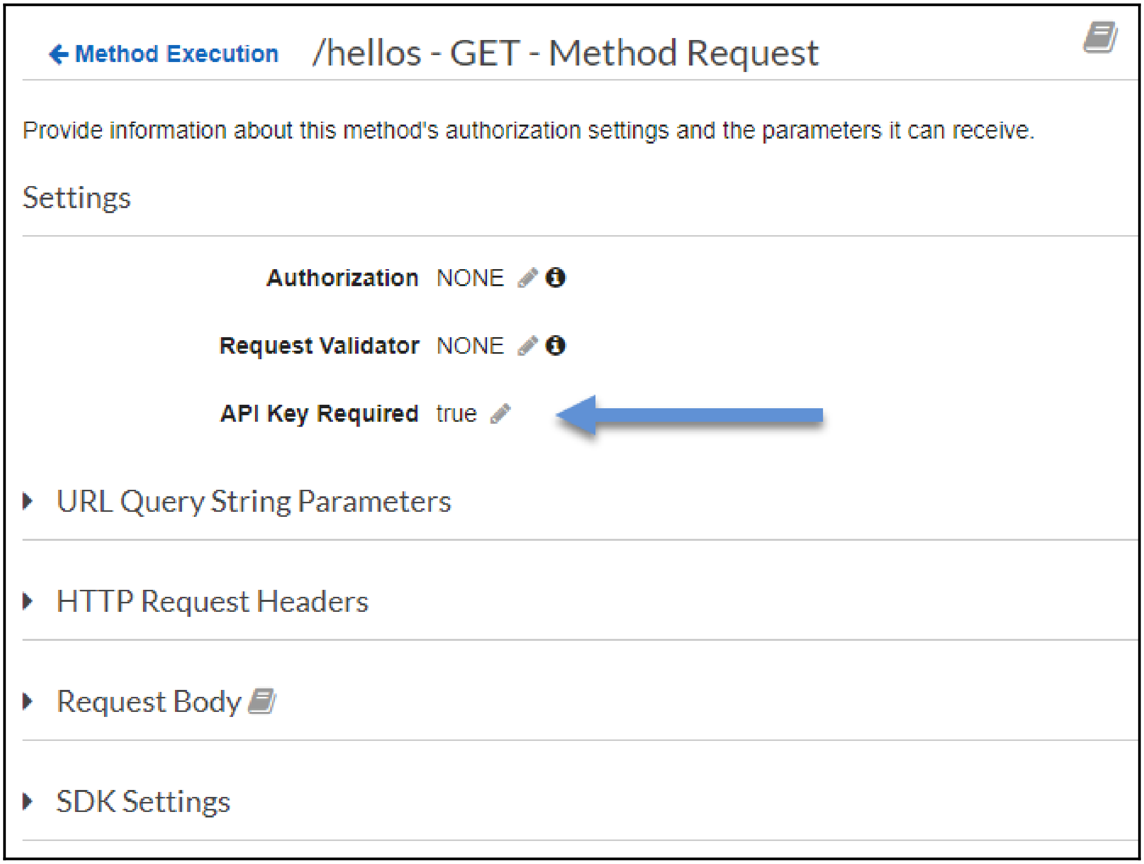
Integration type ● Lambda Function ○ HTTP ○ Mock ○ AWS Service ○ VPC Link

Use Lambda Proxy integration □

Lambda Region ap-southeast-2

Lambda Function hello-world-python

Use Default Timeout ☑



/hellos - GET - Method Request

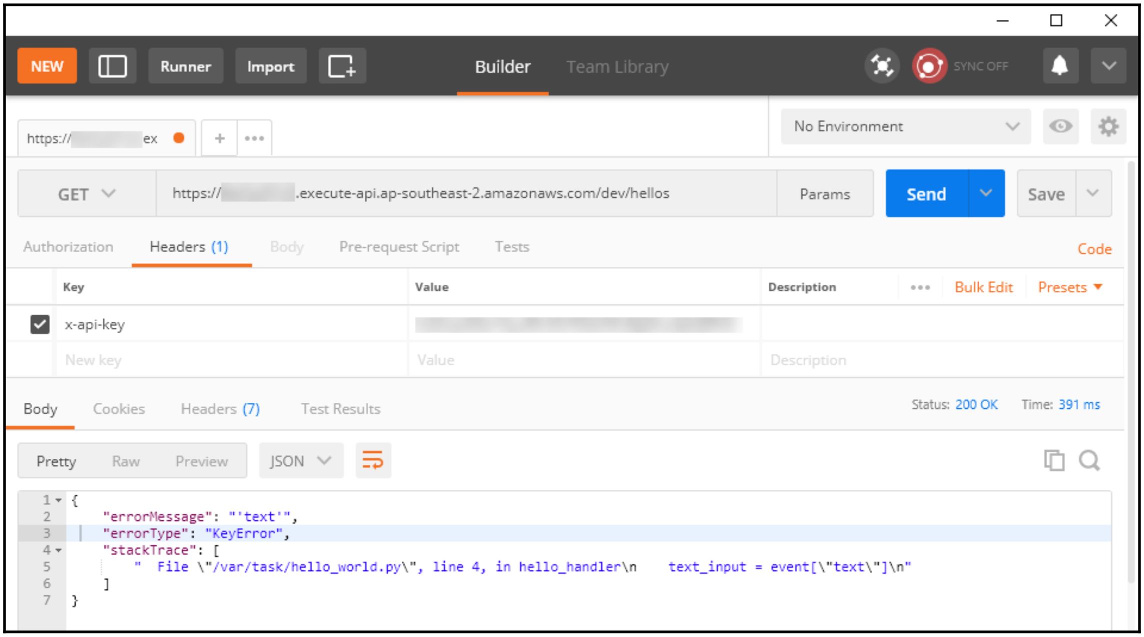
Provide information about this method's authorization settings and the parameters it can receive.

Settings

Authorization NONE

Request Validator NONE

API Key Required true



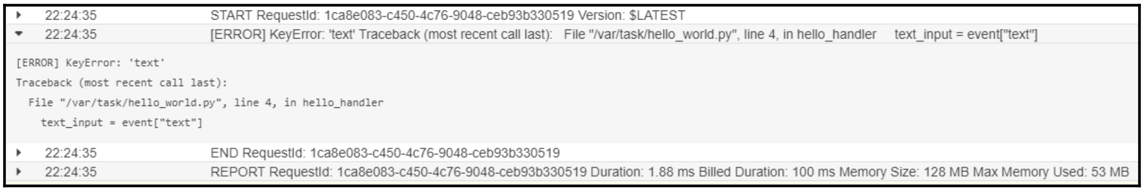
https://[redacted]ex

GET https://[redacted].execute-api.ap-southeast-2.amazonaws.com/dev/hellos

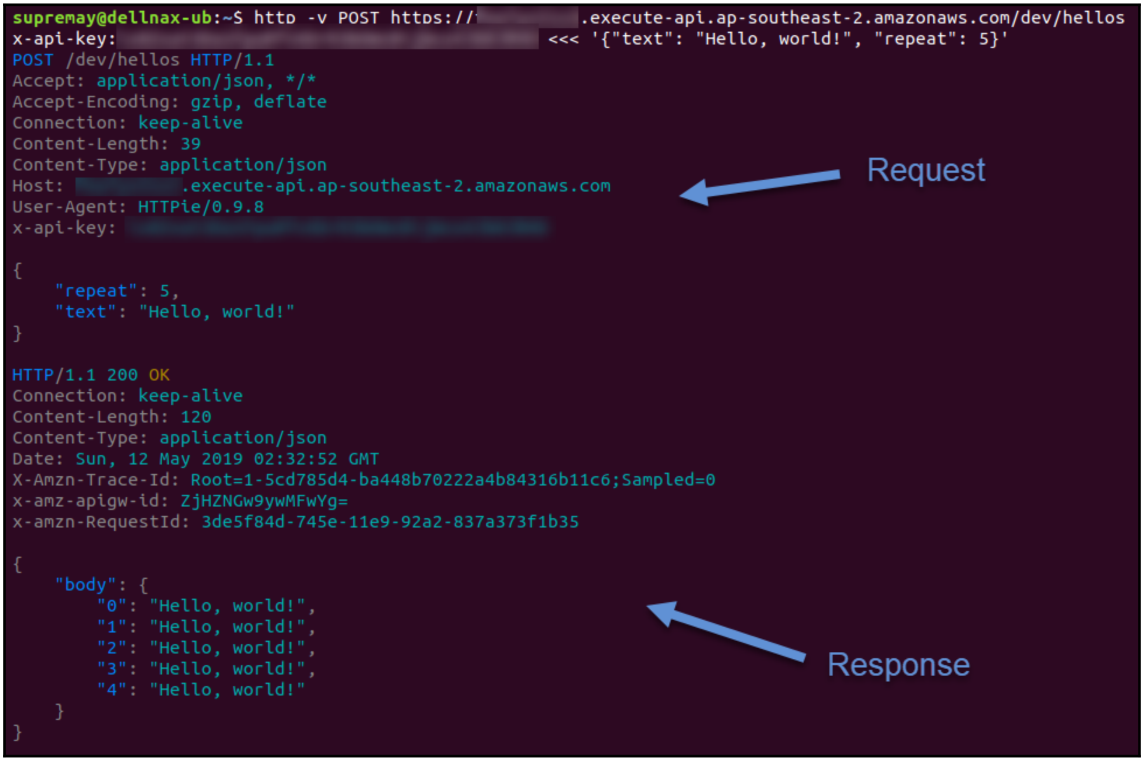
	Key	Value	Description
☑	x-api-key	[redacted]	
	New key	Value	Description

Status: 200 OK Time: 391 ms

```
{
    "errorMessage": "'text'",
    "errorType": "KeyError",
    "stackTrace": [
        "  File \"/var/task/hello_world.py\", line 4, in hello_handler\n    text_input = event[\"text\"]\n"
    ]
}
```

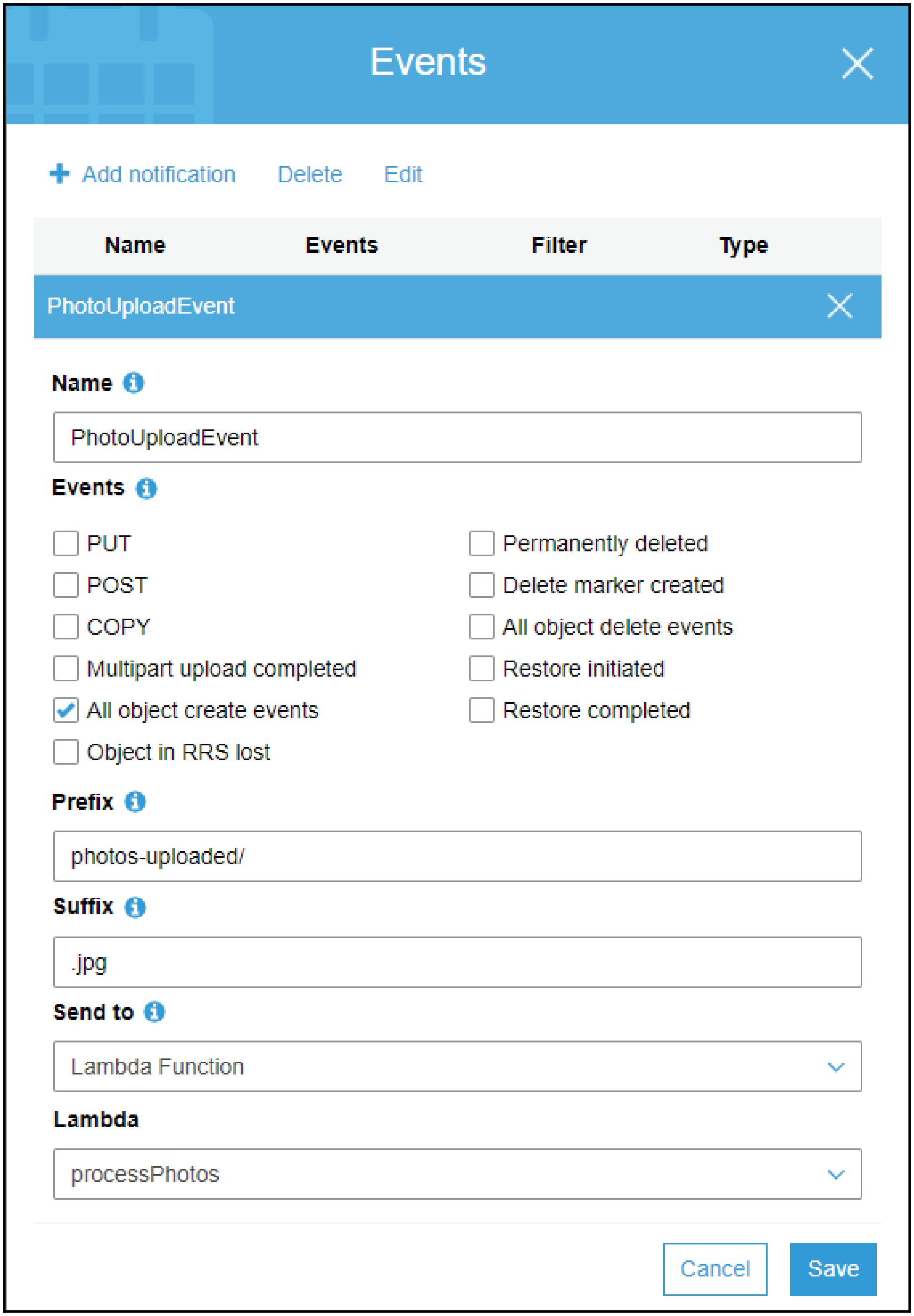


```
▸ 22:24:35    START RequestId: 1ca8e083-c450-4c76-9048-ceb93b330519 Version: $LATEST
▾ 22:24:35    [ERROR] KeyError: 'text' Traceback (most recent call last):   File "/var/task/hello_world.py", line 4, in hello_handler    text_input = event["text"]

[ERROR] KeyError: 'text'
Traceback (most recent call last):
  File "/var/task/hello_world.py", line 4, in hello_handler
    text_input = event["text"]

▸ 22:24:35    END RequestId: 1ca8e083-c450-4c76-9048-ceb93b330519
▸ 22:24:35    REPORT RequestId: 1ca8e083-c450-4c76-9048-ceb93b330519 Duration: 1.88 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 53 MB
```



```
supremay@dellnax-ub:~$ http -v POST https://[redacted].execute-api.ap-southeast-2.amazonaws.com/dev/hellos
x-api-key:[redacted] <<< '{"text": "Hello, world!", "repeat": 5}'
POST /dev/hellos HTTP/1.1
Accept: application/json, */*
Accept-Encoding: gzip, deflate
Connection: keep-alive
Content-Length: 39
Content-Type: application/json
Host: [redacted].execute-api.ap-southeast-2.amazonaws.com
User-Agent: HTTPie/0.9.8
x-api-key: [redacted]

{
    "repeat": 5,
    "text": "Hello, world!"
}

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 120
Content-Type: application/json
Date: Sun, 12 May 2019 02:32:52 GMT
X-Amzn-Trace-Id: Root=1-5cd785d4-ba448b70222a4b84316b11c6;Sampled=0
x-amz-apigw-id: ZjHZNGw9ywMFwYg=
x-amzn-RequestId: 3de5f84d-745e-11e9-92a2-837a373f1b35

{
    "body": {
        "0": "Hello, world!",
        "1": "Hello, world!",
        "2": "Hello, world!",
        "3": "Hello, world!",
        "4": "Hello, world!"
    }
}
```

Chapter 5: Leveraging AWS Services



Name	Events	Filter	Type

PhotoUploadEvent

Name

PhotoUploadEvent

Events

☐ PUT
☐ POST
☐ COPY
☐ Multipart upload completed
☑ All object create events
☐ Object in RRS lost
☐ Permanently deleted
☐ Delete marker created
☐ All object delete events
☐ Restore initiated
☐ Restore completed

Prefix

photos-uploaded/

Suffix

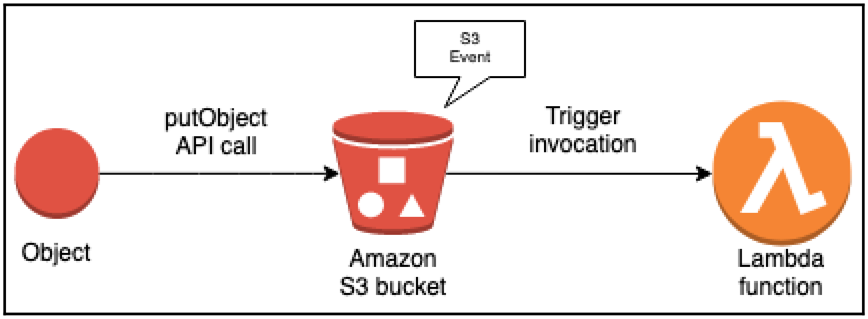
.jpg

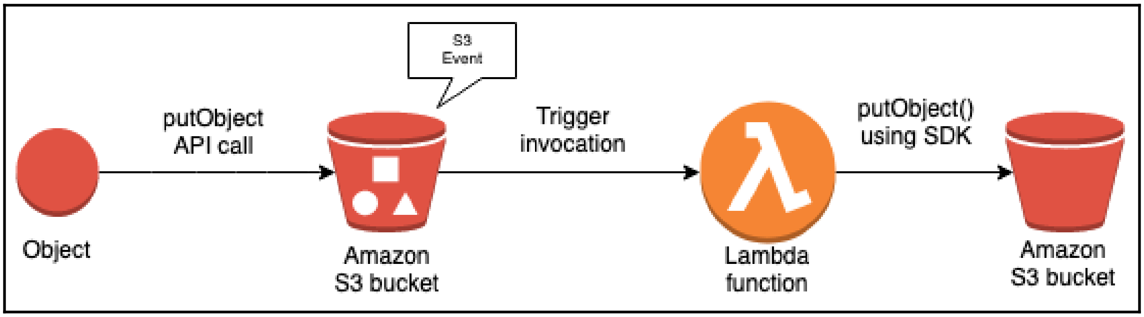
Send to

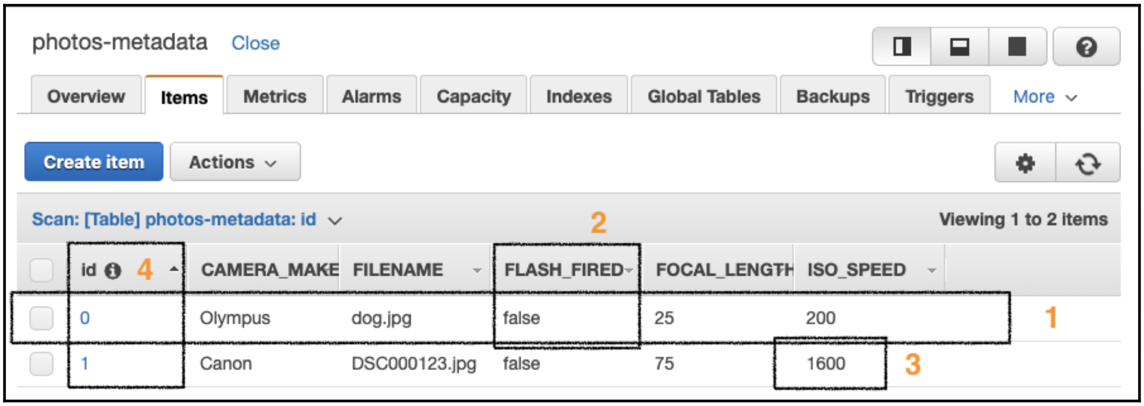
Lambda Function

Lambda

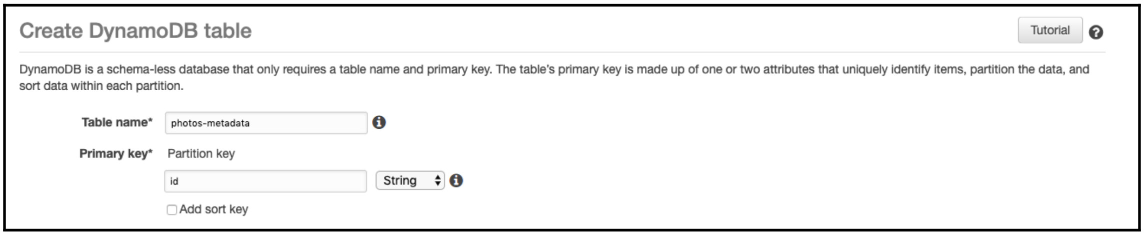
processPhotos







id	CAMERA_MAKE	FILENAME	FLASH_FIRED	FOCAL_LENGTH	ISO_SPEED
0	Olympus	dog.jpg	false	25	200
1	Canon	DSC000123.jpg	false	75	1600



Create DynamoDB table

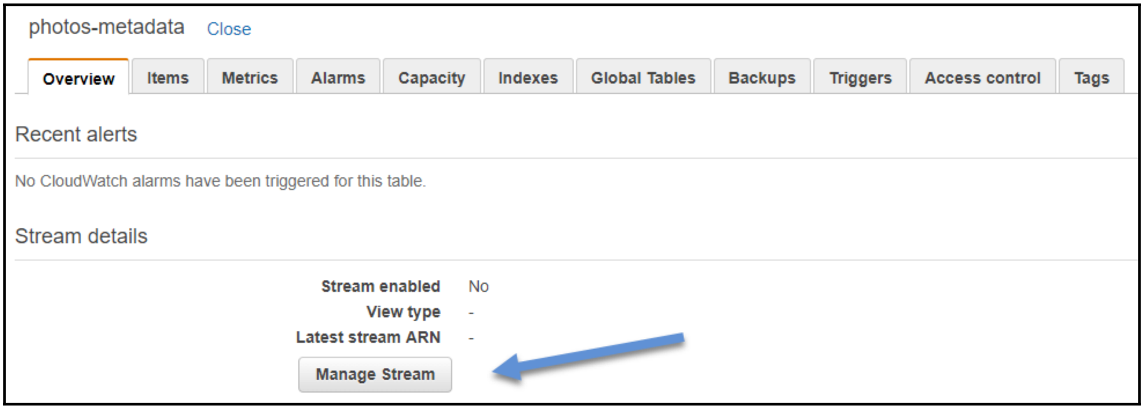
DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* photos-metadata

Primary key* Partition key

id String

□ Add sort key

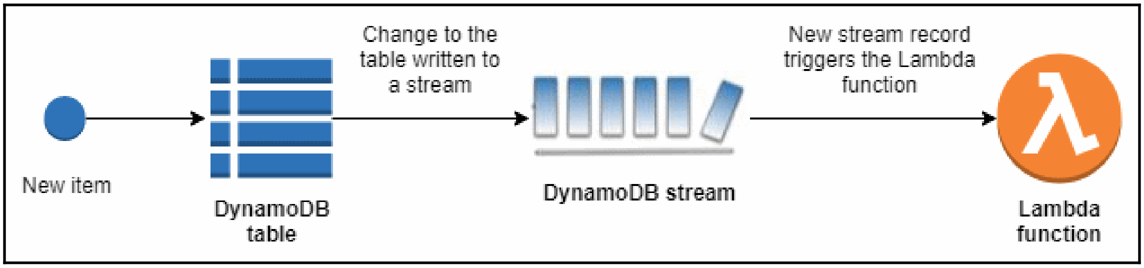


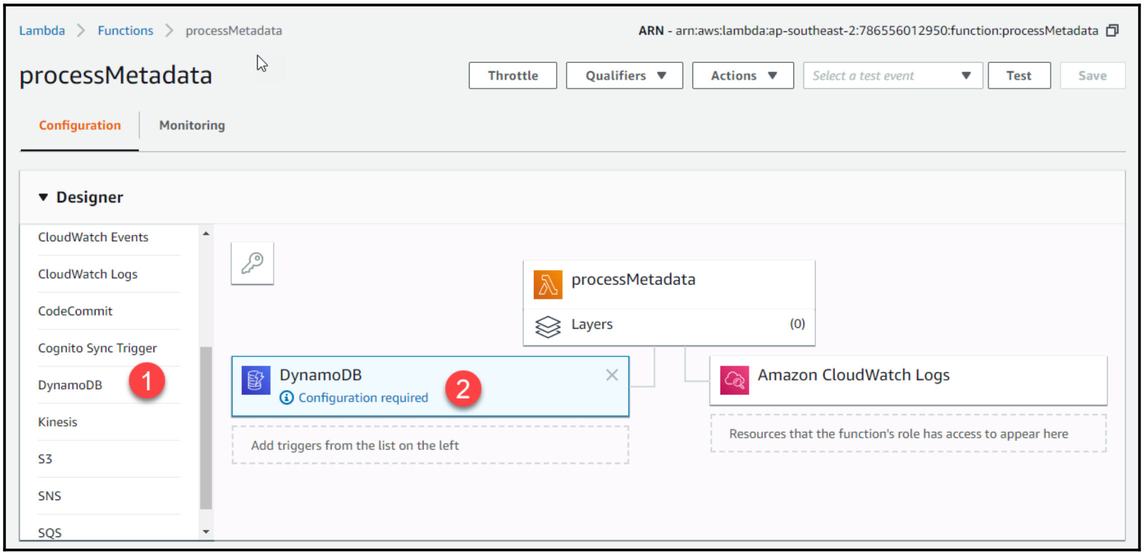
Recent alerts

No CloudWatch alarms have been triggered for this table.

Stream details

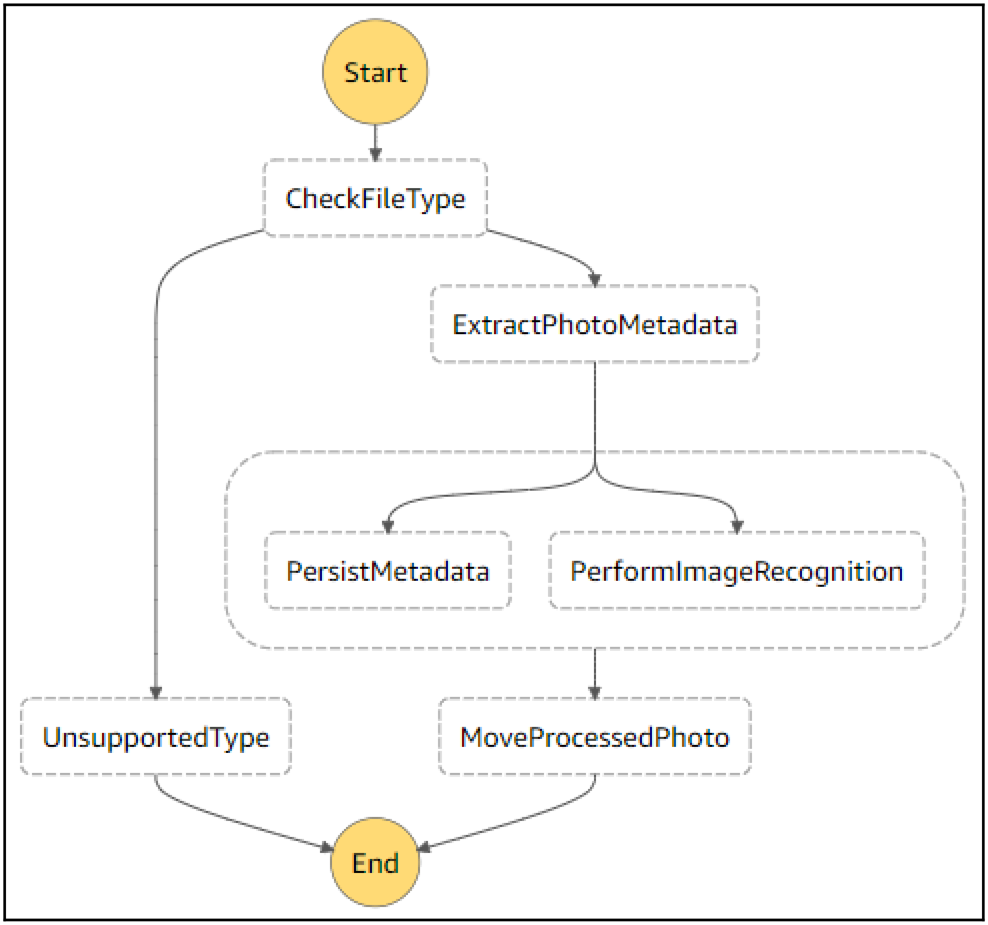
Stream enabled No
View type -
Latest stream ARN -


New item
DynamoDB table
Change to the table written to a stream
DynamoDB stream
New stream record triggers the Lambda function
Lambda function

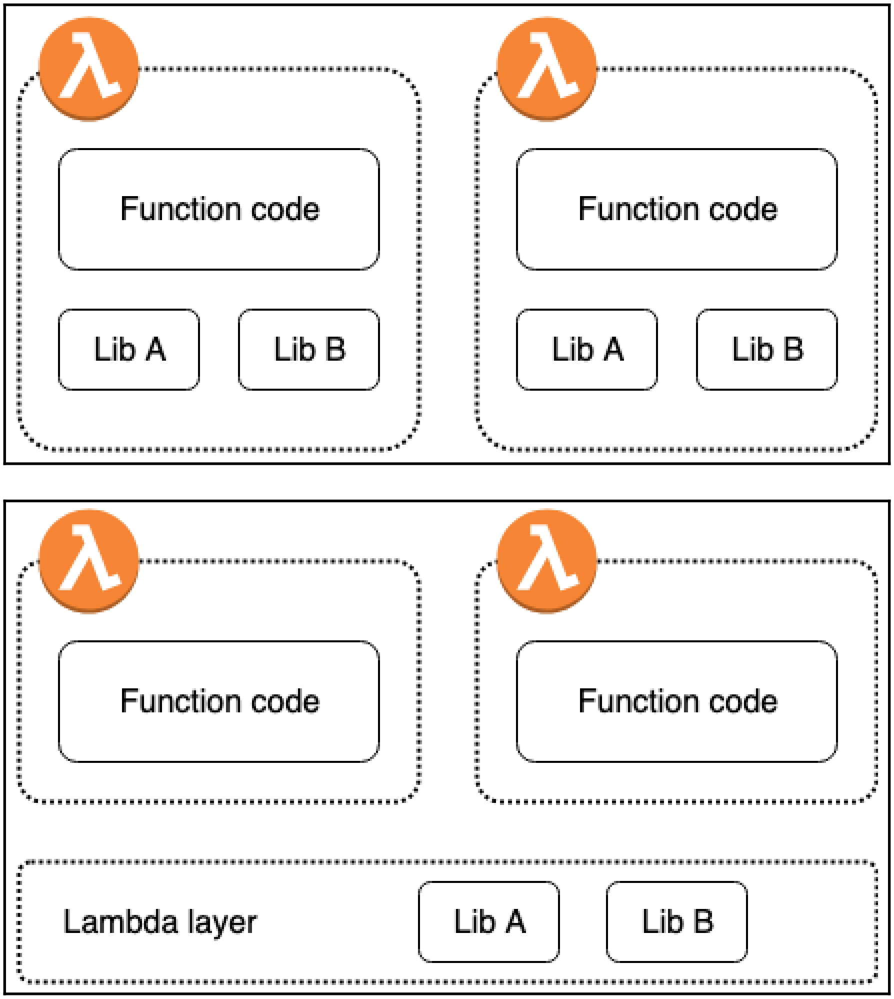


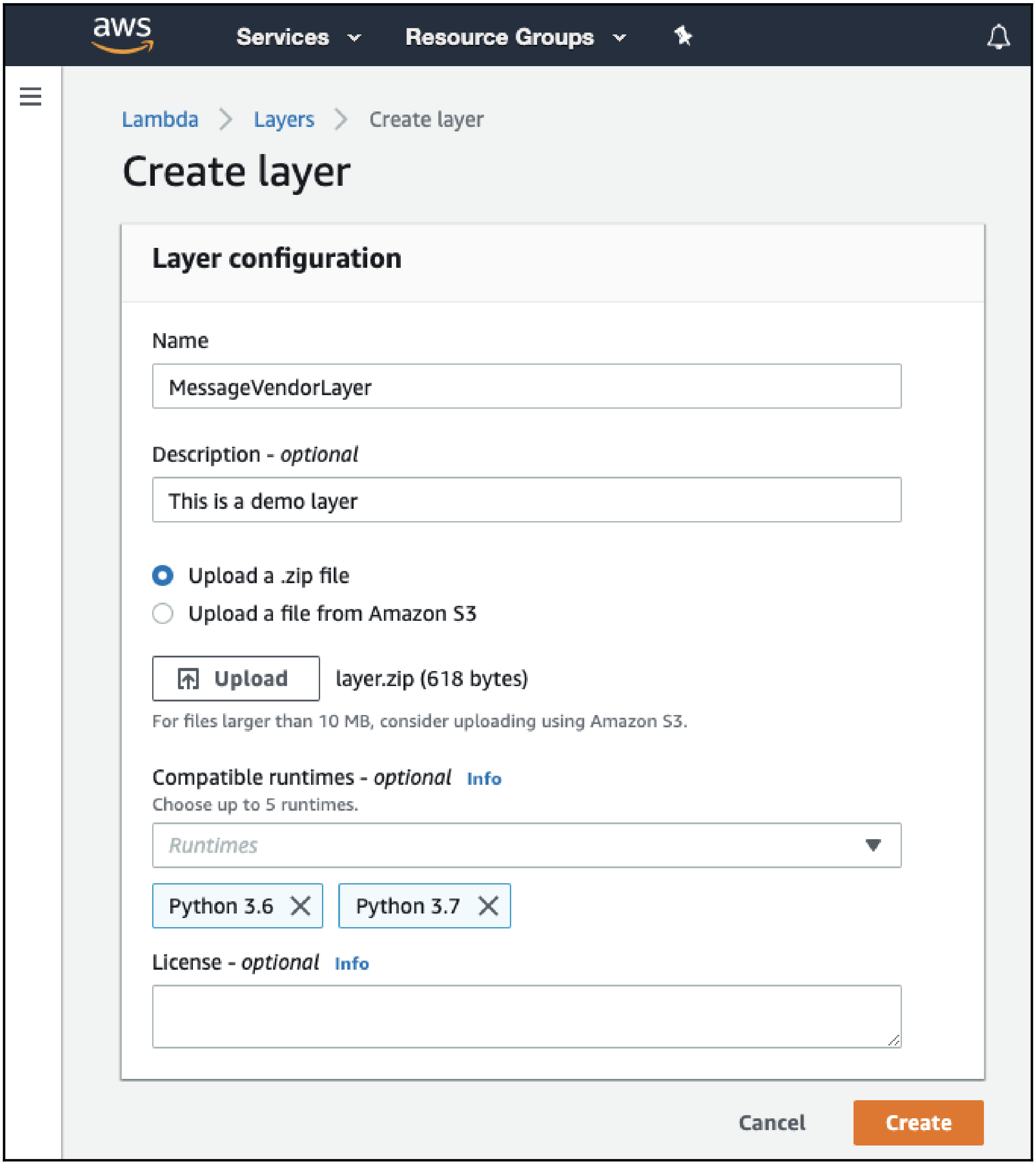
ARN - arn:aws:lambda:ap-southeast-2:786556012950:function:processMetadata

processMetadata



Chapter 6: Going Deeper with Lambda


Function code
Lib A
Lib B
Function code
Lib A
Lib B
Function code
Function code
Lambda layer
Lib A
Lib B



Create layer

Layer configuration

Name

MessageVendorLayer

Description - *optional*

This is a demo layer

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

layer.zip (618 bytes)

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

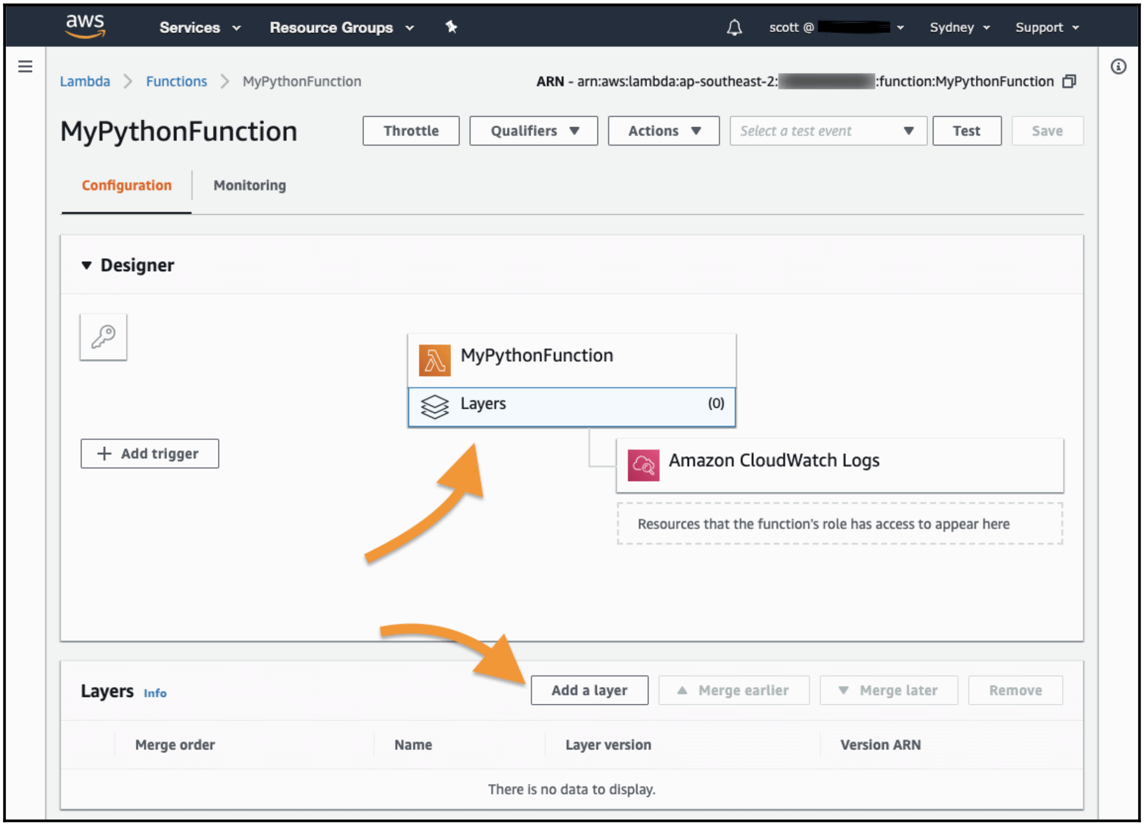
Compatible runtimes - *optional* Info

Choose up to 5 runtimes.

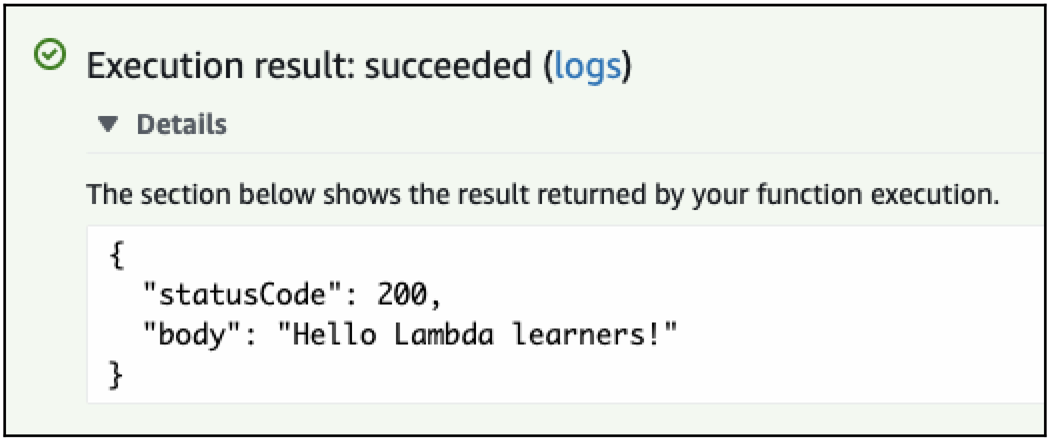
Runtimes

Python 3.6 ✕ Python 3.7 ✕

License - *optional* Info



Merge order	Name	Layer version	Version ARN
There is no data to display.			

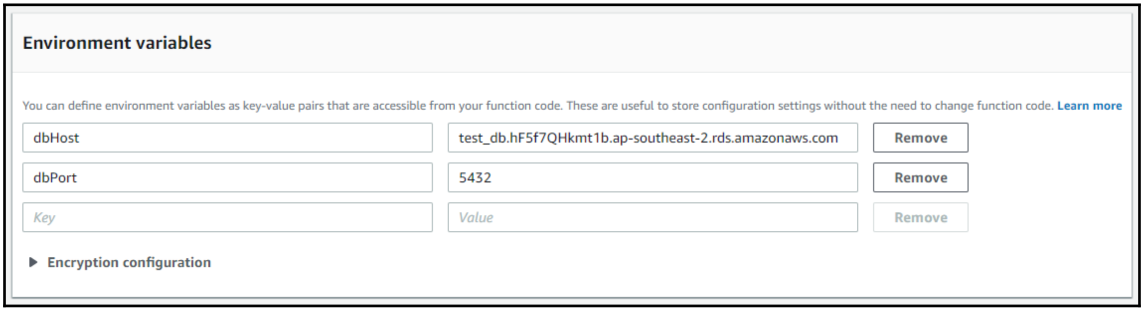


Execution result: succeeded (logs)

Details

The section below shows the result returned by your function execution.

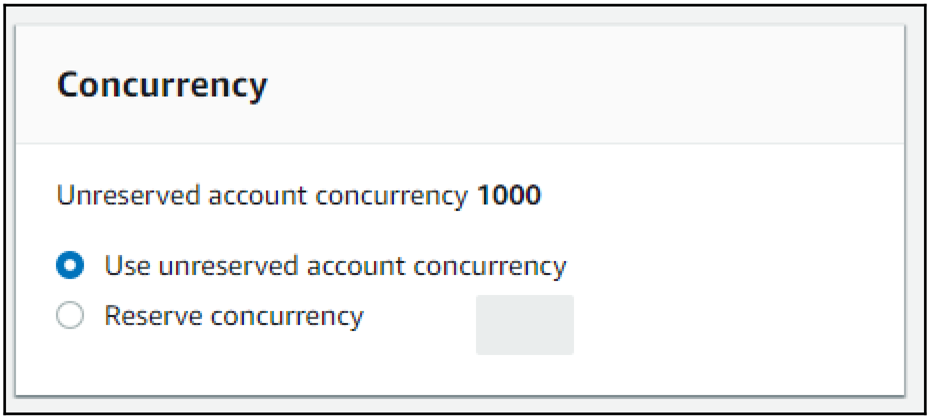
```
{
  "statusCode": 200,
  "body": "Hello Lambda learners!"
}
```



Environment variables

You can define environment variables as key-value pairs that are accessible from your function code. These are useful to store configuration settings without the need to change function code. **Learn more**

dbHost	test_db.hF5f7QHkmt1b.ap-southeast-2.rds.amazonaws.com	Remove
dbPort	5432	Remove
Key	*Value*	Remove

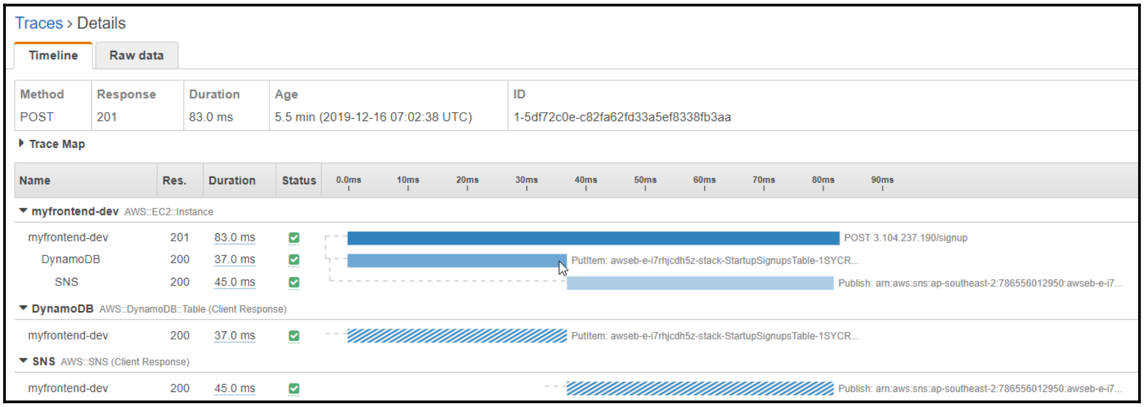


Concurrency

Unreserved account concurrency **1000**

● Use unreserved account concurrency

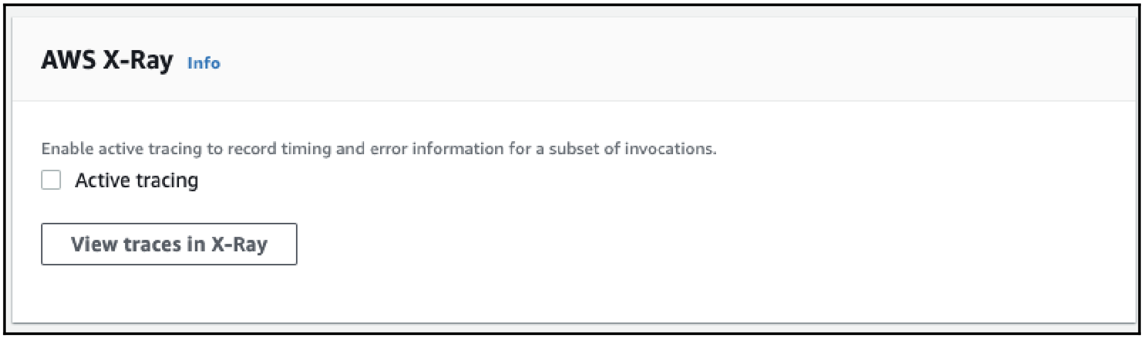
○ Reserve concurrency



Traces › Details

Method	Response	Duration	Age	ID
POST	201	83.0 ms	5.5 min (2019-12-16 07:02:38 UTC)	1-5df72c0e-c82fa62fd33a5ef8338fb3aa

Name	Res.	Duration	Status	
▼ **myfrontend-dev** AWS::EC2::Instance				
myfrontend-dev	201	83.0 ms	✓	POST 3.104.237.190/signup
DynamoDB	200	37.0 ms	✓	PutItem: awseb-e-i7rhjcdh5z-stack-StartupSignupsTable-1SYCR...
SNS	200	45.0 ms	✓	Publish: arn:aws:sns:ap-southeast-2:786556012950:awseb-e-i7...
▼ **DynamoDB** AWS::DynamoDB::Table (Client Response)				
myfrontend-dev	200	37.0 ms	✓	PutItem: awseb-e-i7rhjcdh5z-stack-StartupSignupsTable-1SYCR...
▼ **SNS** AWS::SNS (Client Response)				
myfrontend-dev	200	45.0 ms	✓	Publish: arn:aws:sns:ap-southeast-2:786556012950:awseb-e-i7...

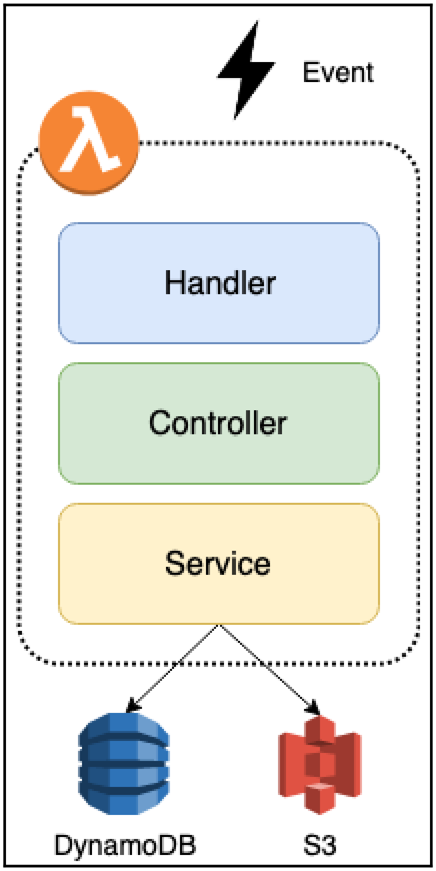


AWS X-Ray Info

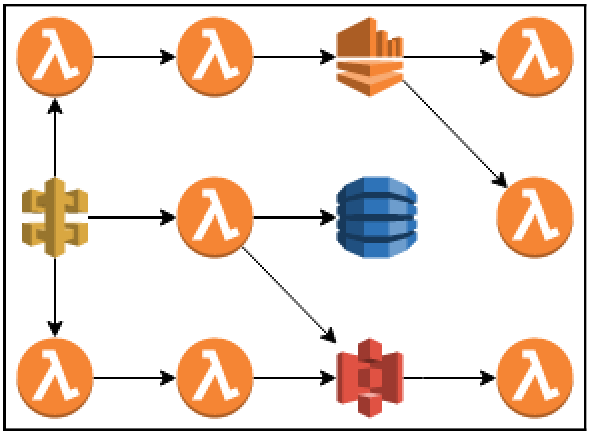
Enable active tracing to record timing and error information for a subset of invocations.

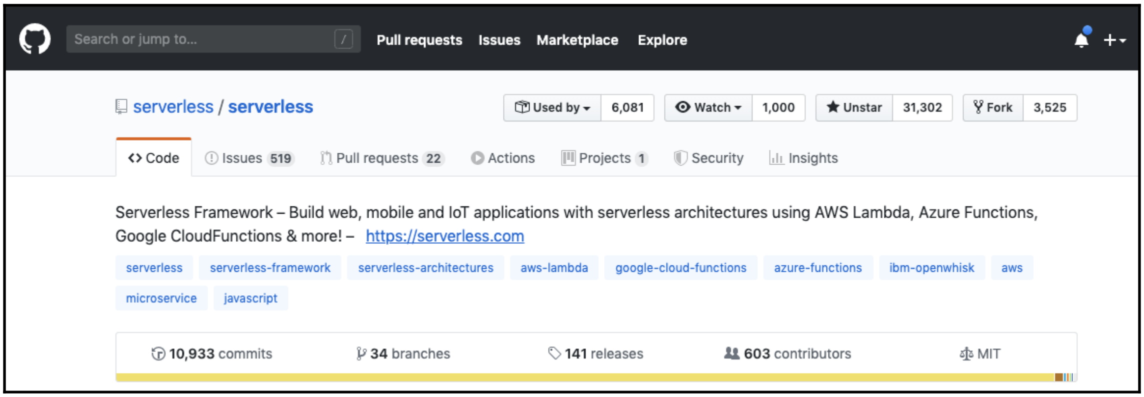
□ Active tracing

View traces in X-Ray



Chapter 7: Serverless Framework





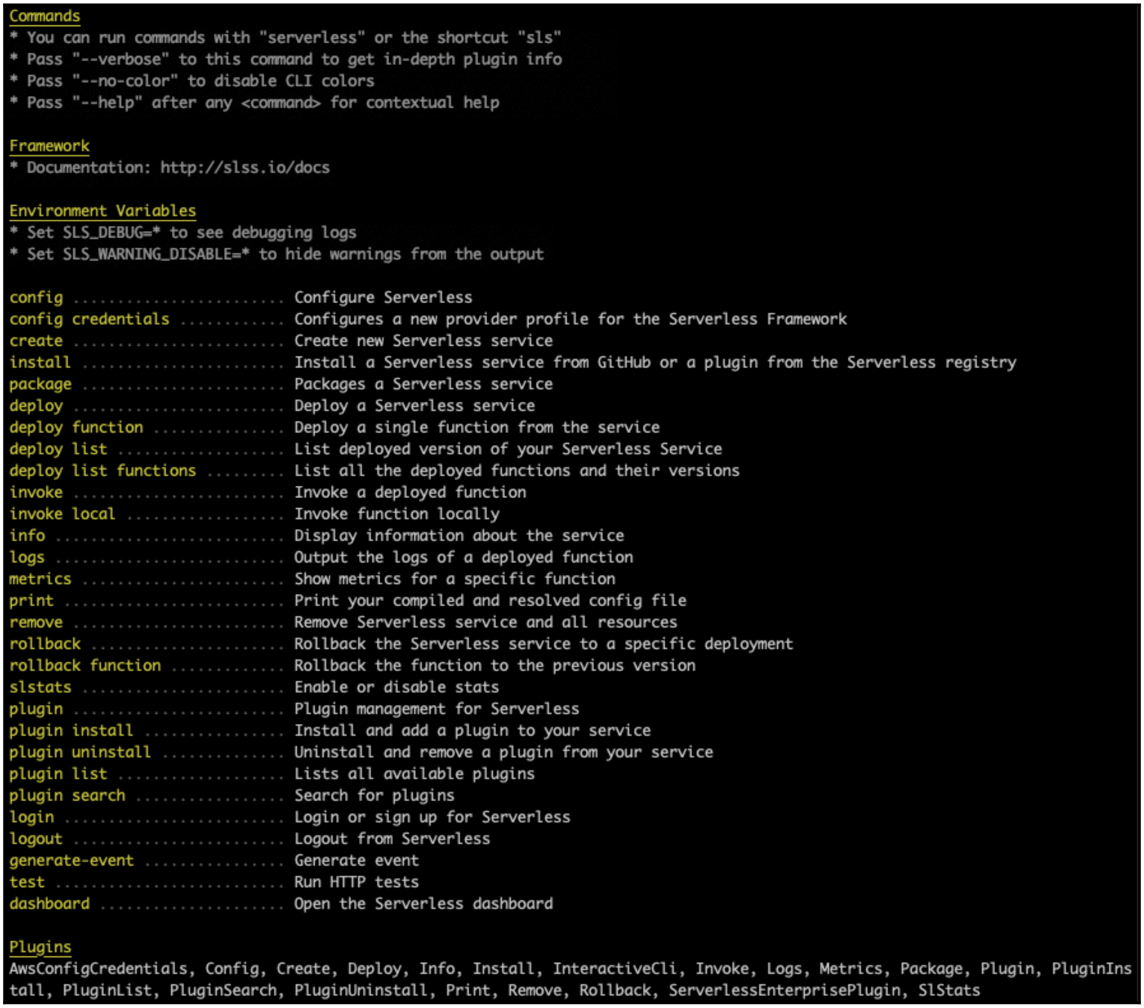
serverless / serverless

Used by 6,081 | Watch 1,000 | Unstar 31,302 | Fork 3,525

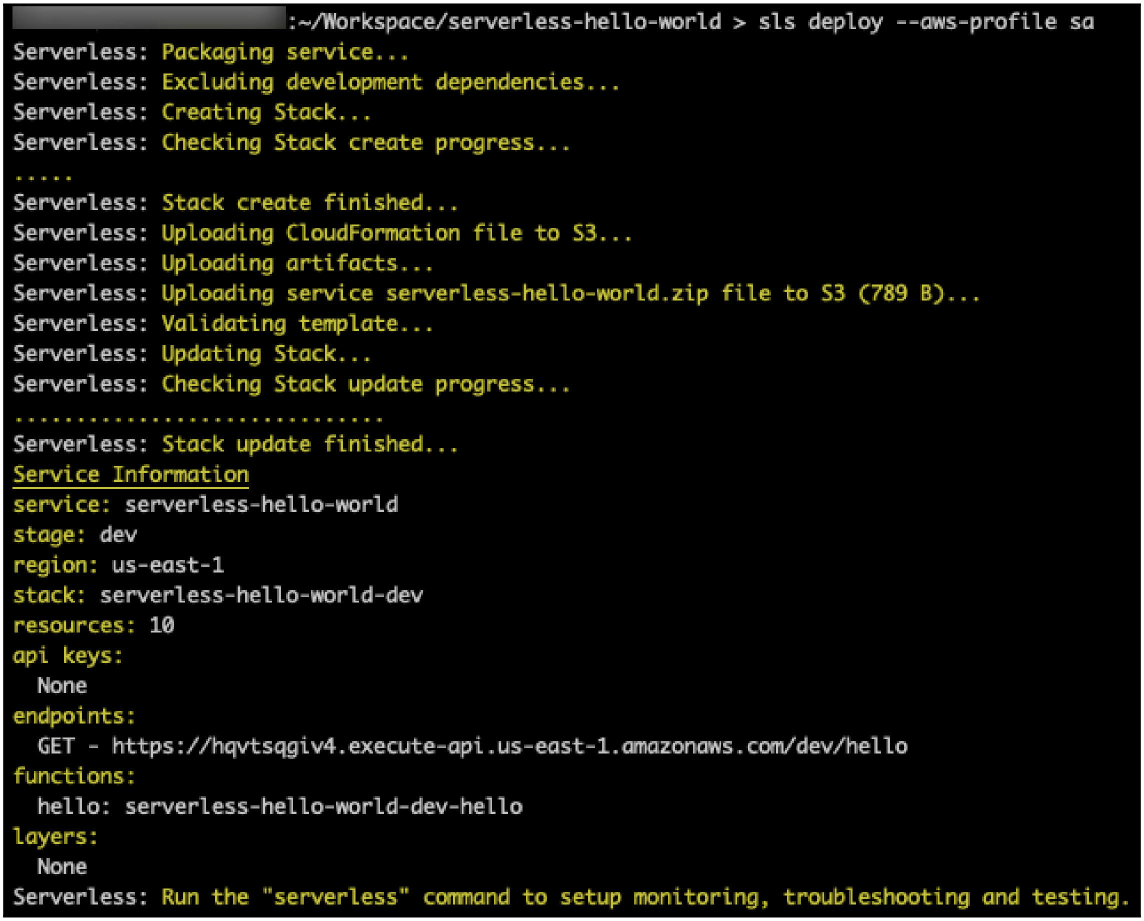
Serverless Framework – Build web, mobile and IoT applications with serverless architectures using AWS Lambda, Azure Functions, Google CloudFunctions & more! – https://serverless.com

serverless serverless-framework serverless-architectures aws-lambda google-cloud-functions azure-functions ibm-openwhisk aws microservice javascript

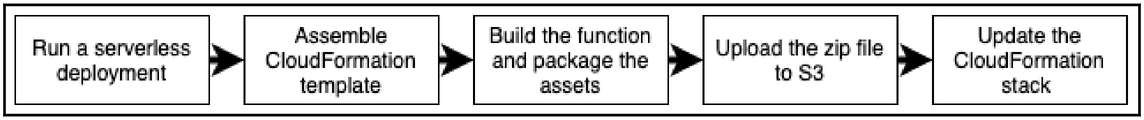
10,933 commits | 34 branches | 141 releases | 603 contributors | MIT

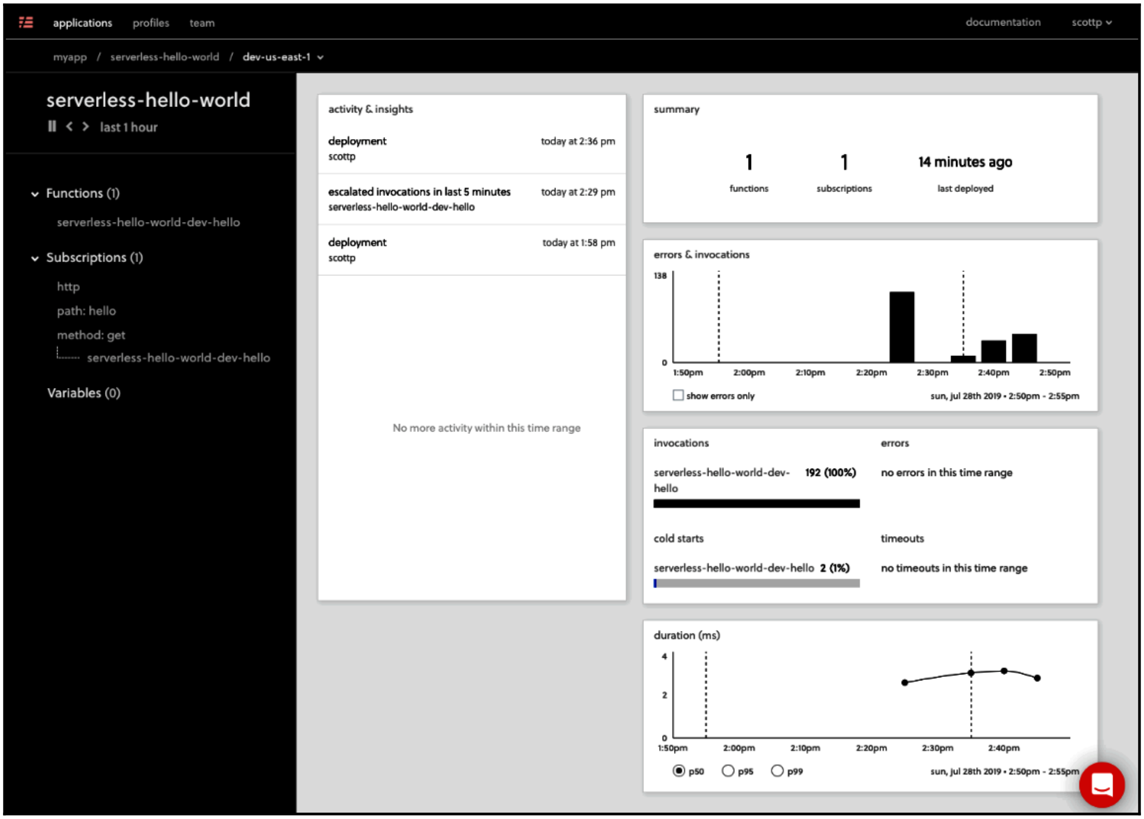


```
Commands
* You can run commands with "serverless" or the shortcut "sls"
* Pass "--verbose" to this command to get in-depth plugin info
* Pass "--no-color" to disable CLI colors
* Pass "--help" after any <command> for contextual help

Framework
* Documentation: http://slss.io/docs

Environment Variables
* Set SLS_DEBUG=* to see debugging logs
* Set SLS_WARNING_DISABLE=* to hide warnings from the output

config ........................ Configure Serverless
config credentials ............ Configures a new provider profile for the Serverless Framework
create ........................ Create new Serverless service
install ....................... Install a Serverless service from GitHub or a plugin from the Serverless registry
package ....................... Packages a Serverless service
deploy ........................ Deploy a Serverless service
deploy function ............... Deploy a single function from the service
deploy list ................... List deployed version of your Serverless Service
deploy list functions ......... List all the deployed functions and their versions
invoke ........................ Invoke a deployed function
invoke local .................. Invoke function locally
info .......................... Display information about the service
logs .......................... Output the logs of a deployed function
metrics ....................... Show metrics for a specific function
print ......................... Print your compiled and resolved config file
remove ........................ Remove Serverless service and all resources
rollback ...................... Rollback the Serverless service to a specific deployment
rollback function ............. Rollback the function to the previous version
slstats ....................... Enable or disable stats
plugin ........................ Plugin management for Serverless
plugin install ................ Install and add a plugin to your service
plugin uninstall .............. Uninstall and remove a plugin from your service
plugin list ................... Lists all available plugins
plugin search ................. Search for plugins
login ......................... Login or sign up for Serverless
logout ........................ Logout from Serverless
generate-event ................ Generate event
test .......................... Run HTTP tests
dashboard ..................... Open the Serverless dashboard

Plugins
AwsConfigCredentials, Config, Create, Deploy, Info, Install, InteractiveCli, Invoke, Logs, Metrics, Package, Plugin, PluginIns
tall, PluginList, PluginSearch, PluginUninstall, Print, Remove, Rollback, ServerlessEnterprisePlugin, SlStats
```



```
[redacted]:~/Workspace/serverless-hello-world > sls deploy --aws-profile sa
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Creating Stack...
Serverless: Checking Stack create progress...
.....
Serverless: Stack create finished...
Serverless: Uploading CloudFormation file to S3...
Serverless: Uploading artifacts...
Serverless: Uploading service serverless-hello-world.zip file to S3 (789 B)...
Serverless: Validating template...
Serverless: Updating Stack...
Serverless: Checking Stack update progress...
..............................
Serverless: Stack update finished...
Service Information
service: serverless-hello-world
stage: dev
region: us-east-1
stack: serverless-hello-world-dev
resources: 10
api keys:
  None
endpoints:
  GET - https://hqvtsqgiv4.execute-api.us-east-1.amazonaws.com/dev/hello
functions:
  hello: serverless-hello-world-dev-hello
layers:
  None
Serverless: Run the "serverless" command to setup monitoring, troubleshooting and testing.
```


Run a serverless deployment
Assemble CloudFormation template
Build the function and package the assets
Upload the zip file to S3
Update the CloudFormation stack



serverless-hello-world

Functions (1)

serverless-hello-world-dev-hello

Subscriptions (1)

http

path: hello

method: get

serverless-hello-world-dev-hello

Variables (0)

activity & insights

deployment today at 2:36 pm
scottp

escalated invocations in last 5 minutes today at 2:29 pm
serverless-hello-world-dev-hello

deployment today at 1:58 pm
scottp

No more activity within this time range

summary

1 functions

1 subscriptions

14 minutes ago last deployed

errors & invocations

□ show errors only

sun, jul 28th 2019 • 2:50pm - 2:55pm

invocations

serverless-hello-world-dev-hello 192 (100%)

errors

no errors in this time range

cold starts

serverless-hello-world-dev-hello 2 (1%)

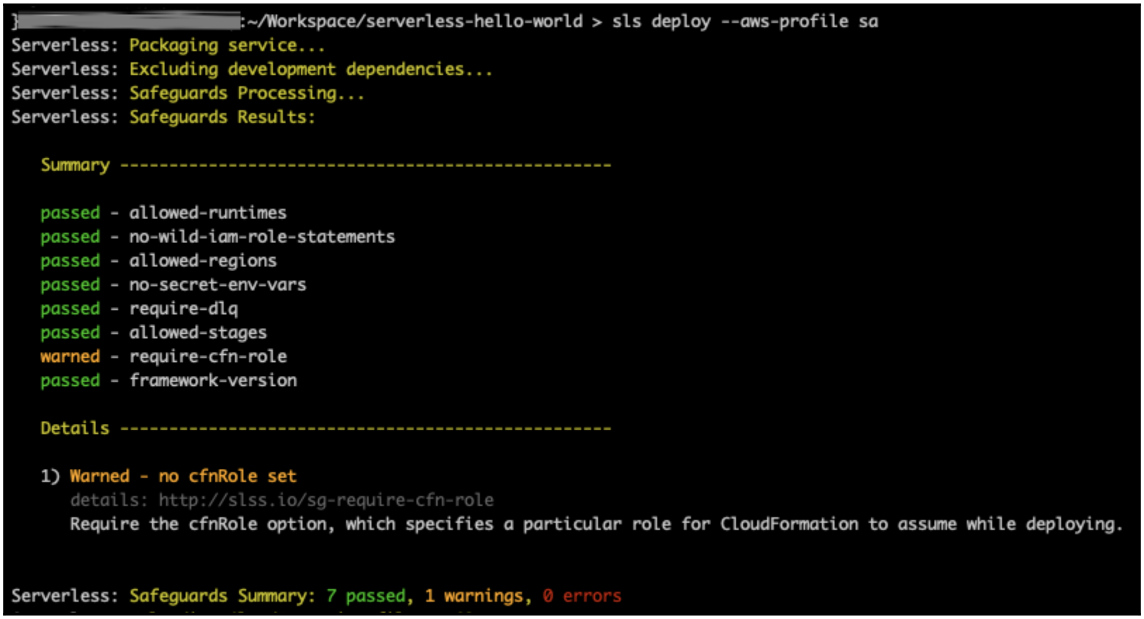
timeouts

no timeouts in this time range

duration (ms)

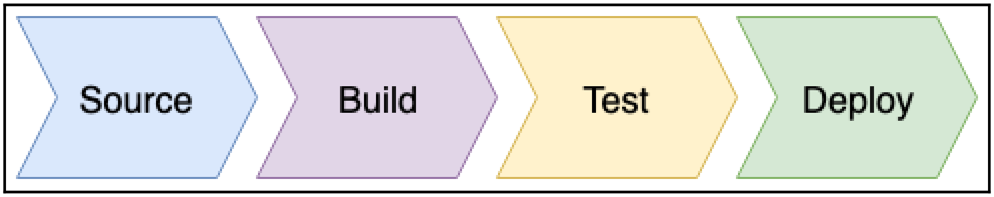
● p50 ○ p95 ○ p99

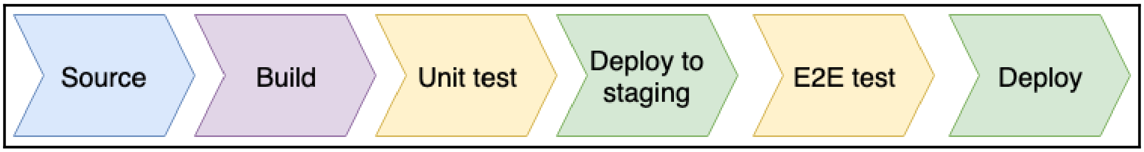
sun, jul 28th 2019 • 2:50pm - 2:55pm

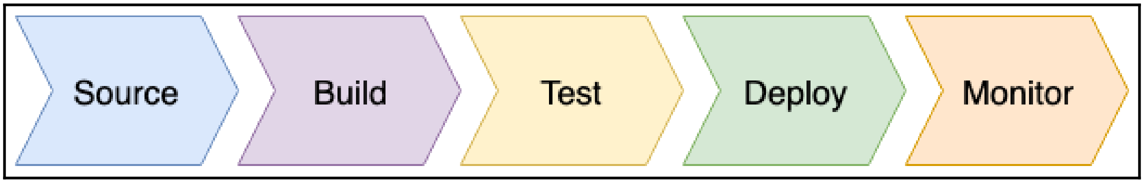


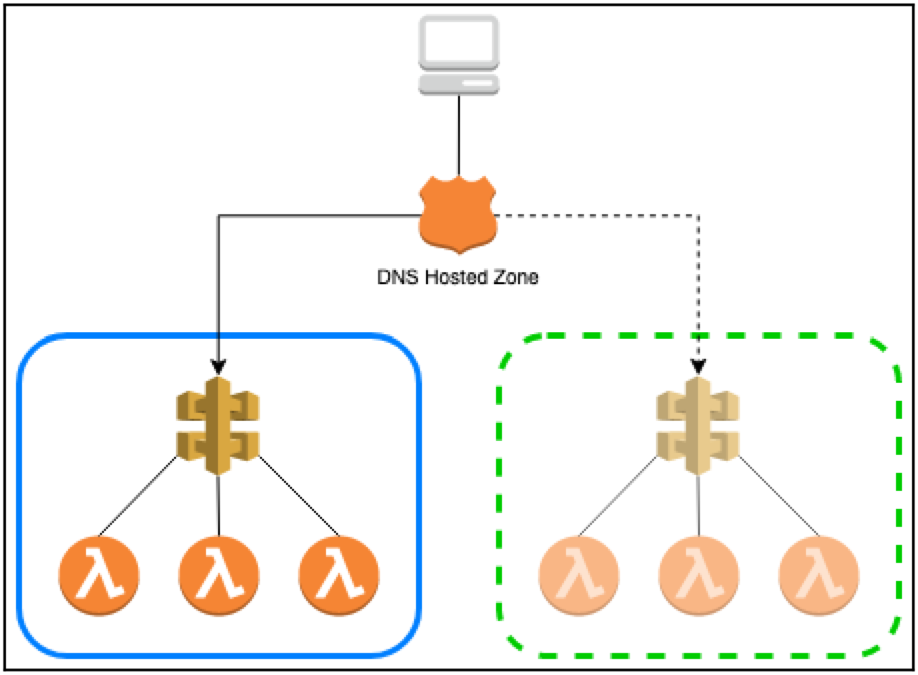
```
}[redacted]:~/Workspace/serverless-hello-world > sls deploy --aws-profile sa
Serverless: Packaging service...
Serverless: Excluding development dependencies...
Serverless: Safeguards Processing...
Serverless: Safeguards Results:

   Summary --------------------------------------------------

   passed - allowed-runtimes
   passed - no-wild-iam-role-statements
   passed - allowed-regions
   passed - no-secret-env-vars
   passed - require-dlq
   passed - allowed-stages
   warned - require-cfn-role
   passed - framework-version

   Details --------------------------------------------------

   1) Warned - no cfnRole set
      details: http://slss.io/sg-require-cfn-role
      Require the cfnRole option, which specifies a particular role for CloudFormation to assume while deploying.


Serverless: Safeguards Summary: 7 passed, 1 warnings, 0 errors
```

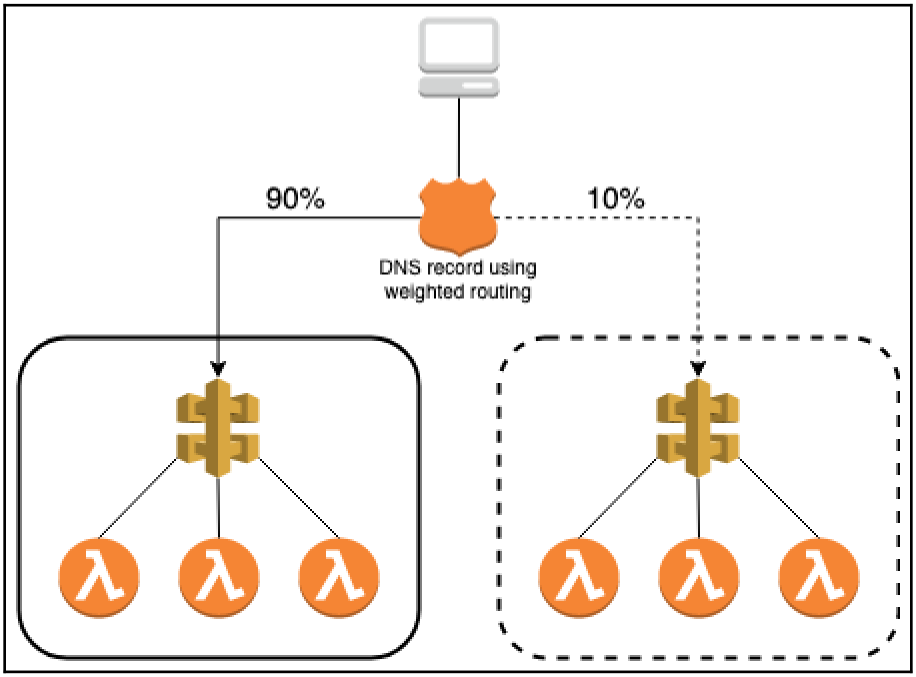
Chapter 8: CI/CD with the Serverless Framework

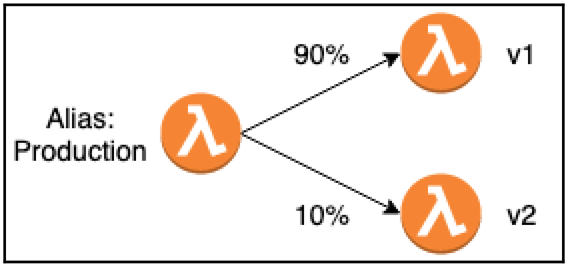


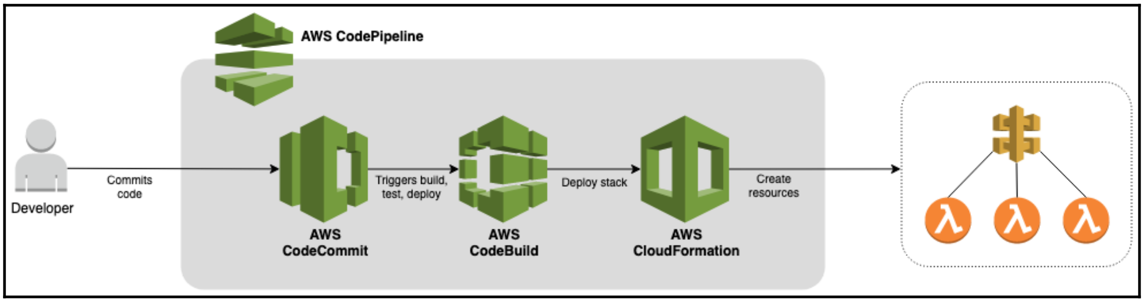


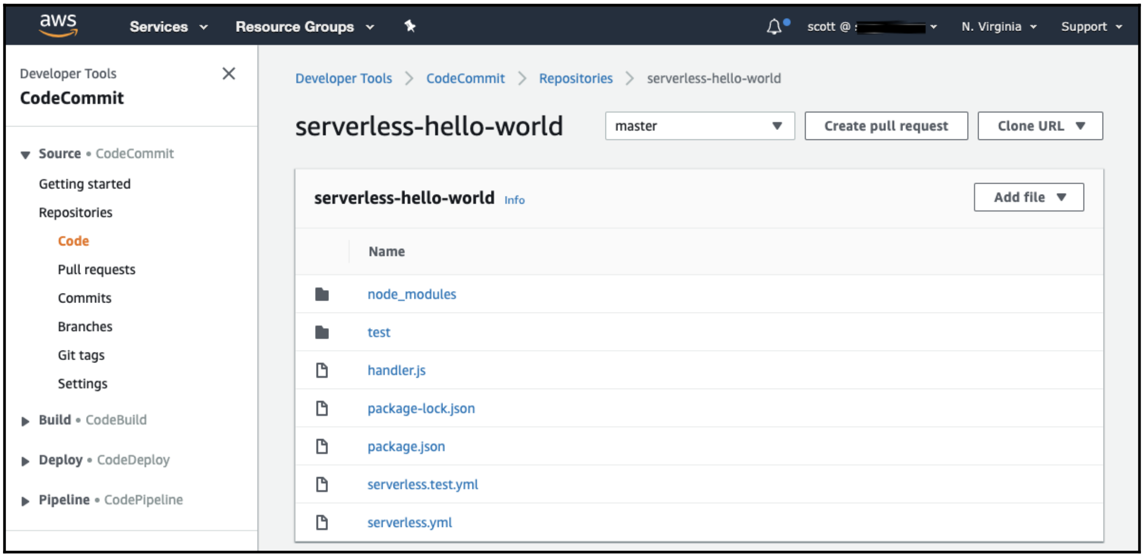



DNS Hosted Zone


90%
10%
DNS record using weighted routing



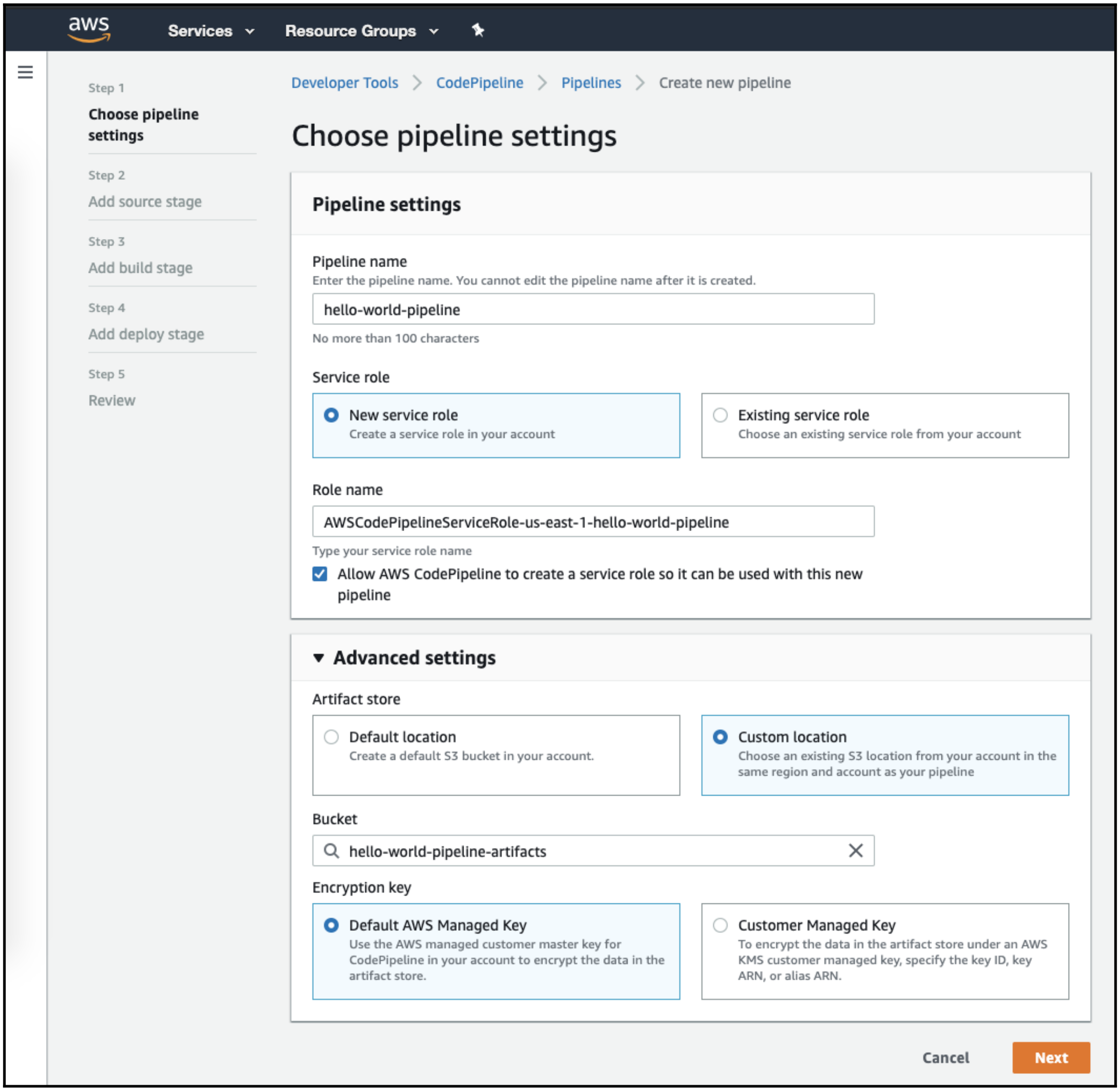




serverless-hello-world

serverless-hello-world

Name
node_modules
test
handler.js
package-lock.json
package.json
serverless.test.yml
serverless.yml



Choose pipeline settings

Pipeline settings

Pipeline name
Enter the pipeline name. You cannot edit the pipeline name after it is created.
hello-world-pipeline
No more than 100 characters

Service role

● New service role
Create a service role in your account

○ Existing service role
Choose an existing service role from your account

Role name
AWSCodePipelineServiceRole-us-east-1-hello-world-pipeline
Type your service role name

☑ Allow AWS CodePipeline to create a service role so it can be used with this new pipeline

▼ Advanced settings

Artifact store

○ Default location
Create a default S3 bucket in your account.

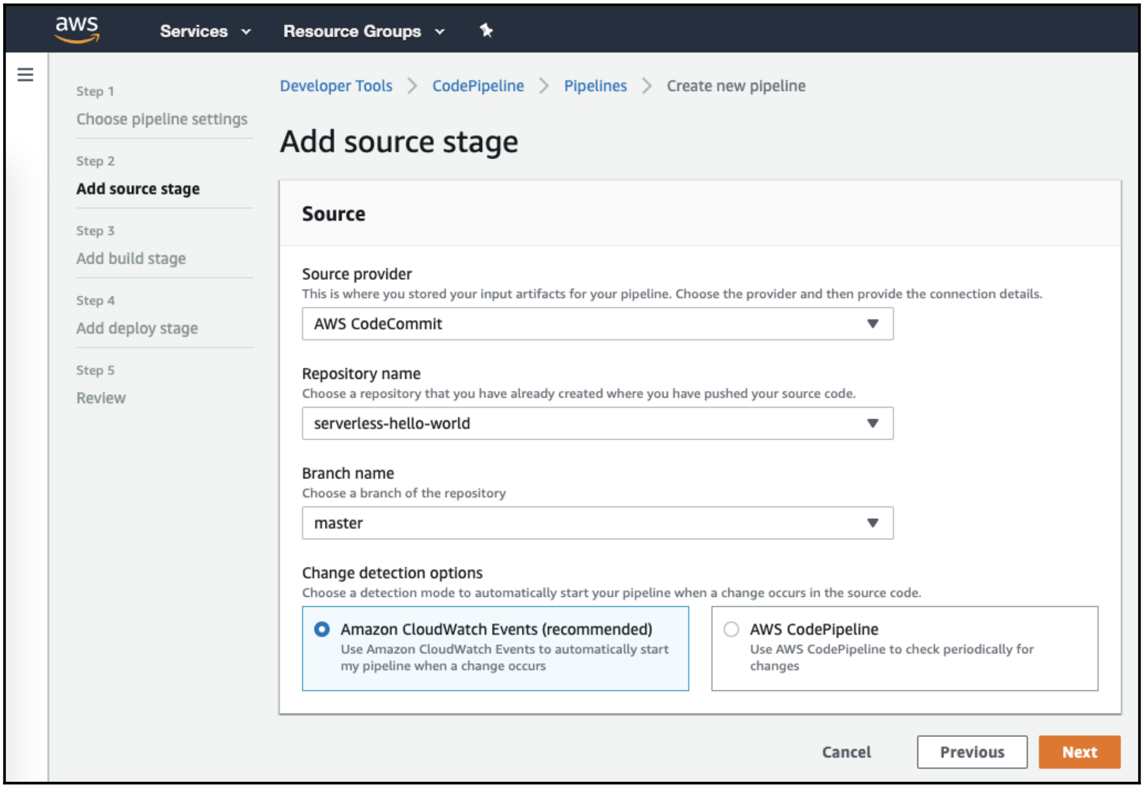
● Custom location
Choose an existing S3 location from your account in the same region and account as your pipeline

Bucket
hello-world-pipeline-artifacts

Encryption key

● Default AWS Managed Key
Use the AWS managed customer master key for CodePipeline in your account to encrypt the data in the artifact store.

○ Customer Managed Key
To encrypt the data in the artifact store under an AWS KMS customer managed key, specify the key ID, key ARN, or alias ARN.



Add source stage

Source

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

AWS CodeCommit

Repository name

Choose a repository that you have already created where you have pushed your source code.

serverless-hello-world

Branch name

Choose a branch of the repository

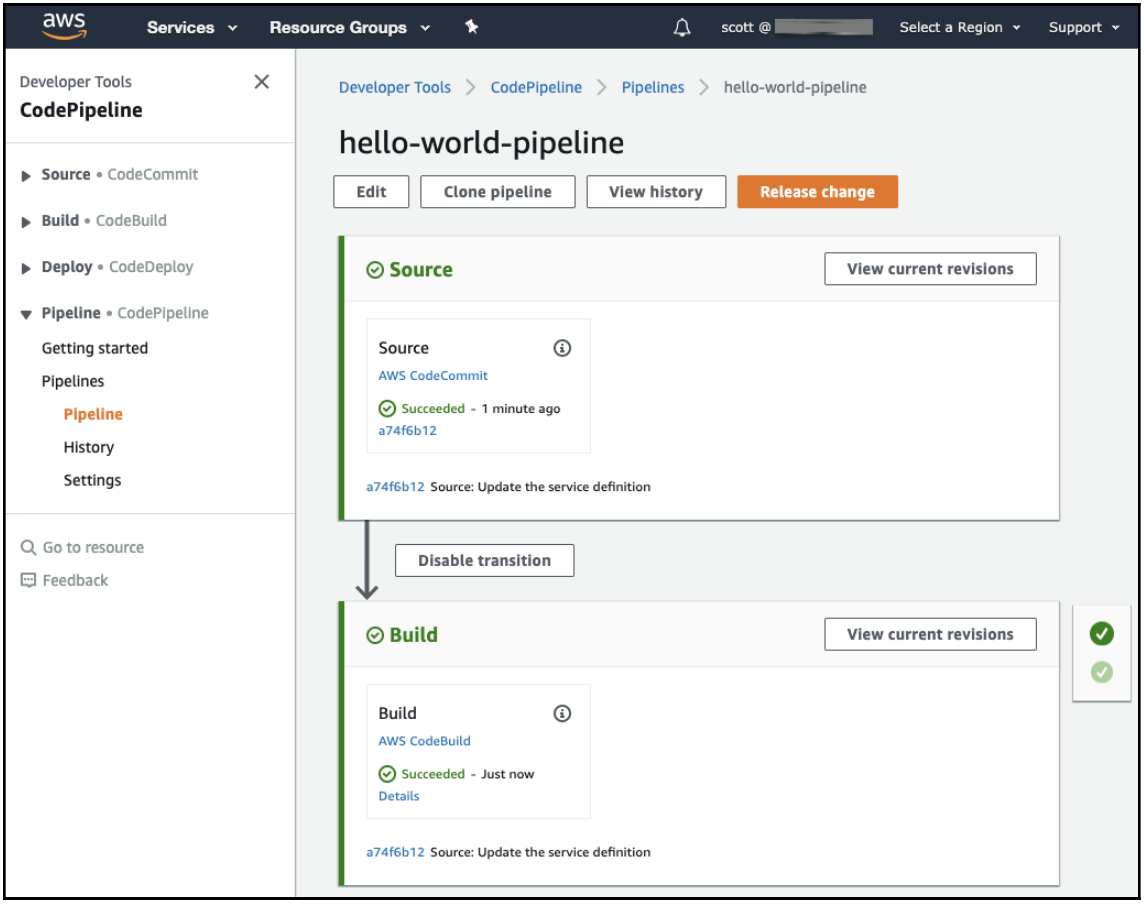
master

Change detection options

Choose a detection mode to automatically start your pipeline when a change occurs in the source code.

● Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

○ AWS CodePipeline
Use AWS CodePipeline to check periodically for changes



hello-world-pipeline

Source

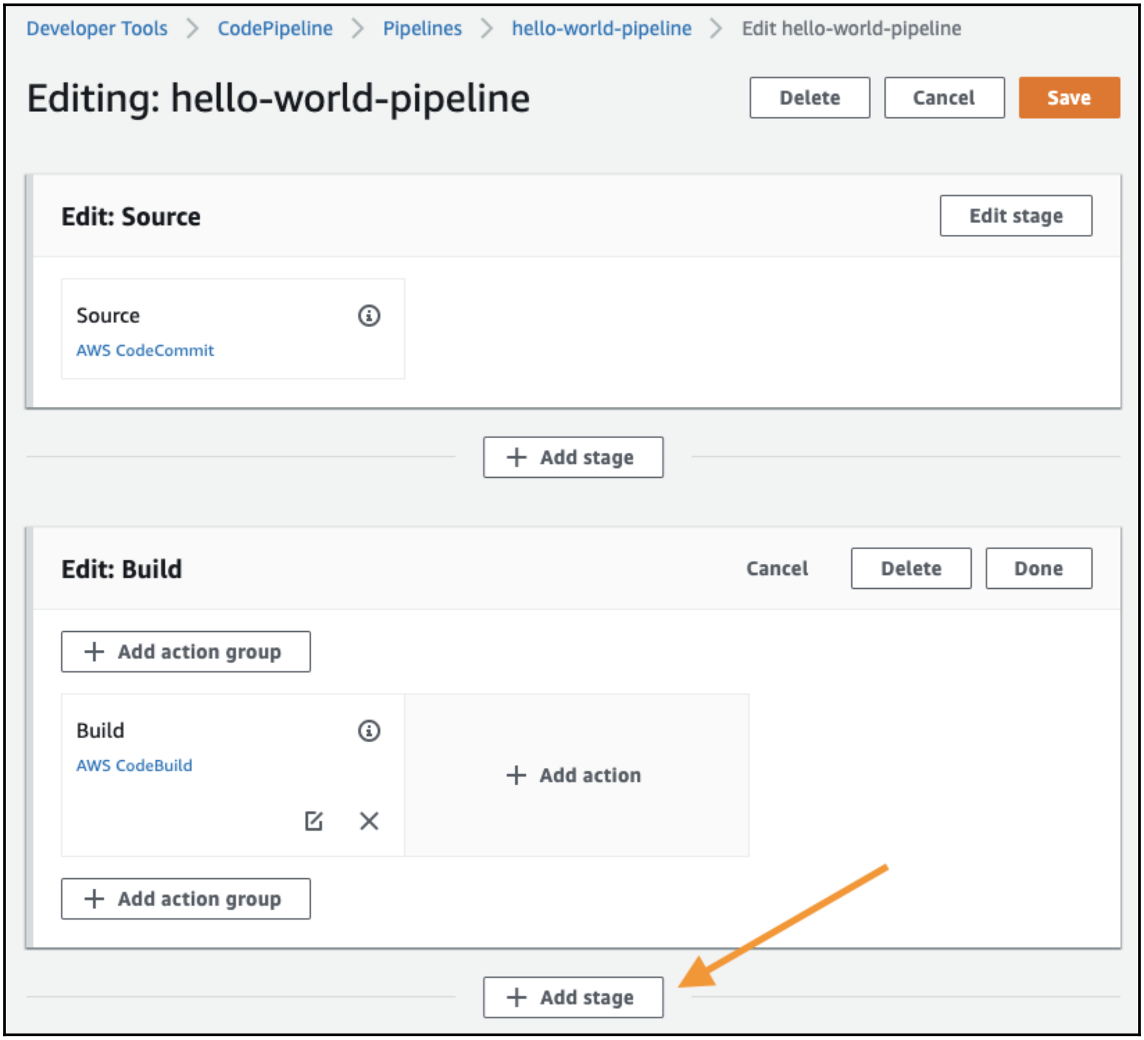
Source
AWS CodeCommit
Succeeded - 1 minute ago
a74f6b12

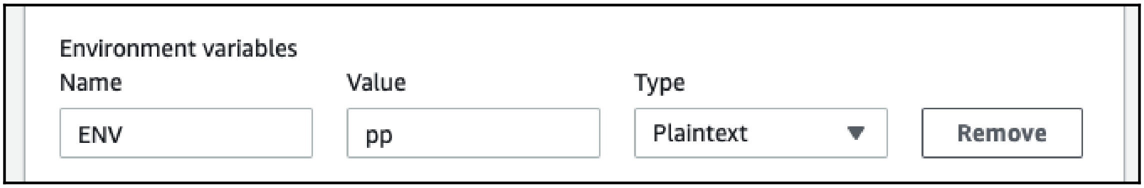
a74f6b12 Source: Update the service definition

Build

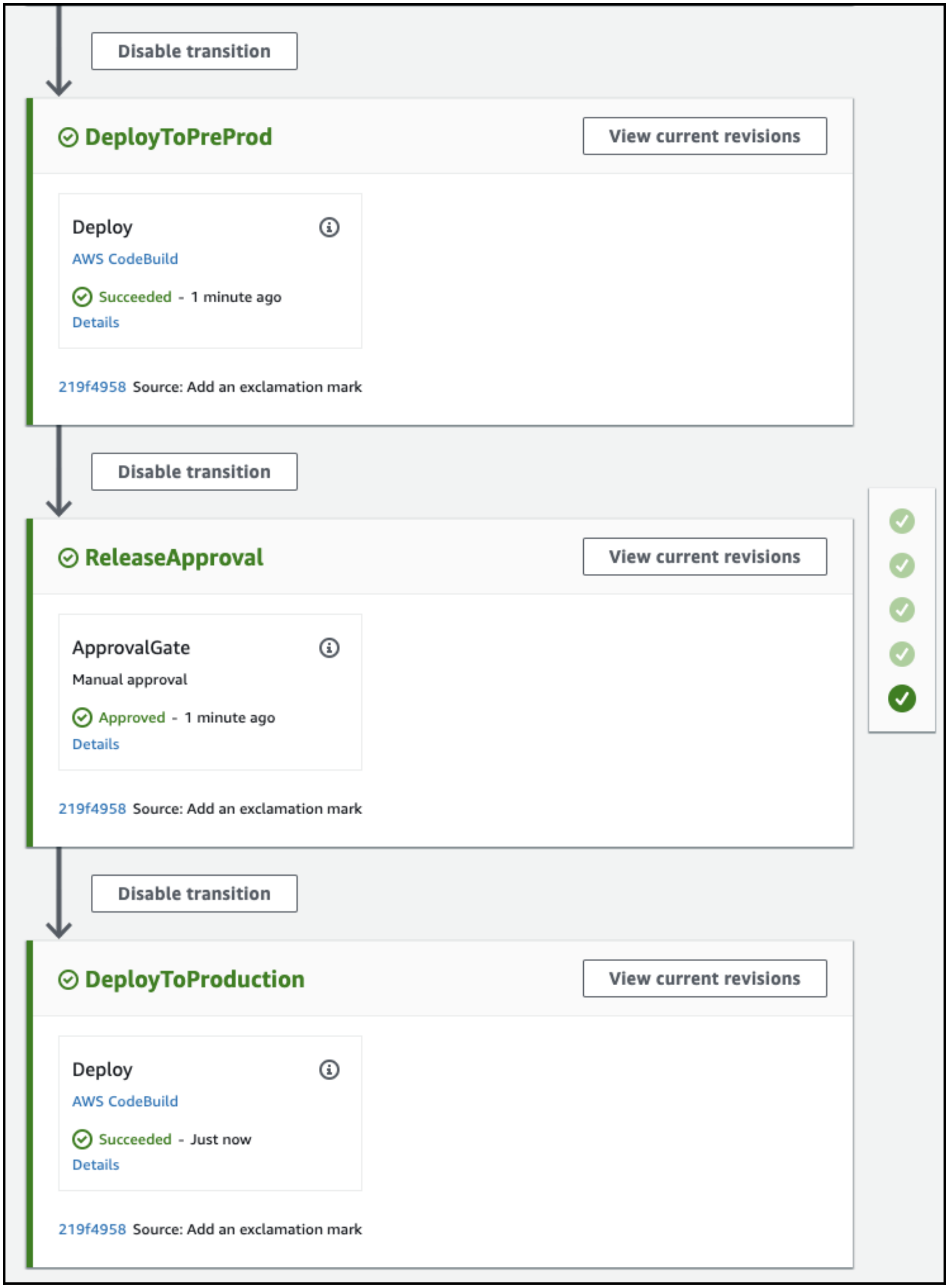
Build
AWS CodeBuild
Succeeded - Just now
Details

a74f6b12 Source: Update the service definition




Environment variables

Name	Value	Type	
ENV	pp	Plaintext	Remove



DeployToPreProd

Deploy
AWS CodeBuild
Succeeded - 1 minute ago
Details

219f4958 Source: Add an exclamation mark

ReleaseApproval

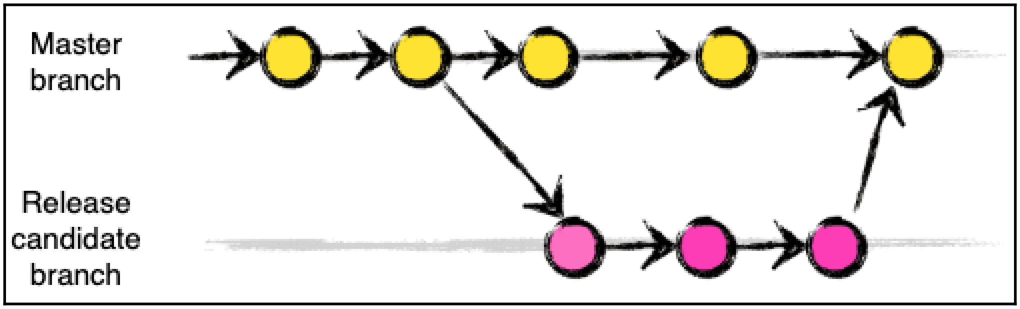
ApprovalGate
Manual approval
Approved - 1 minute ago
Details

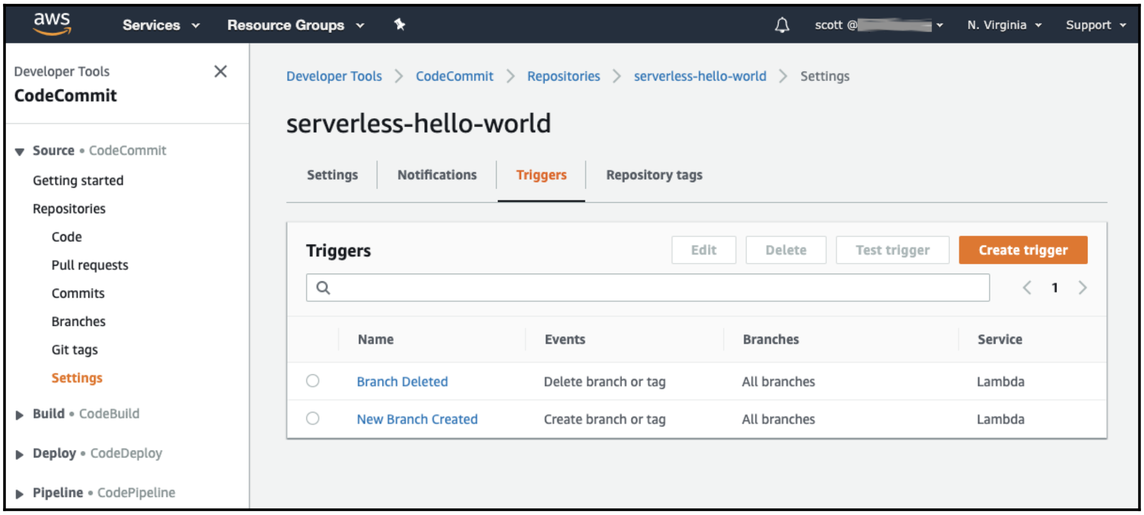
219f4958 Source: Add an exclamation mark

DeployToProduction

Deploy
AWS CodeBuild
Succeeded - Just now
Details

219f4958 Source: Add an exclamation mark





Developer Tools
CodeCommit

- ▼ **Source** • CodeCommit
  - Getting started
  - Repositories
    - Code
    - Pull requests
    - Commits
    - Branches
    - Git tags
    - Settings
- ▶ **Build** • CodeBuild
- ▶ **Deploy** • CodeDeploy
- ▶ **Pipeline** • CodePipeline

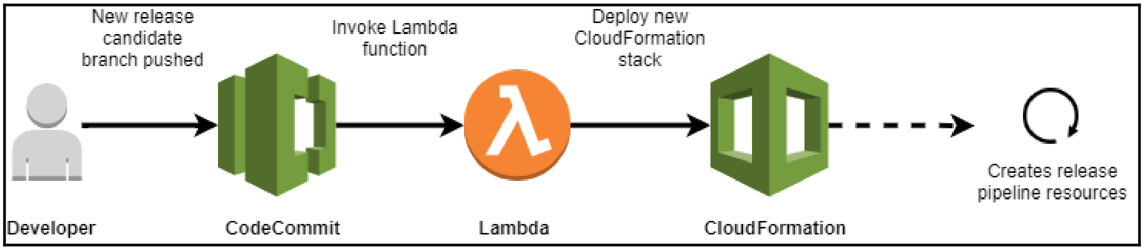
Developer Tools > CodeCommit > Repositories > serverless-hello-world > Settings

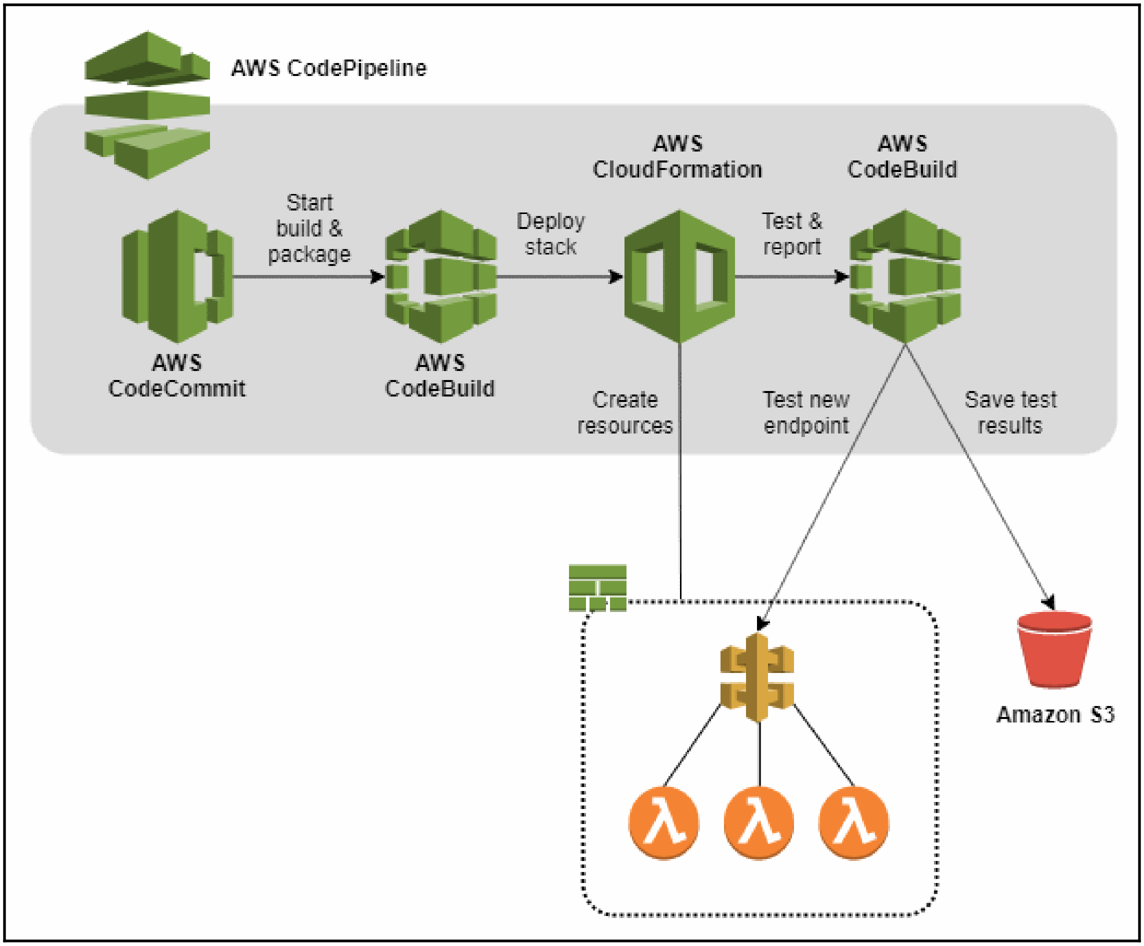
serverless-hello-world

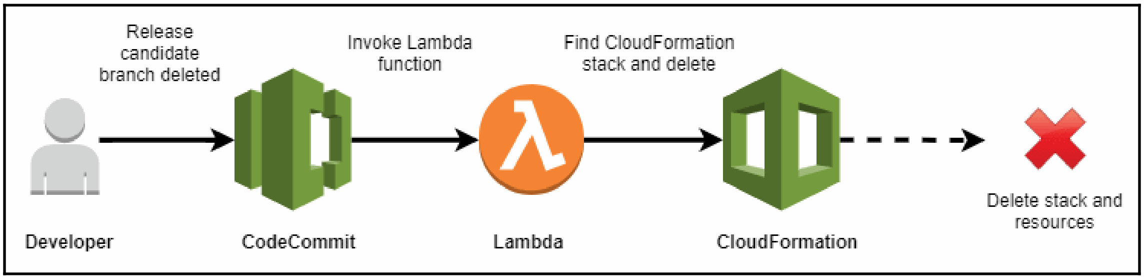
Settings | Notifications | Triggers | Repository tags

Triggers Edit | Delete | Test trigger | Create trigger

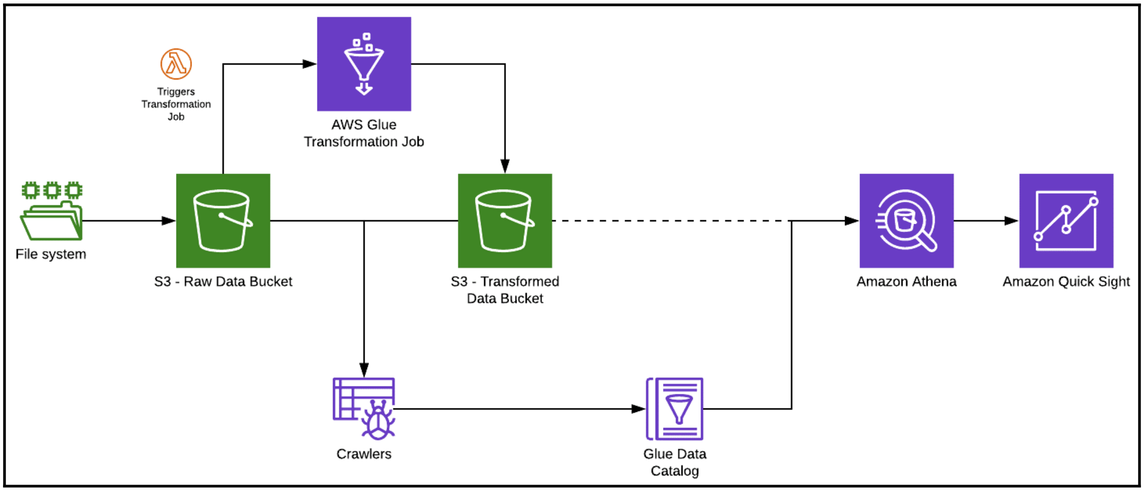
	Name	Events	Branches	Service
○	Branch Deleted	Delete branch or tag	All branches	Lambda
○	New Branch Created	Create branch or tag	All branches	Lambda

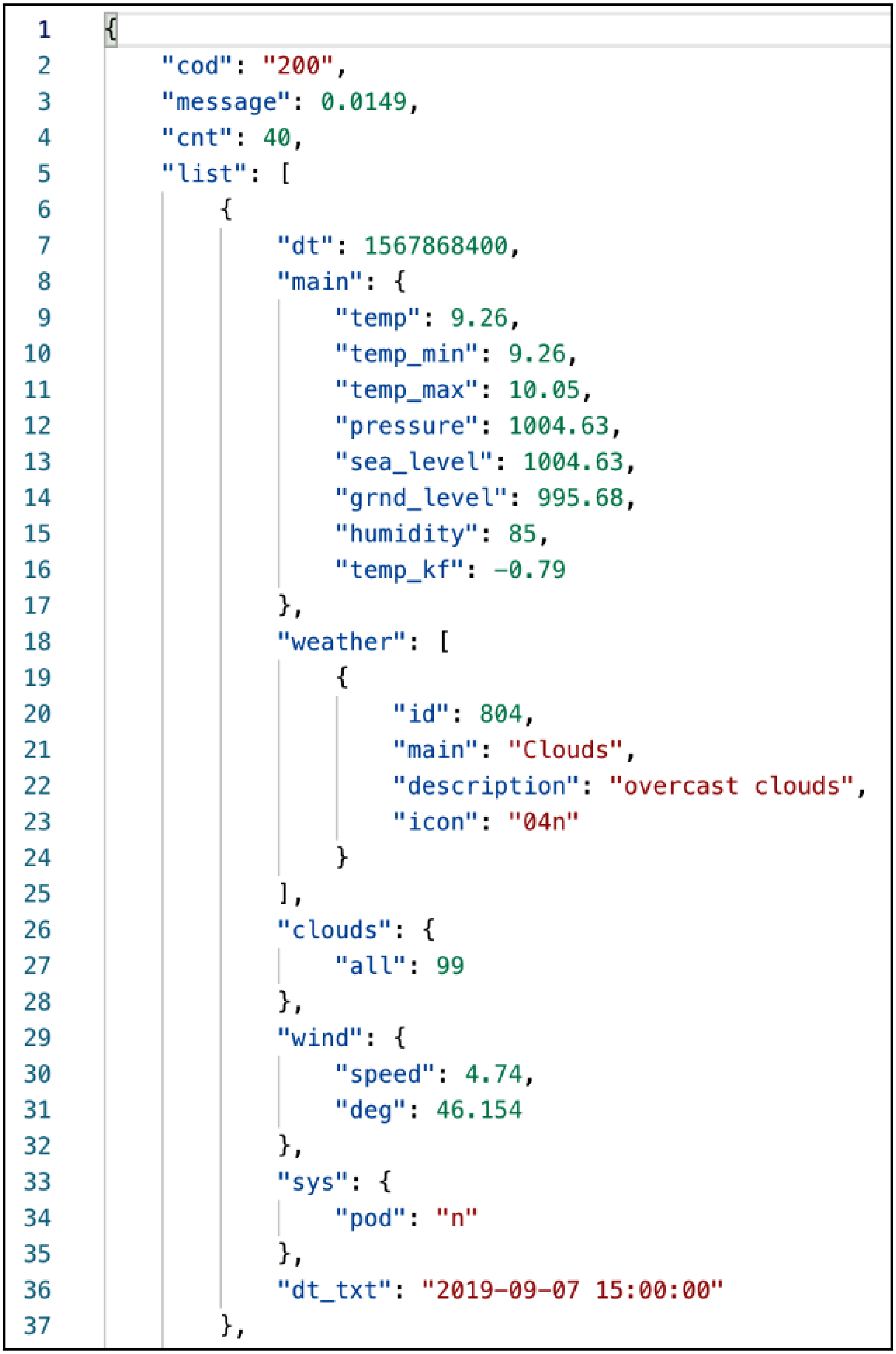

New release candidate branch pushed
Invoke Lambda function
Deploy new CloudFormation stack
Creates release pipeline resources
Developer
CodeCommit
Lambda
CloudFormation


AWS CodePipeline
Start build & package
Deploy stack
AWS CloudFormation
Test & report
AWS CodeBuild
AWS CodeCommit
AWS CodeBuild
Create resources
Test new endpoint
Save test results
Amazon S3

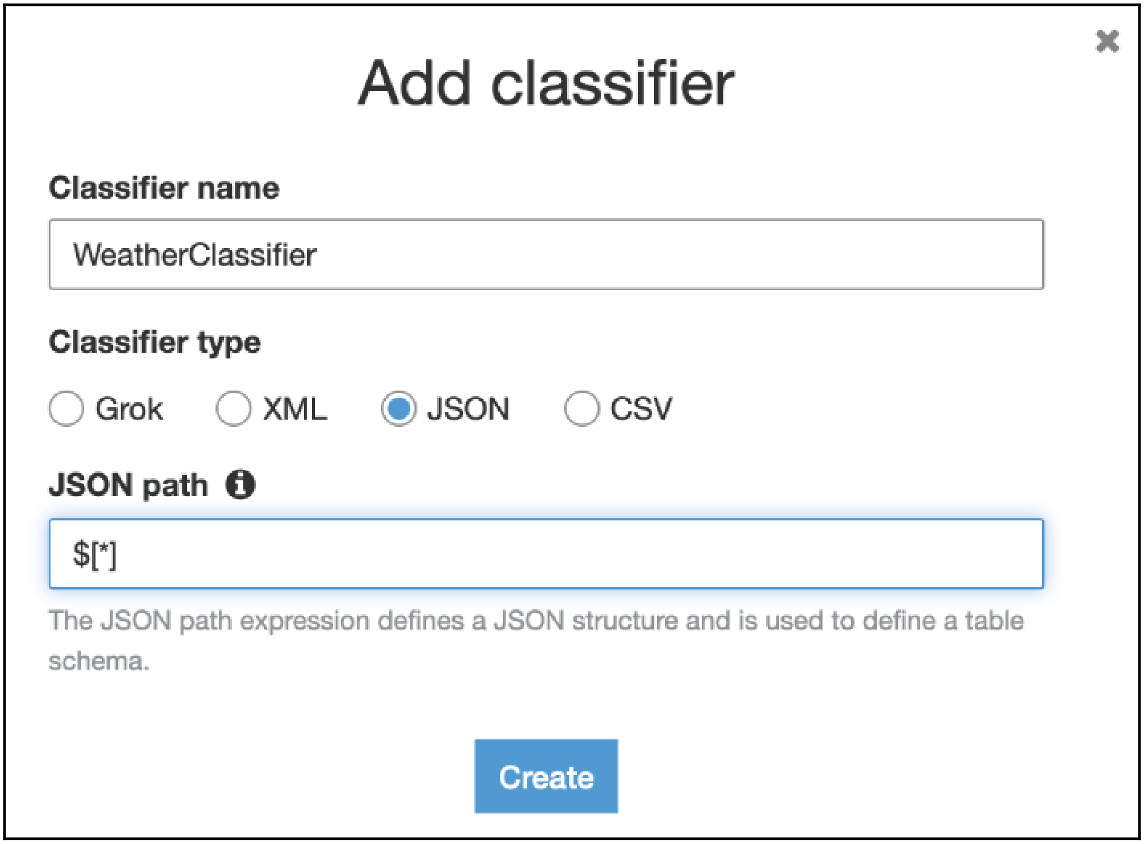

Release candidate branch deleted
Invoke Lambda function
Find CloudFormation stack and delete
Developer
CodeCommit
Lambda
CloudFormation
Delete stack and resources

Chapter 9: Data Processing





```
{
    "cod": "200",
    "message": 0.0149,
    "cnt": 40,
    "list": [
        {
            "dt": 1567868400,
            "main": {
                "temp": 9.26,
                "temp_min": 9.26,
                "temp_max": 10.05,
                "pressure": 1004.63,
                "sea_level": 1004.63,
                "grnd_level": 995.68,
                "humidity": 85,
                "temp_kf": -0.79
            },
            "weather": [
                {
                    "id": 804,
                    "main": "Clouds",
                    "description": "overcast clouds",
                    "icon": "04n"
                }
            ],
            "clouds": {
                "all": 99
            },
            "wind": {
                "speed": 4.74,
                "deg": 46.154
            },
            "sys": {
                "pod": "n"
            },
            "dt_txt": "2019-09-07 15:00:00"
        },
```



Add classifier

Classifier name

WeatherClassifier

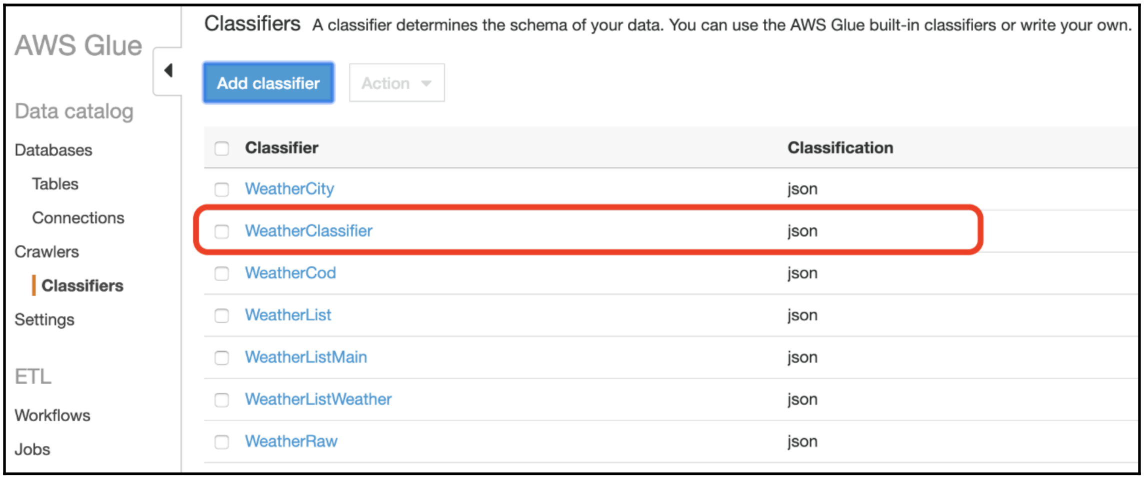
Classifier type

○ Grok ○ XML ● JSON ○ CSV

JSON path

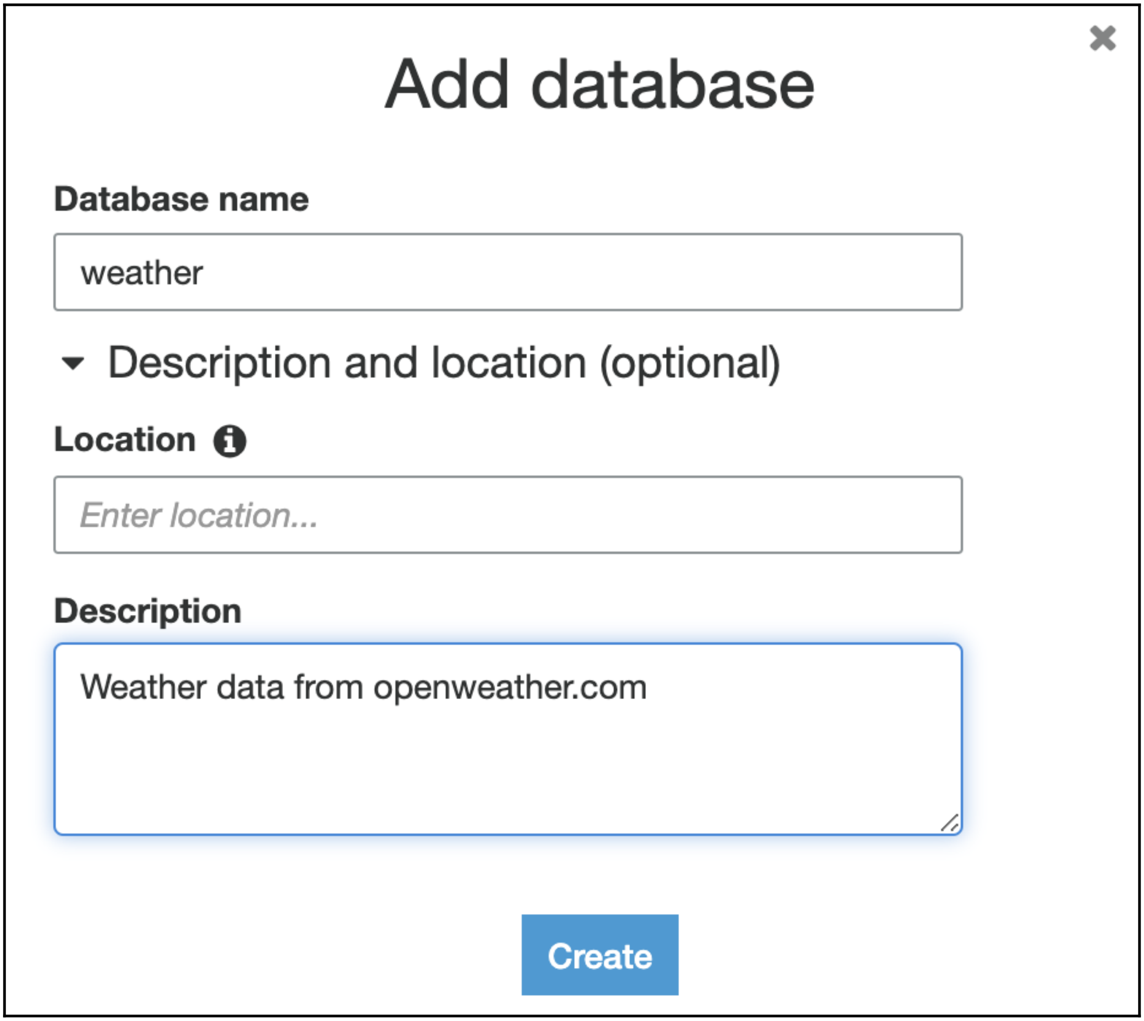
$[*]

The JSON path expression defines a JSON structure and is used to define a table schema.



Classifiers A classifier determines the schema of your data. You can use the AWS Glue built-in classifiers or write your own.

□	Classifier	Classification
□	WeatherCity	json
□	WeatherClassifier	json
□	WeatherCod	json
□	WeatherList	json
□	WeatherListMain	json
□	WeatherListWeather	json
□	WeatherRaw	json



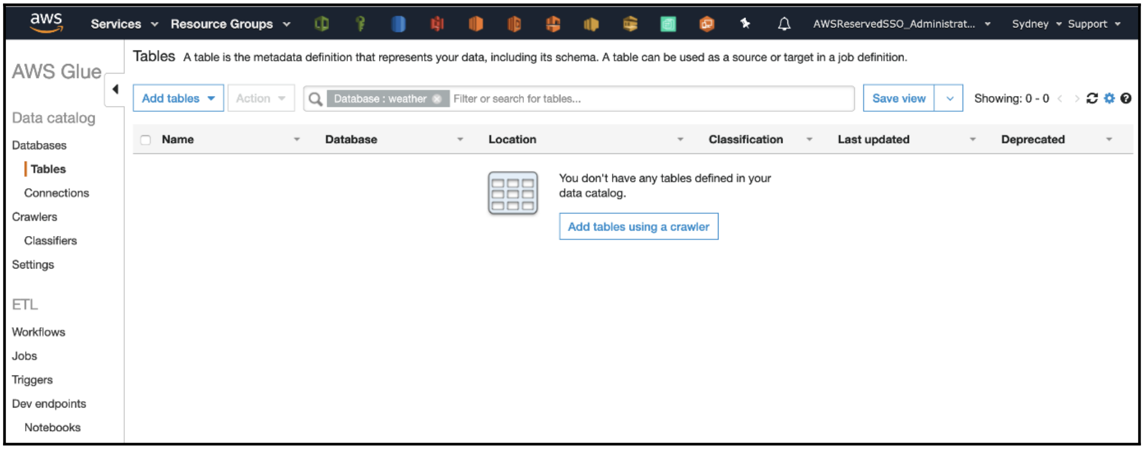
Add database

Database name: weather

▾ Description and location (optional)

Location: Enter location...

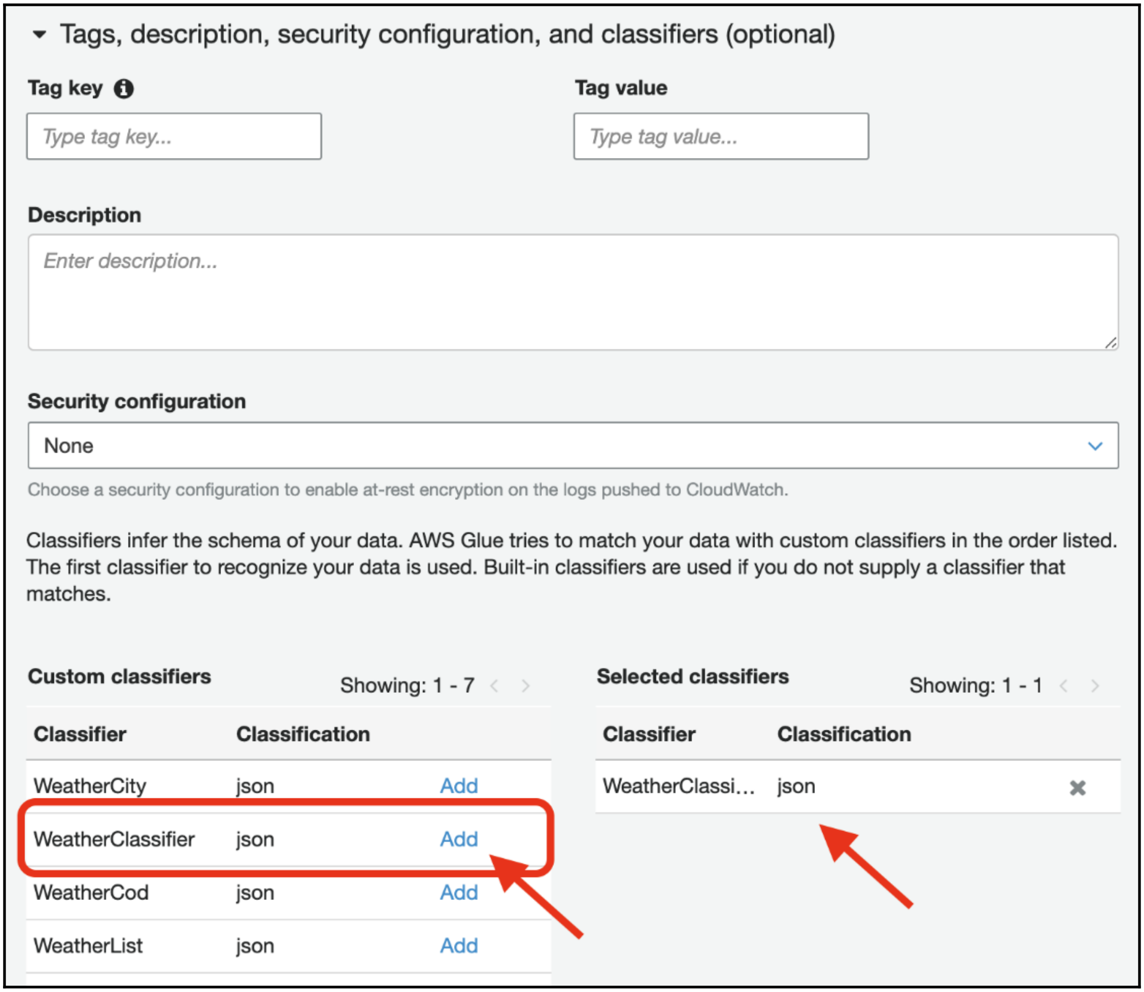
Description: Weather data from openweather.com



Tables A table is the metadata definition that represents your data, including its schema. A table can be used as a source or target in a job definition.

□	Name	Database	Location	Classification	Last updated	Deprecated

You don't have any tables defined in your data catalog.



▾ Tags, description, security configuration, and classifiers (optional)

Tag key	Tag value
Type tag key...	Type tag value...

Description

Enter description...

Security configuration

None

Choose a security configuration to enable at-rest encryption on the logs pushed to CloudWatch.

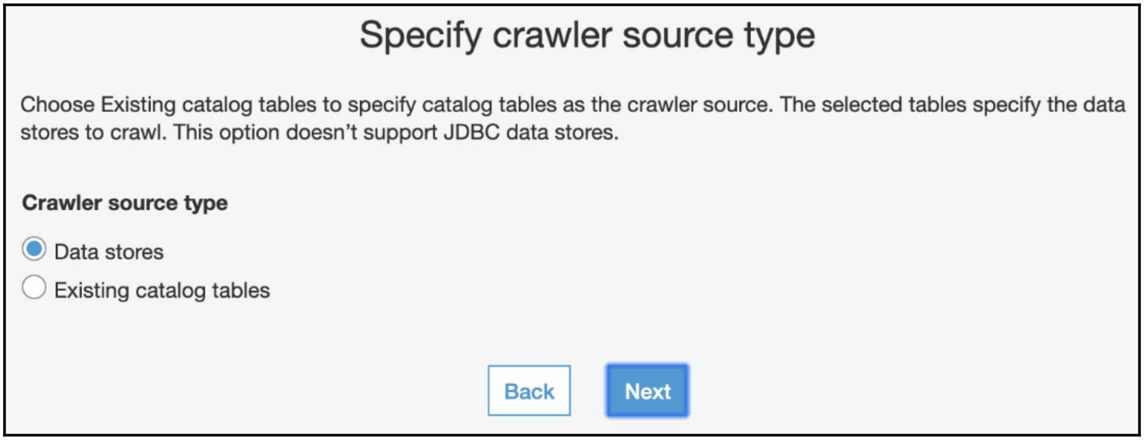
Classifiers infer the schema of your data. AWS Glue tries to match your data with custom classifiers in the order listed. The first classifier to recognize your data is used. Built-in classifiers are used if you do not supply a classifier that matches.

Custom classifiers Showing: 1 - 7

Classifier	Classification	
WeatherCity	json	Add
WeatherClassifier	json	Add
WeatherCod	json	Add
WeatherList	json	Add

Selected classifiers Showing: 1 - 1

Classifier	Classification	
WeatherClassi...	json	✖

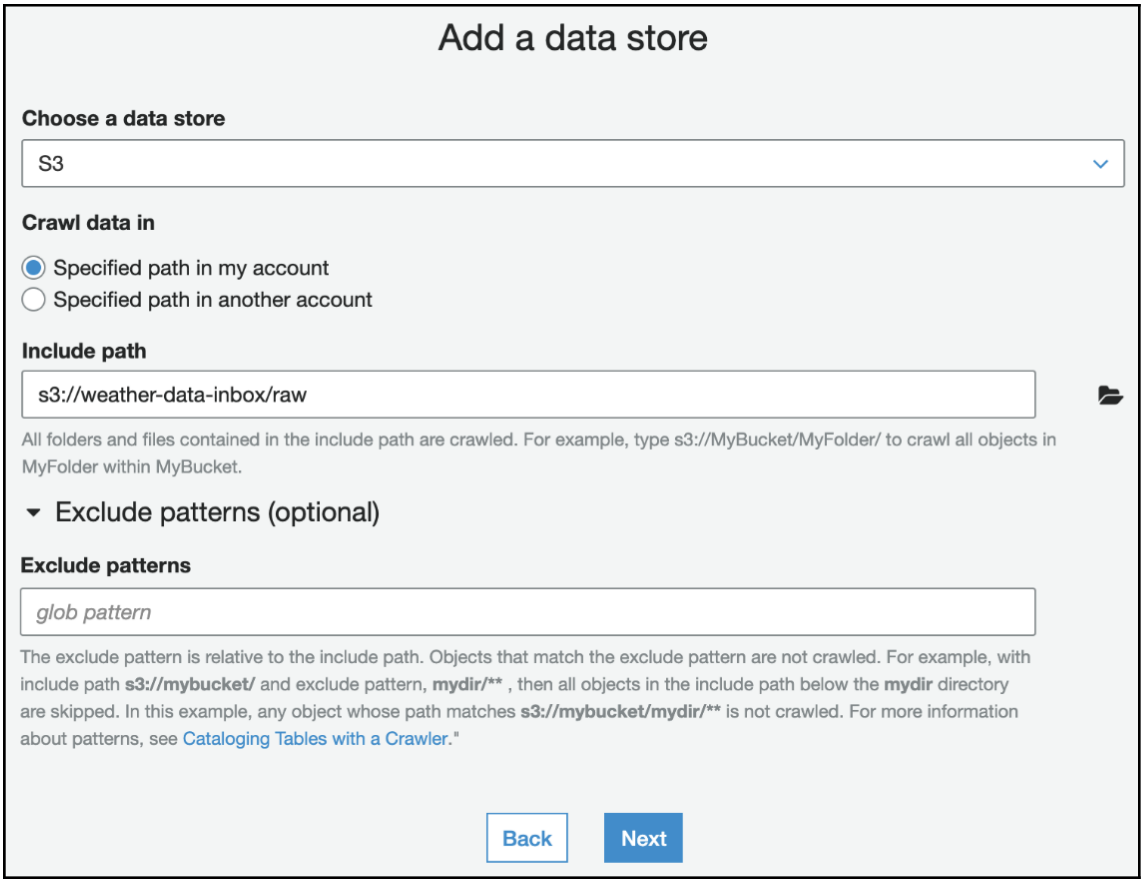


Specify crawler source type

Choose Existing catalog tables to specify catalog tables as the crawler source. The selected tables specify the data stores to crawl. This option doesn't support JDBC data stores.

Crawler source type

● Data stores
○ Existing catalog tables



Add a data store

Choose a data store

S3

Crawl data in

● Specified path in my account
○ Specified path in another account

Include path

s3://weather-data-inbox/raw

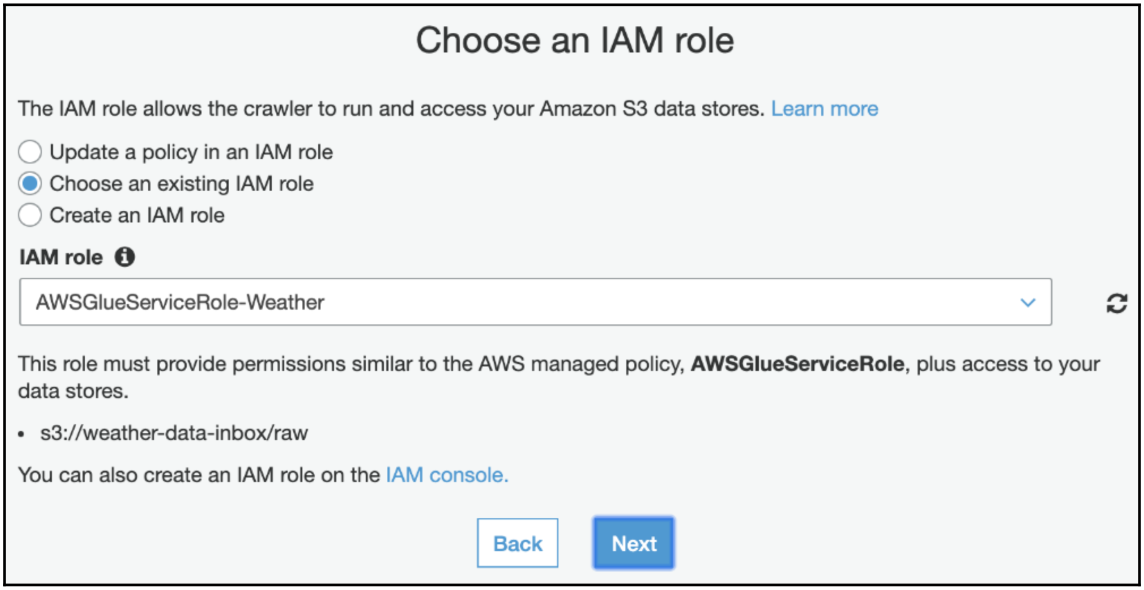
All folders and files contained in the include path are crawled. For example, type s3://MyBucket/MyFolder/ to crawl all objects in MyFolder within MyBucket.

▾ Exclude patterns (optional)

Exclude patterns

glob pattern

The exclude pattern is relative to the include path. Objects that match the exclude pattern are not crawled. For example, with include path **s3://mybucket/** and exclude pattern, **mydir/**** , then all objects in the include path below the **mydir** directory are skipped. In this example, any object whose path matches **s3://mybucket/mydir/**** is not crawled. For more information about patterns, see Cataloging Tables with a Crawler."



Choose an IAM role

The IAM role allows the crawler to run and access your Amazon S3 data stores. Learn more

○ Update a policy in an IAM role
● Choose an existing IAM role
○ Create an IAM role

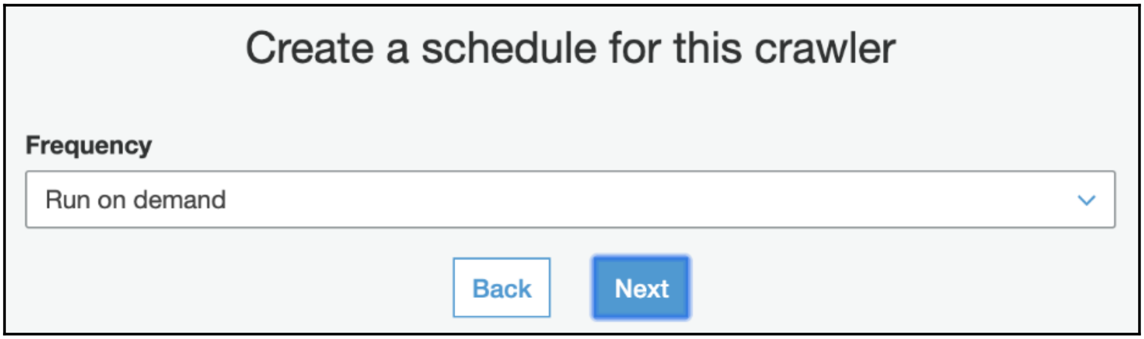
IAM role

AWSGlueServiceRole-Weather

This role must provide permissions similar to the AWS managed policy, **AWSGlueServiceRole**, plus access to your data stores.

- s3://weather-data-inbox/raw

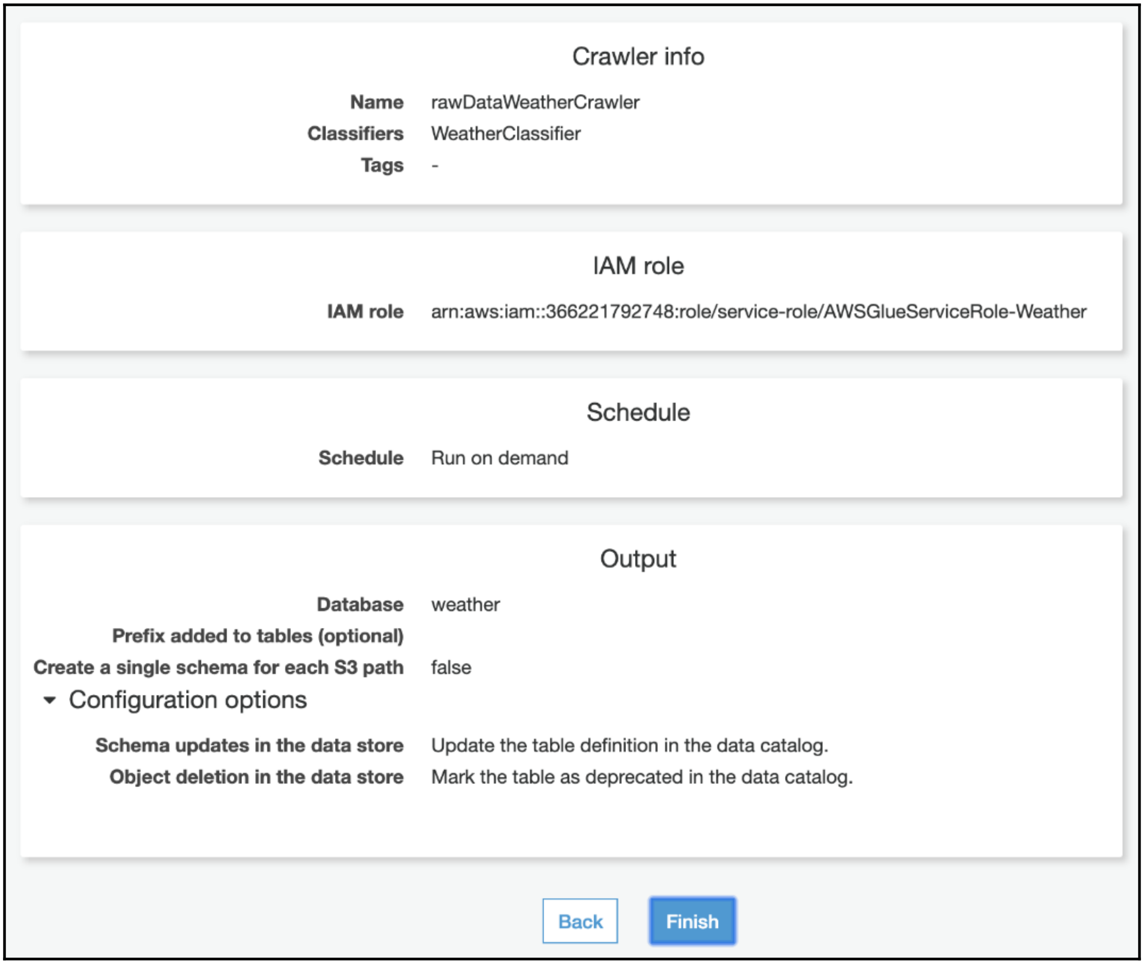
You can also create an IAM role on the IAM console.



Create a schedule for this crawler

Frequency

Run on demand



Crawler info

Name	rawDataWeatherCrawler
Classifiers	WeatherClassifier
Tags	-

IAM role

IAM role	arn:aws:iam::366221792748:role/service-role/AWSGlueServiceRole-Weather

Schedule

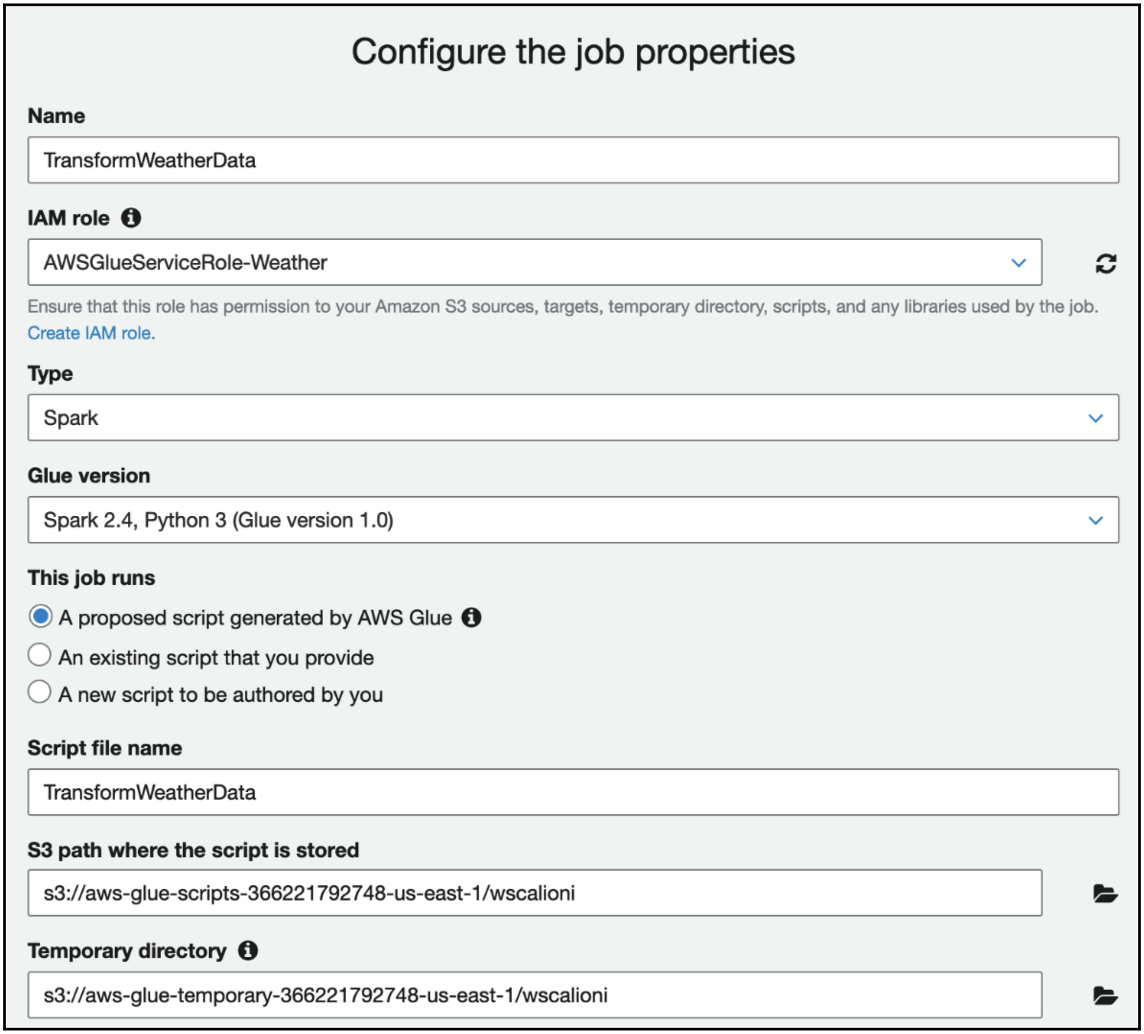
Schedule	Run on demand

Output

Database	weather
Prefix added to tables (optional)	
Create a single schema for each S3 path	false

▾ Configuration options

Schema updates in the data store	Update the table definition in the data catalog.
Object deletion in the data store	Mark the table as deprecated in the data catalog.



Configure the job properties

Name

TransformWeatherData

IAM role ℹ

AWSGlueServiceRole-Weather

Ensure that this role has permission to your Amazon S3 sources, targets, temporary directory, scripts, and any libraries used by the job. Create IAM role.

Type

Spark

Glue version

Spark 2.4, Python 3 (Glue version 1.0)

This job runs

● A proposed script generated by AWS Glue ℹ

○ An existing script that you provide

○ A new script to be authored by you

Script file name

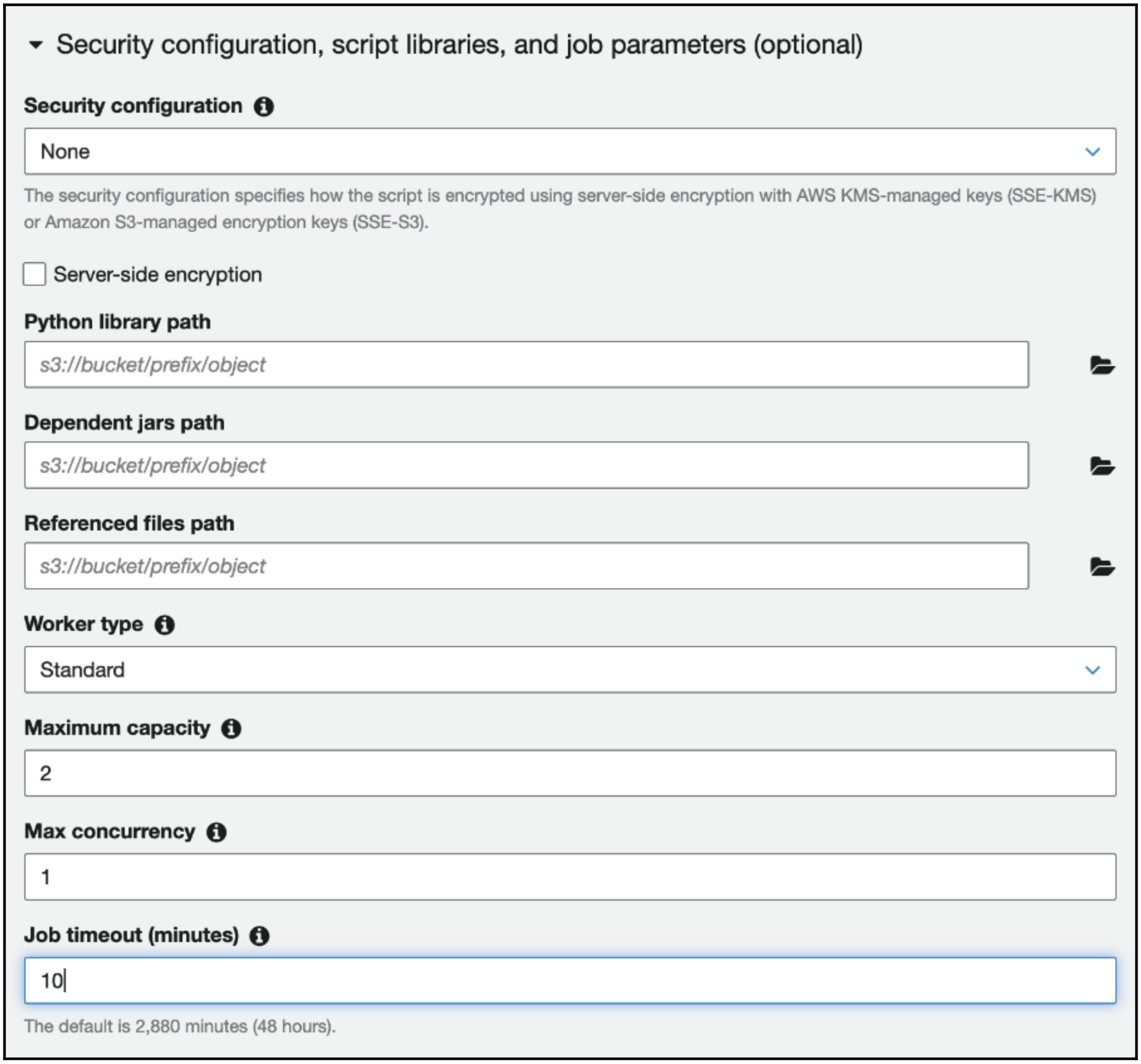
TransformWeatherData

S3 path where the script is stored

s3://aws-glue-scripts-366221792748-us-east-1/wscalioni

Temporary directory ℹ

s3://aws-glue-temporary-366221792748-us-east-1/wscalioni



▾ Security configuration, script libraries, and job parameters (optional)

Security configuration ℹ

None

The security configuration specifies how the script is encrypted using server-side encryption with AWS KMS-managed keys (SSE-KMS) or Amazon S3-managed encryption keys (SSE-S3).

□ Server-side encryption

Python library path

s3://bucket/prefix/object

Dependent jars path

s3://bucket/prefix/object

Referenced files path

s3://bucket/prefix/object

Worker type ℹ

Standard

Maximum capacity ℹ

2

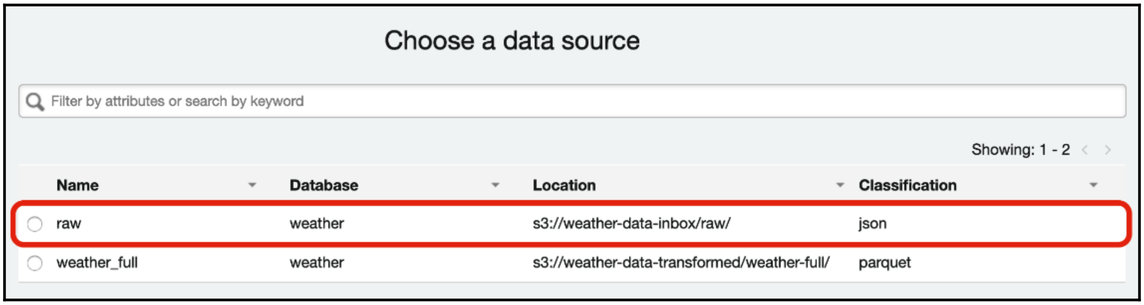
Max concurrency ℹ

1

Job timeout (minutes) ℹ

10

The default is 2,880 minutes (48 hours).

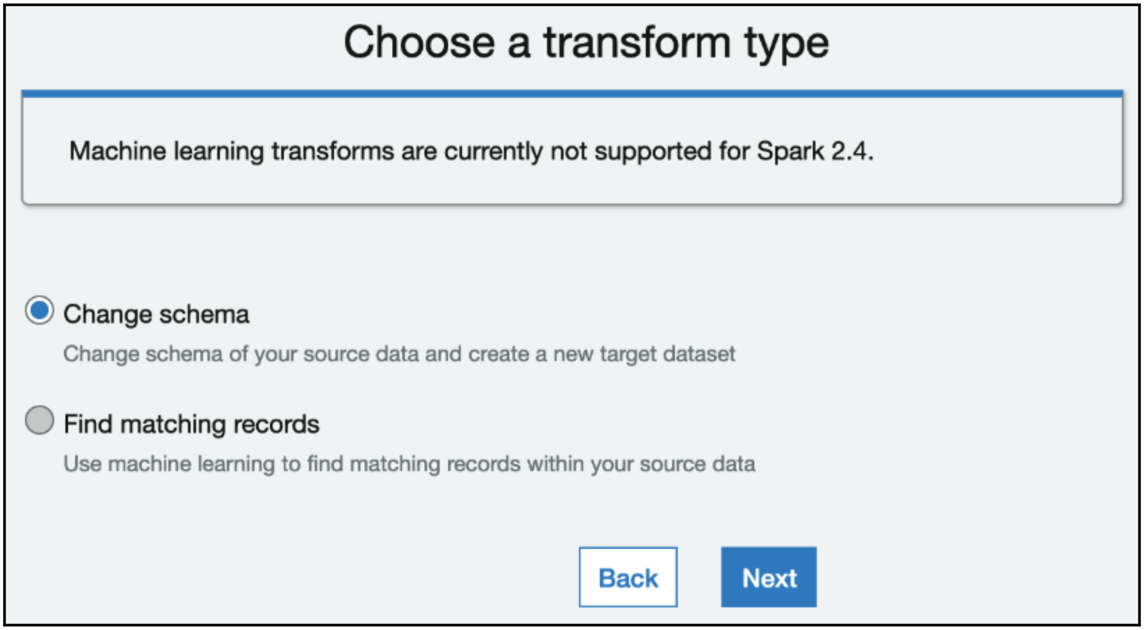


Choose a data source

Filter by attributes or search by keyword

Showing: 1 - 2

	Name	Database	Location	Classification
○	raw	weather	s3://weather-data-inbox/raw/	json
○	weather_full	weather	s3://weather-data-transformed/weather-full/	parquet



Choose a transform type

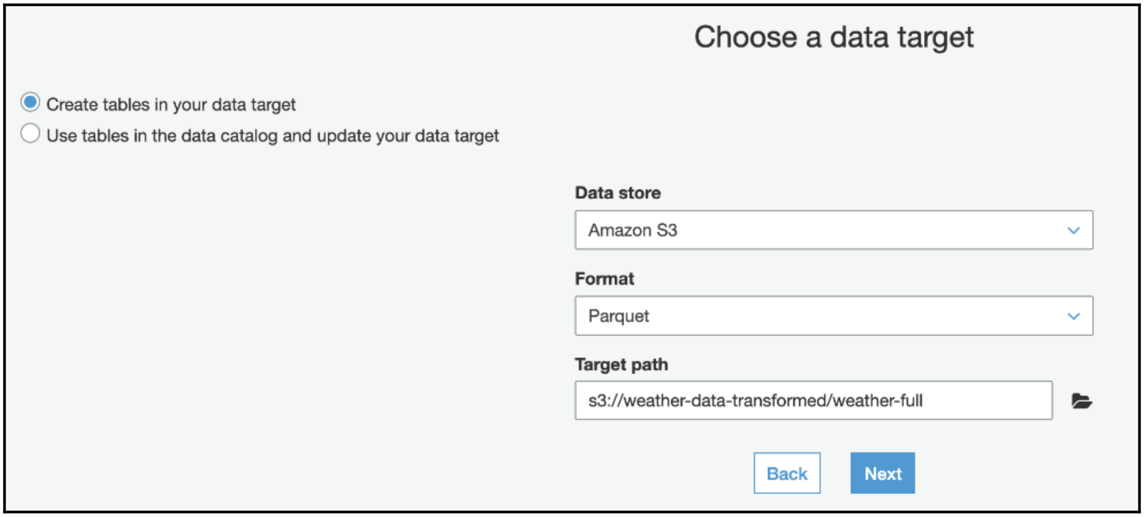
Machine learning transforms are currently not supported for Spark 2.4.

● Change schema

Change schema of your source data and create a new target dataset

○ Find matching records

Use machine learning to find matching records within your source data



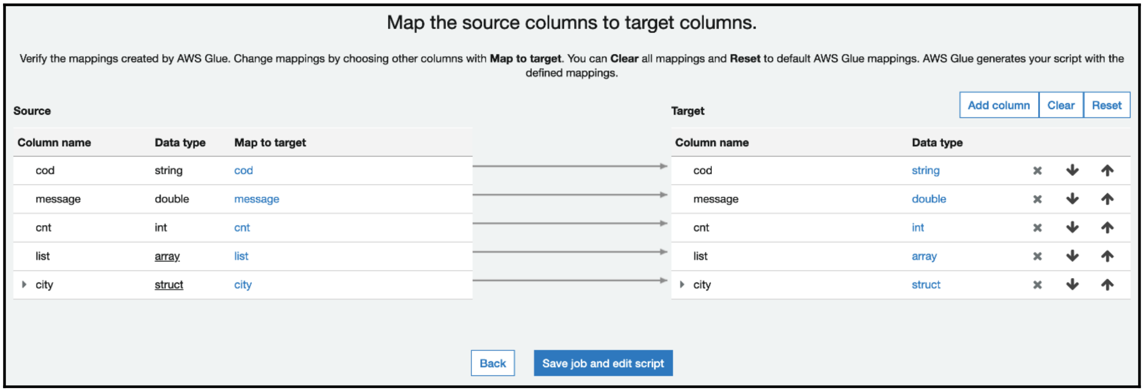
Choose a data target

● Create tables in your data target
○ Use tables in the data catalog and update your data target

Data store: Amazon S3

Format: Parquet

Target path: s3://weather-data-transformed/weather-full



Map the source columns to target columns.

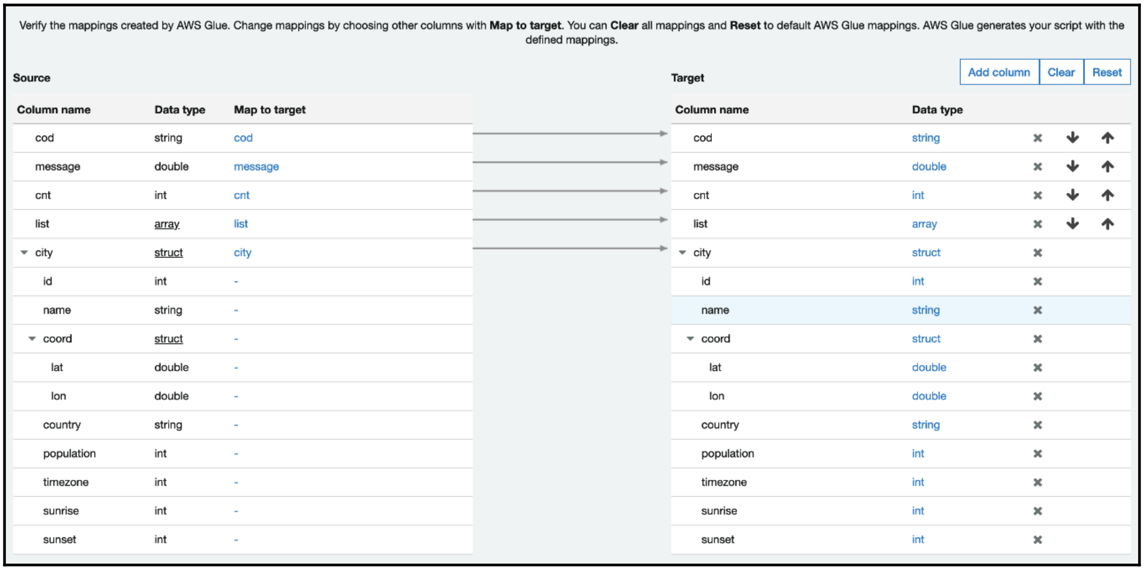
Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

Source

Column name	Data type	Map to target
cod	string	cod
message	double	message
cnt	int	cnt
list	array	list
▸ city	struct	city

Target

Column name	Data type			
cod	string	✖	↓	↑
message	double	✖	↓	↑
cnt	int	✖	↓	↑
list	array	✖	↓	↑
▸ city	struct	✖	↓	↑



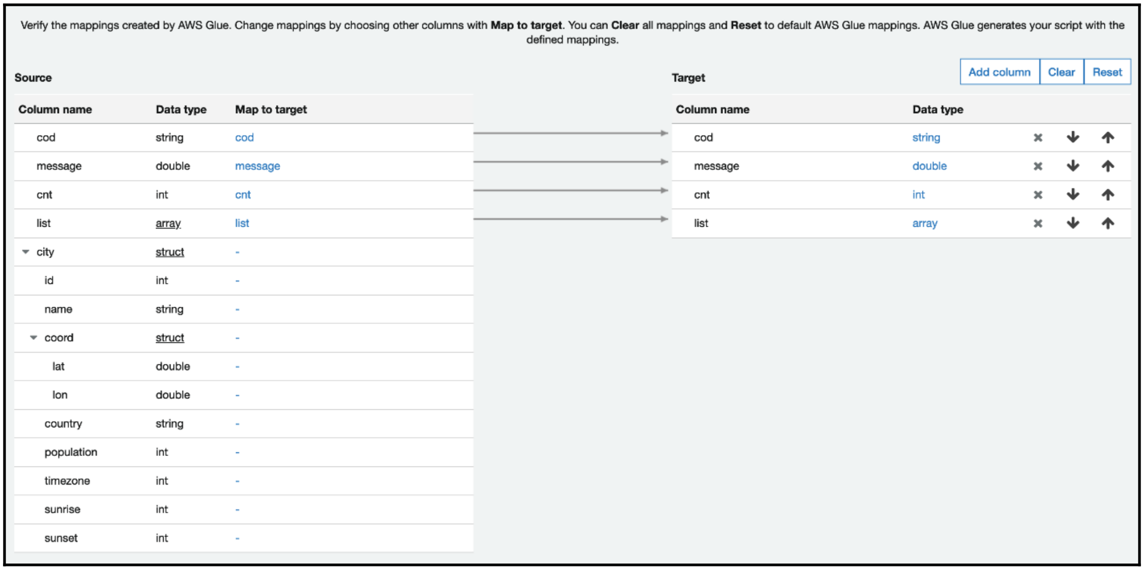
Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

Source

Column name	Data type	Map to target
cod	string	cod
message	double	message
cnt	int	cnt
list	array	list
▾ city	struct	city
id	int	-
name	string	-
▾ coord	struct	-
lat	double	-
lon	double	-
country	string	-
population	int	-
timezone	int	-
sunrise	int	-
sunset	int	-

Target

Column name	Data type			
cod	string	✖	↓	↑
message	double	✖	↓	↑
cnt	int	✖	↓	↑
list	array	✖	↓	↑
▾ city	struct	✖		
id	int	✖		
name	string	✖		
▾ coord	struct	✖		
lat	double	✖		
lon	double	✖		
country	string	✖		
population	int	✖		
timezone	int	✖		
sunrise	int	✖		
sunset	int	✖		



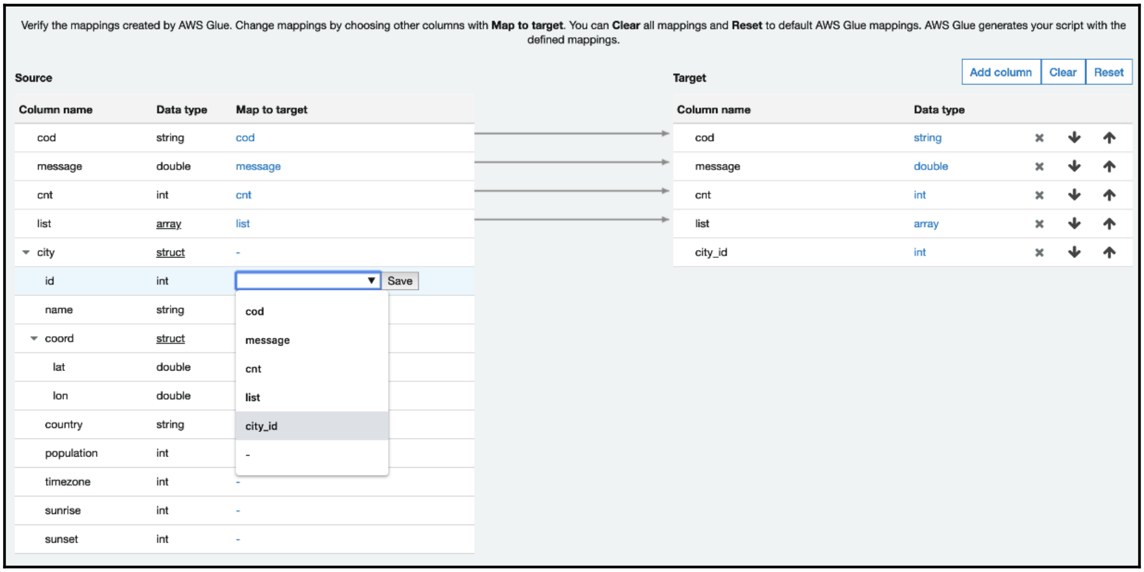
Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

Source

Column name	Data type	Map to target
cod	string	cod
message	double	message
cnt	int	cnt
list	array	list
city	struct	-
id	int	-
name	string	-
coord	struct	-
lat	double	-
lon	double	-
country	string	-
population	int	-
timezone	int	-
sunrise	int	-
sunset	int	-

Target

Column name	Data type			
cod	string	✖	↓	↑
message	double	✖	↓	↑
cnt	int	✖	↓	↑
list	array	✖	↓	↑



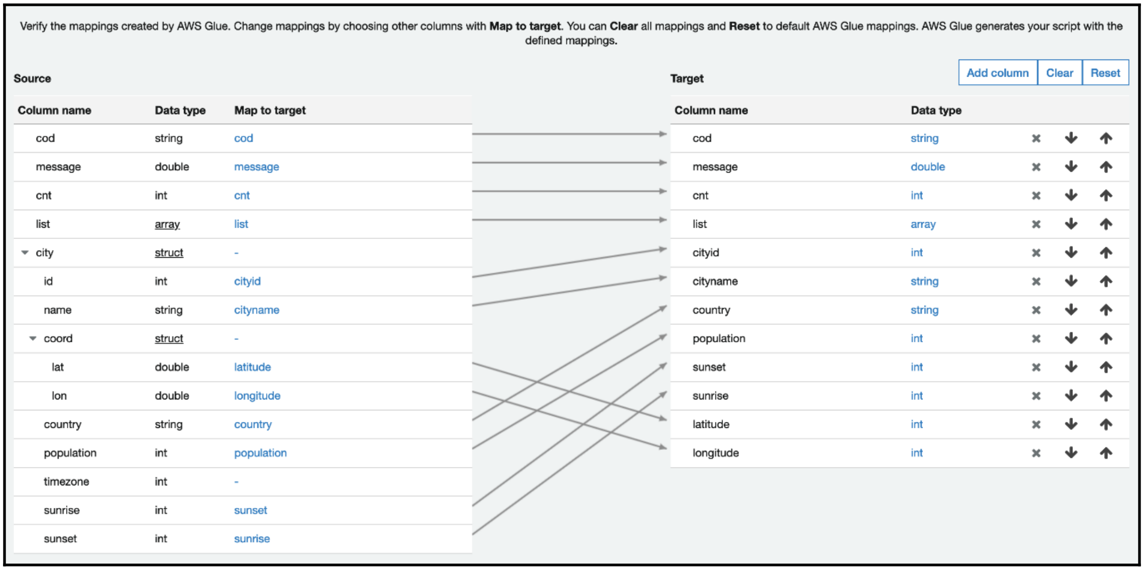
Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

Source

Column name	Data type	Map to target
cod	string	cod
message	double	message
cnt	int	cnt
list	array	list
city	struct	-
id	int	
name	string	
coord	struct	
lat	double	
lon	double	
country	string	
population	int	
timezone	int	-
sunrise	int	-
sunset	int	-

Target

Column name	Data type			
cod	string	✖	↓	↑
message	double	✖	↓	↑
cnt	int	✖	↓	↑
list	array	✖	↓	↑
city_id	int	✖	↓	↑



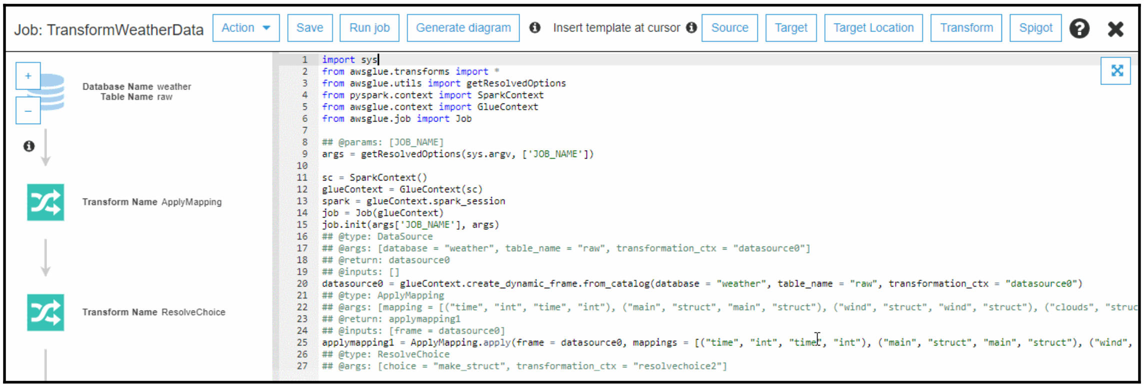
Verify the mappings created by AWS Glue. Change mappings by choosing other columns with **Map to target**. You can **Clear** all mappings and **Reset** to default AWS Glue mappings. AWS Glue generates your script with the defined mappings.

Source

Column name	Data type	Map to target
cod	string	cod
message	double	message
cnt	int	cnt
list	array	list
▾ city	struct	-
id	int	cityid
name	string	cityname
▾ coord	struct	-
lat	double	latitude
lon	double	longitude
country	string	country
population	int	population
timezone	int	-
sunrise	int	sunset
sunset	int	sunrise

Target

Column name	Data type
cod	string
message	double
cnt	int
list	array
cityid	int
cityname	string
country	string
population	int
sunset	int
sunrise	int
latitude	int
longitude	int

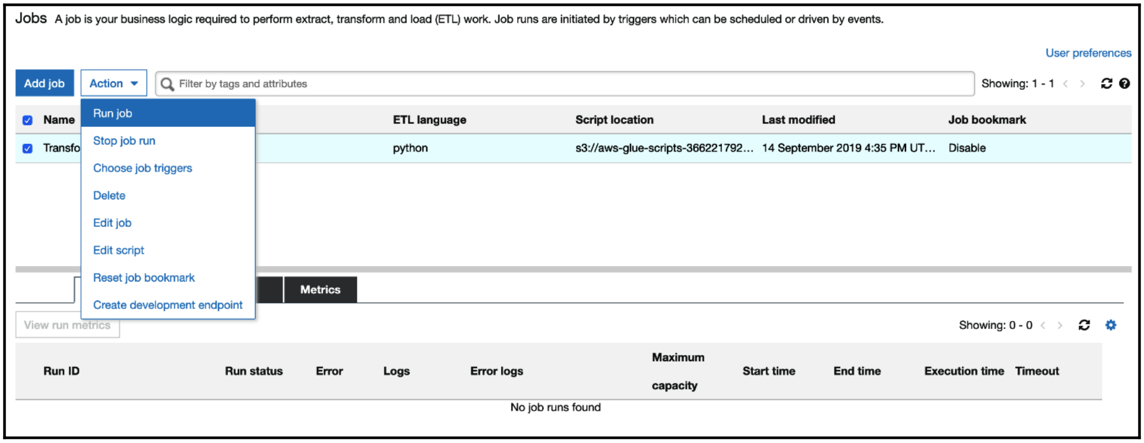


Database Name weather
Table Name raw

Transform Name ApplyMapping

Transform Name ResolveChoice

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)
## @type: DataSource
## @args: [database = "weather", table_name = "raw", transformation_ctx = "datasource0"]
## @return: datasource0
## @inputs: []
datasource0 = glueContext.create_dynamic_frame.from_catalog(database = "weather", table_name = "raw", transformation_ctx = "datasource0")
## @type: ApplyMapping
## @args: [mapping = [("time", "int", "time", "int"), ("main", "struct", "main", "struct"), ("wind", "struct", "wind", "struct"), ("clouds", "struc
## @return: applymapping1
## @inputs: [frame = datasource0]
applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [("time", "int", "time", "int"), ("main", "struct", "main", "struct"), ("wind",
## @type: ResolveChoice
## @args: [choice = "make_struct", transformation_ctx = "resolvechoice2"]
```

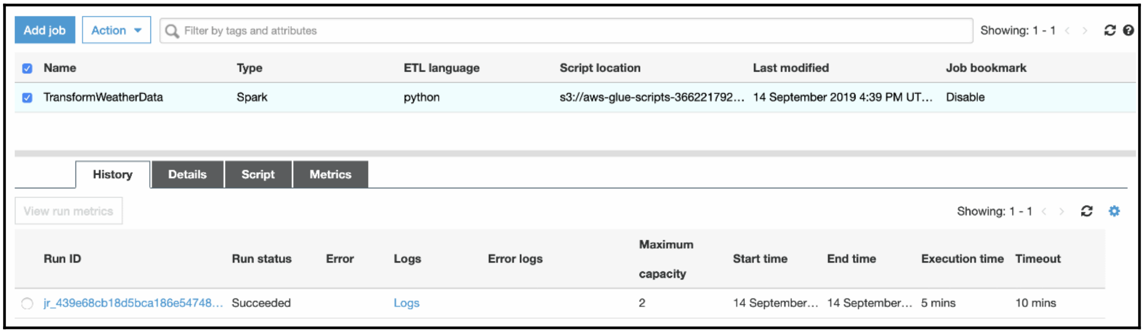


Jobs A job is your business logic required to perform extract, transform and load (ETL) work. Job runs are initiated by triggers which can be scheduled or driven by events.

☑ Name	ETL language	Script location	Last modified	Job bookmark
☑ Transfo	python	s3://aws-glue-scripts-366221792…	14 September 2019 4:35 PM UT…	Disable

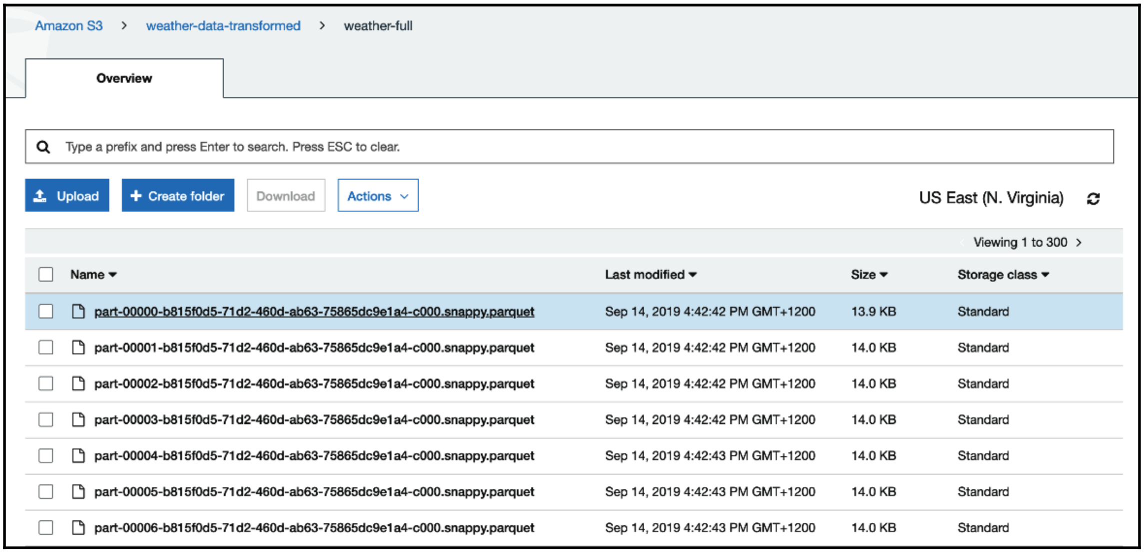
Run ID	Run status	Error	Logs	Error logs	Maximum capacity	Start time	End time	Execution time	Timeout

No job runs found



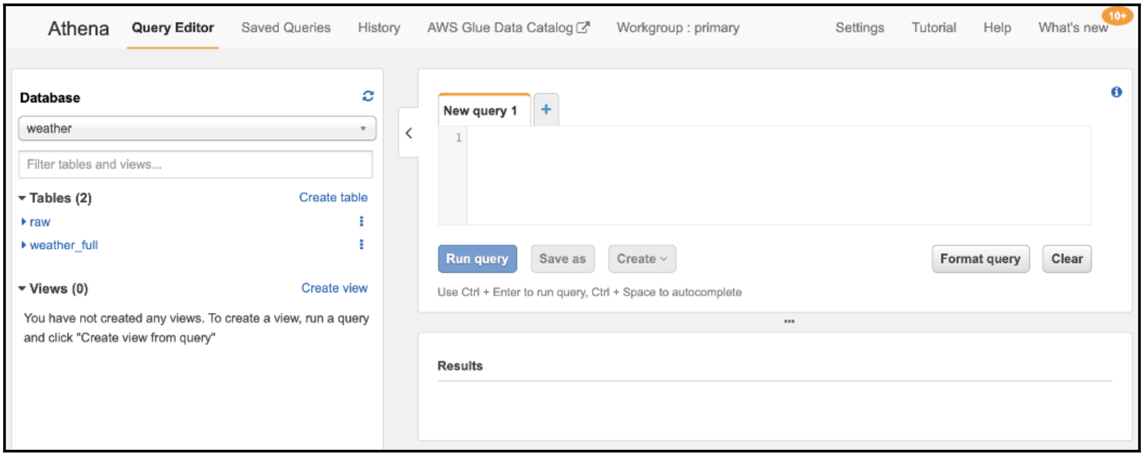
☑ Name	Type	ETL language	Script location	Last modified	Job bookmark
☑ TransformWeatherData	Spark	python	s3://aws-glue-scripts-366221792…	14 September 2019 4:39 PM UT…	Disable

	Run ID	Run status	Error	Logs	Error logs	Maximum capacity	Start time	End time	Execution time	Timeout
○	jr_439e68cb18d5bca186e54748…	Succeeded		Logs		2	14 September…	14 September…	5 mins	10 mins



Amazon S3 > weather-data-transformed > weather-full

□	Name ▾	Last modified ▾	Size ▾	Storage class ▾
□	part-00000-b815f0d5-71d2-460d-ab63-75865dc9e1a4-c000.snappy.parquet	Sep 14, 2019 4:42:42 PM GMT+1200	13.9 KB	Standard
□	part-00001-b815f0d5-71d2-460d-ab63-75865dc9e1a4-c000.snappy.parquet	Sep 14, 2019 4:42:42 PM GMT+1200	14.0 KB	Standard
□	part-00002-b815f0d5-71d2-460d-ab63-75865dc9e1a4-c000.snappy.parquet	Sep 14, 2019 4:42:42 PM GMT+1200	14.0 KB	Standard
□	part-00003-b815f0d5-71d2-460d-ab63-75865dc9e1a4-c000.snappy.parquet	Sep 14, 2019 4:42:42 PM GMT+1200	14.0 KB	Standard
□	part-00004-b815f0d5-71d2-460d-ab63-75865dc9e1a4-c000.snappy.parquet	Sep 14, 2019 4:42:43 PM GMT+1200	14.0 KB	Standard
□	part-00005-b815f0d5-71d2-460d-ab63-75865dc9e1a4-c000.snappy.parquet	Sep 14, 2019 4:42:43 PM GMT+1200	14.0 KB	Standard
□	part-00006-b815f0d5-71d2-460d-ab63-75865dc9e1a4-c000.snappy.parquet	Sep 14, 2019 4:42:43 PM GMT+1200	14.0 KB	Standard



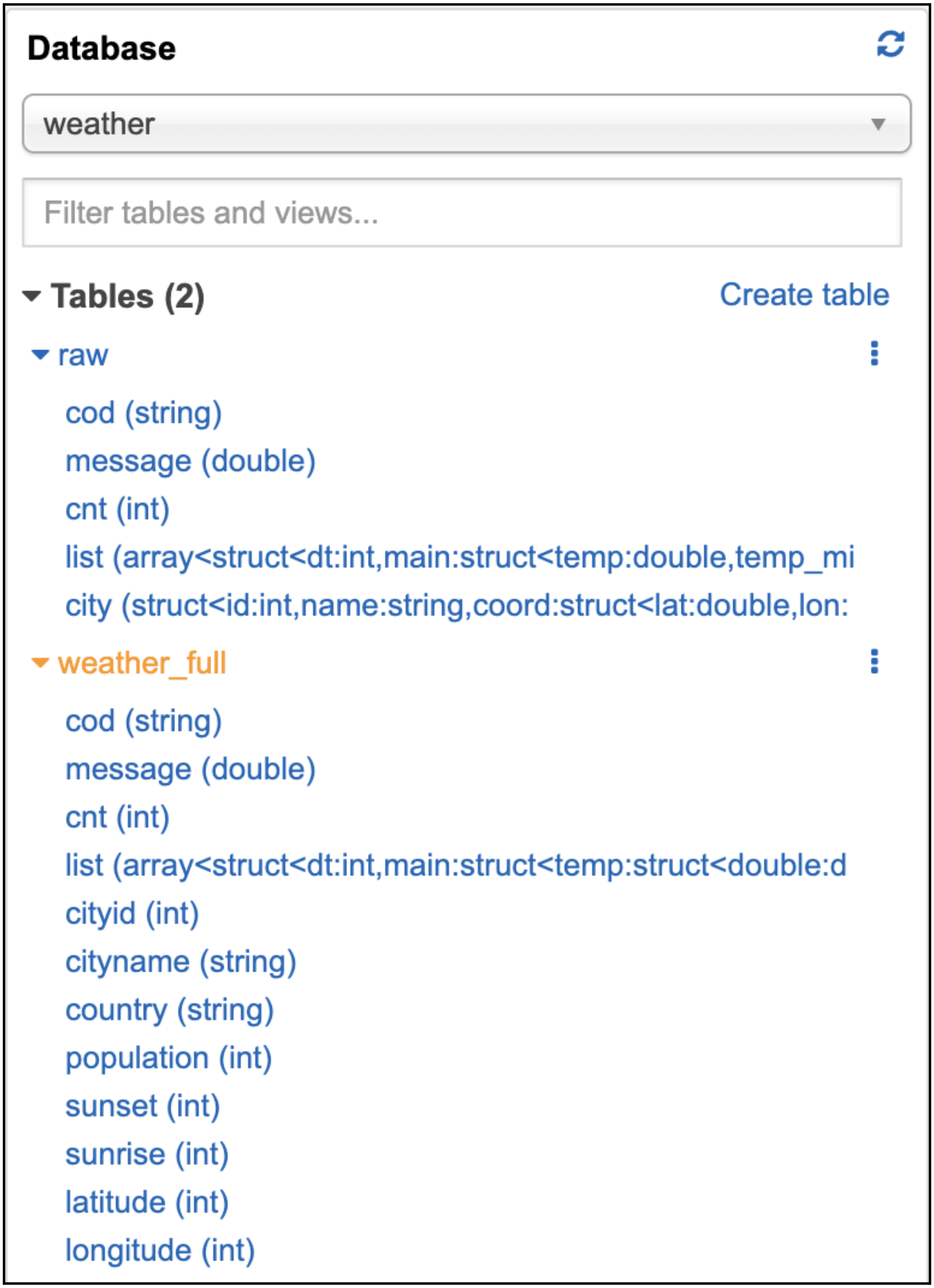
Workgroup : primary

Database

weather

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

You have not created any views. To create a view, run a query and click "Create view from query"



Database

weather

Filter tables and views...

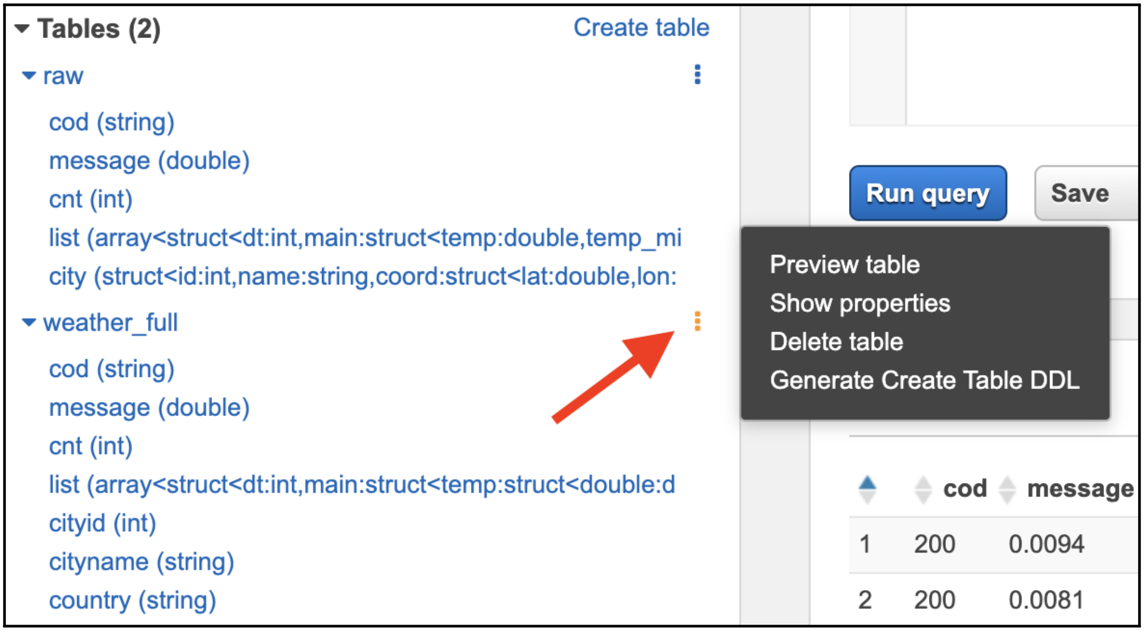
Tables (2) Create table

raw

- cod (string)
- message (double)
- cnt (int)
- list (array<struct<dt:int,main:struct<temp:double,temp_mi
- city (struct<id:int,name:string,coord:struct<lat:double,lon:

weather_full

- cod (string)
- message (double)
- cnt (int)
- list (array<struct<dt:int,main:struct<temp:struct<double:d
- cityid (int)
- cityname (string)
- country (string)
- population (int)
- sunset (int)
- sunrise (int)
- latitude (int)
- longitude (int)



Tables (2)

raw

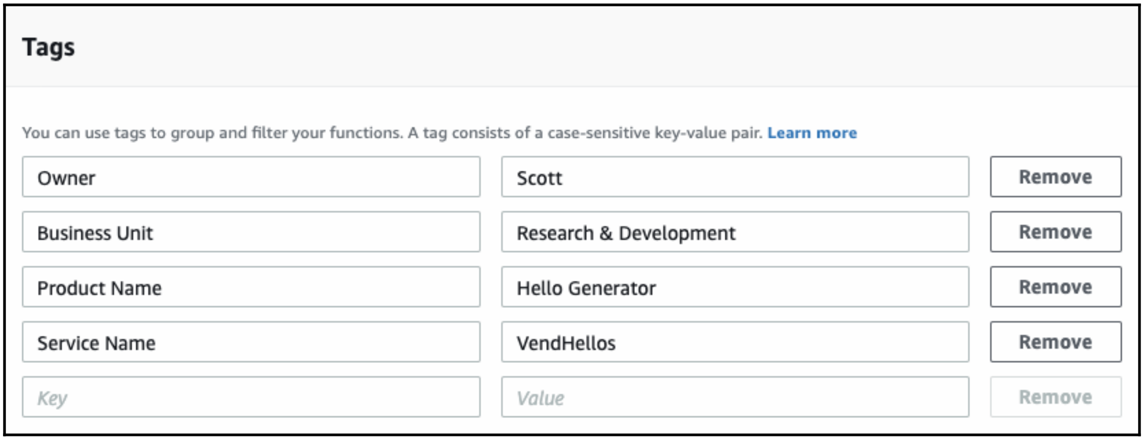
cod (string)
message (double)
cnt (int)
list (array<struct<dt:int,main:struct<temp:double,temp_mi
city (struct<id:int,name:string,coord:struct<lat:double,lon:

weather_full

cod (string)
message (double)
cnt (int)
list (array<struct<dt:int,main:struct<temp:struct<double:d
cityid (int)
cityname (string)
country (string)

	cod	message
1	200	0.0094
2	200	0.0081

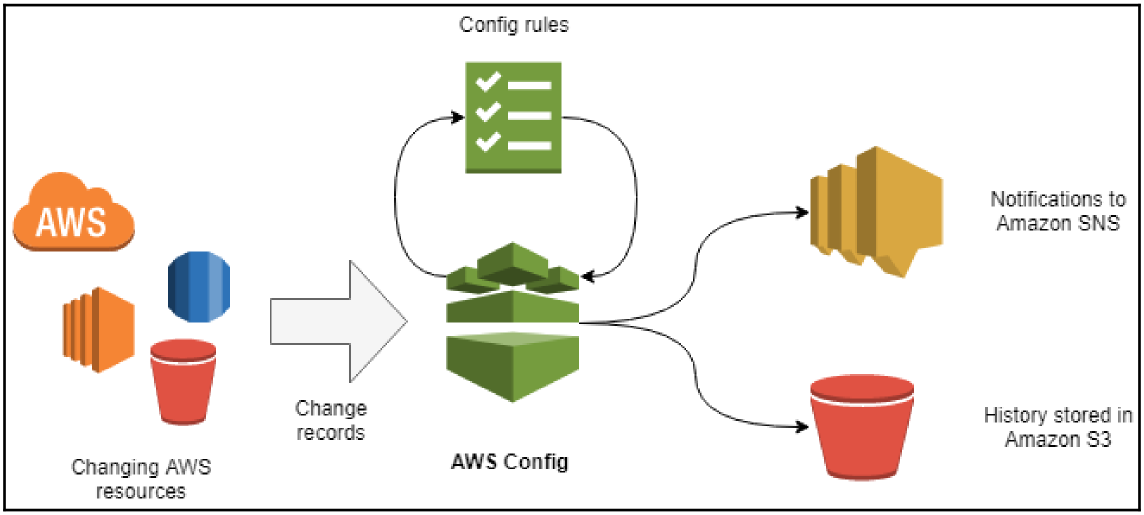
Chapter 10: AWS Automation

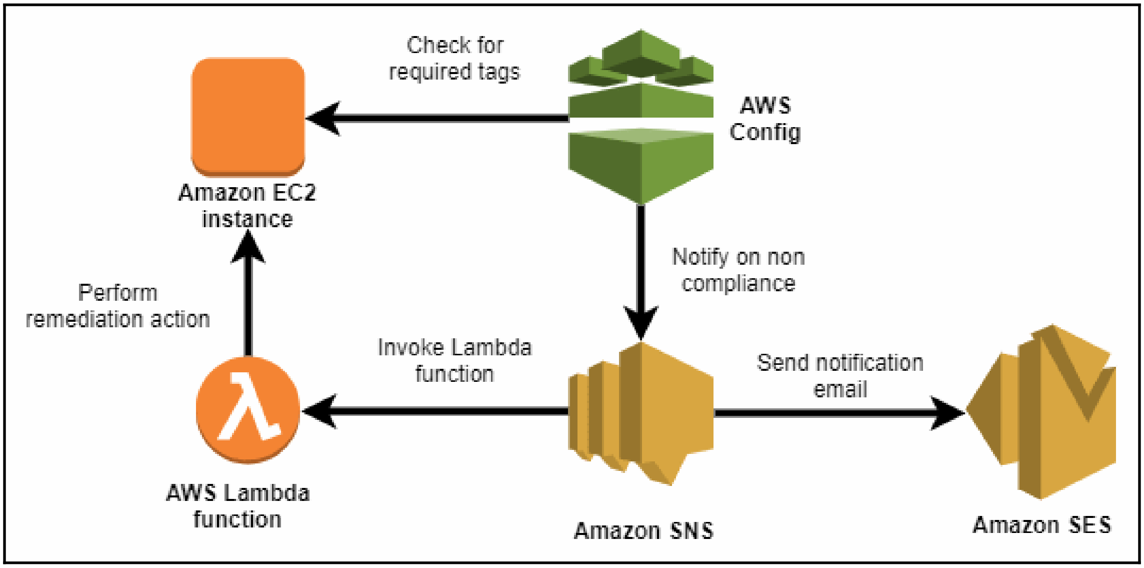


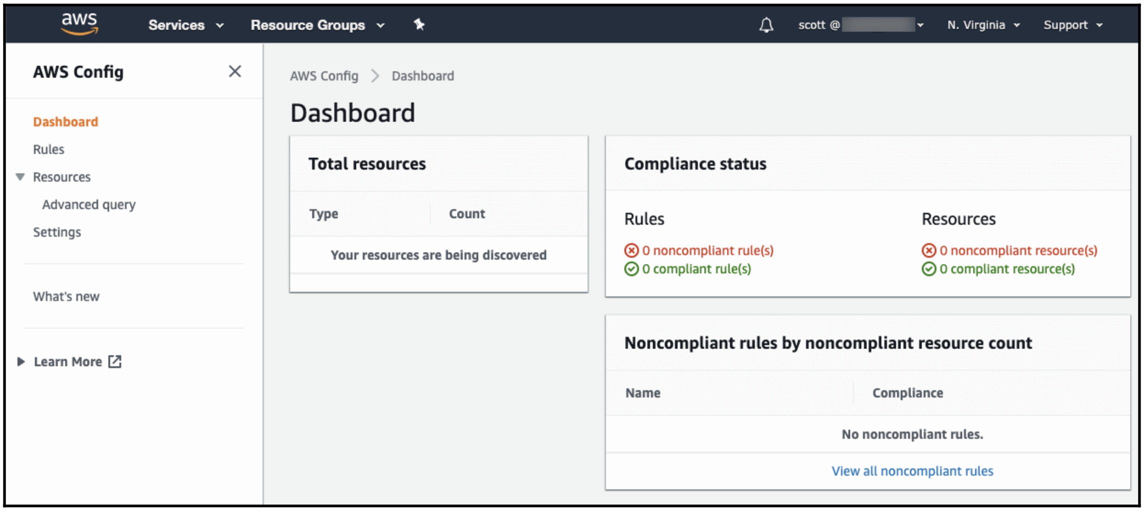
Tags

You can use tags to group and filter your functions. A tag consists of a case-sensitive key-value pair. **Learn more**

Owner	Scott	Remove
Business Unit	Research & Development	Remove
Product Name	Hello Generator	Remove
Service Name	VendHellos	Remove
Key	*Value*	Remove







Dashboard

Total resources

Type	Count
Your resources are being discovered	

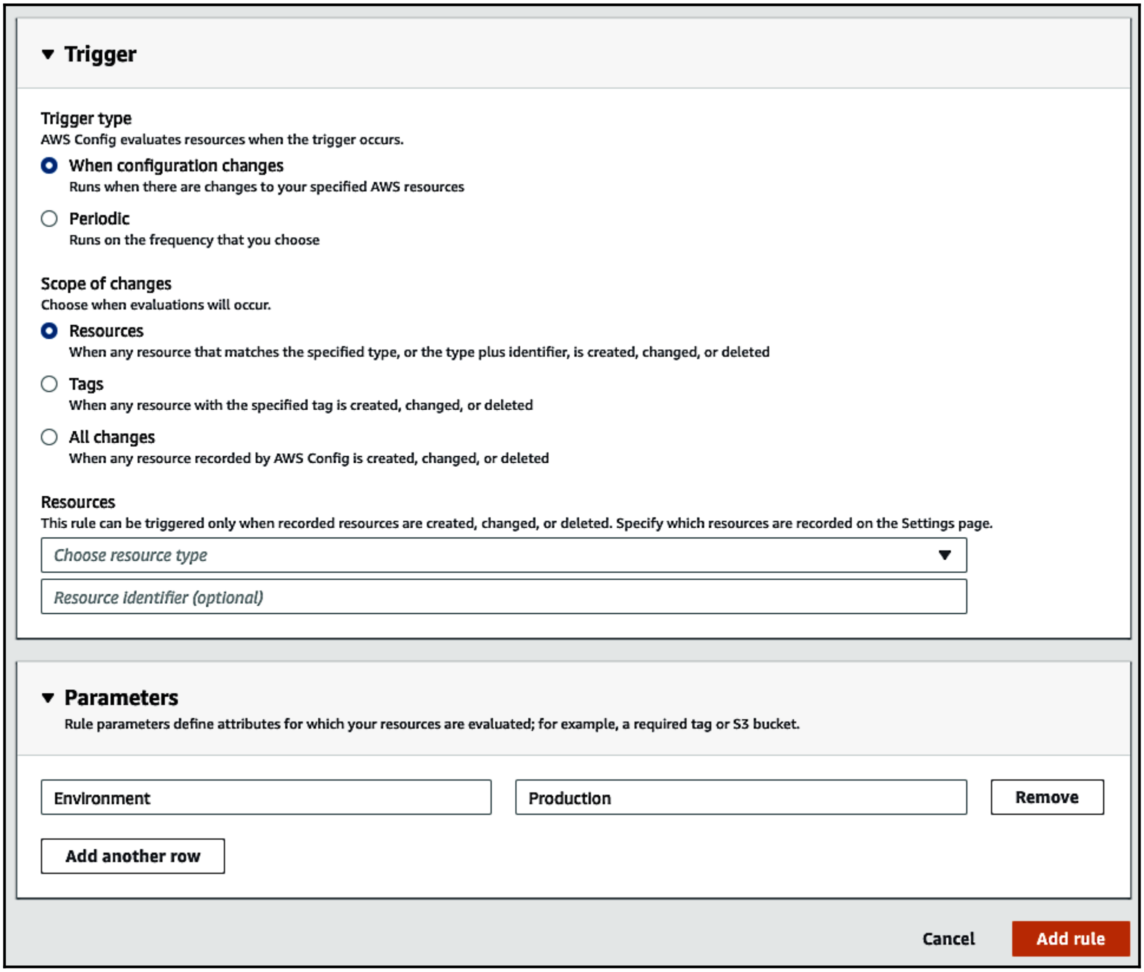
Compliance status

Rules	Resources
⊗ 0 noncompliant rule(s)	⊗ 0 noncompliant resource(s)
✓ 0 compliant rule(s)	✓ 0 compliant resource(s)

Noncompliant rules by noncompliant resource count

Name	Compliance
No noncompliant rules.	

View all noncompliant rules



▼ **Trigger**

Trigger type

AWS Config evaluates resources when the trigger occurs.

● When configuration changes

Runs when there are changes to your specified AWS resources

○ Periodic

Runs on the frequency that you choose

Scope of changes

Choose when evaluations will occur.

● Resources

When any resource that matches the specified type, or the type plus identifier, is created, changed, or deleted

○ Tags

When any resource with the specified tag is created, changed, or deleted

○ All changes

When any resource recorded by AWS Config is created, changed, or deleted

Resources

This rule can be triggered only when recorded resources are created, changed, or deleted. Specify which resources are recorded on the Settings page.

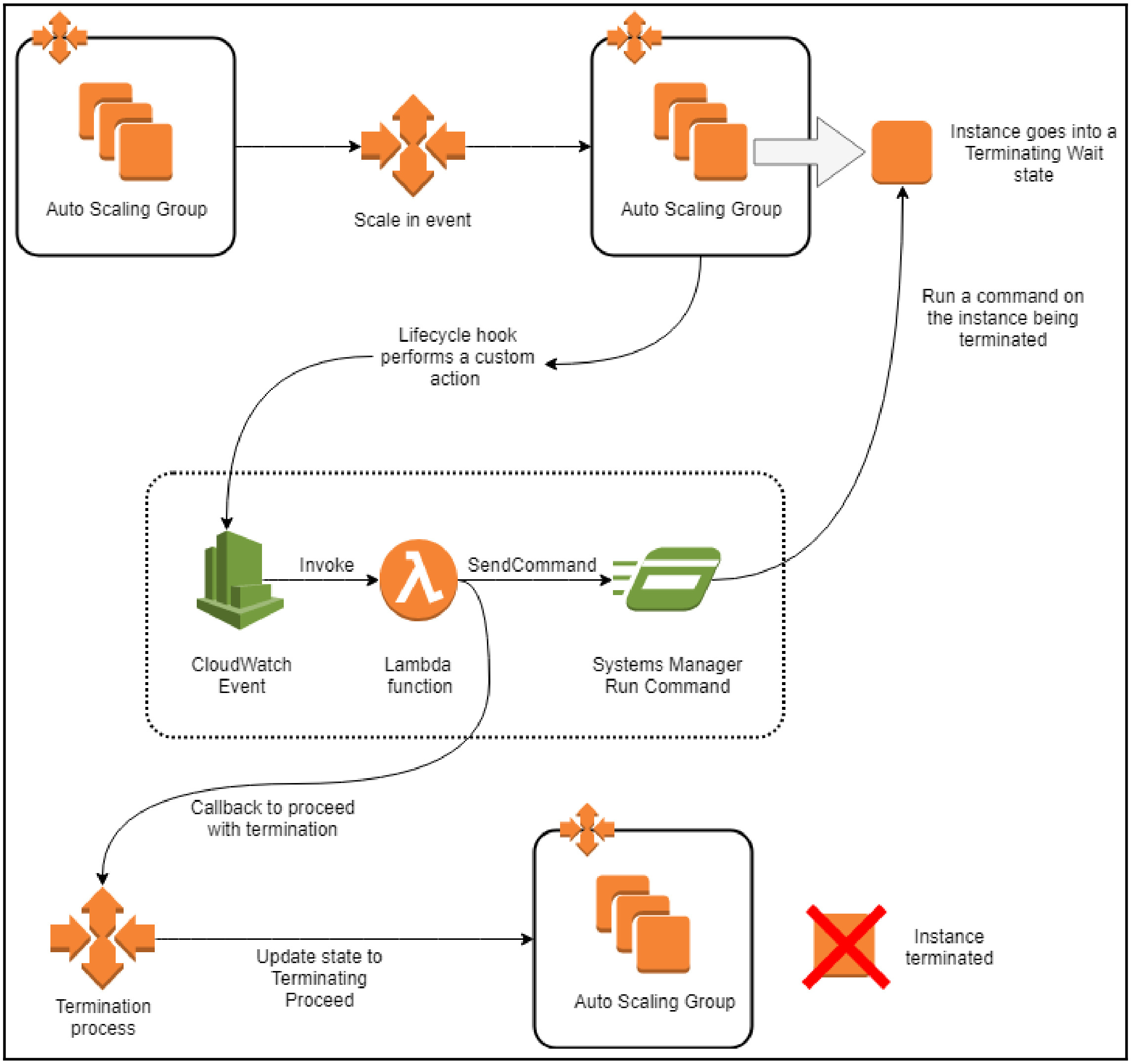
Choose resource type ▼

Resource identifier (optional)

▼ **Parameters**

Rule parameters define attributes for which your resources are evaluated; for example, a required tag or S3 bucket.

Environment	Production	Remove


Auto Scaling Group
Scale in event
Auto Scaling Group
Instance goes into a Terminating Wait state
Run a command on the instance being terminated
Lifecycle hook performs a custom action
Invoke
SendCommand
CloudWatch Event
Lambda function
Systems Manager Run Command
Callback to proceed with termination
Termination process
Update state to Terminating Proceed
Auto Scaling Group
Instance terminated

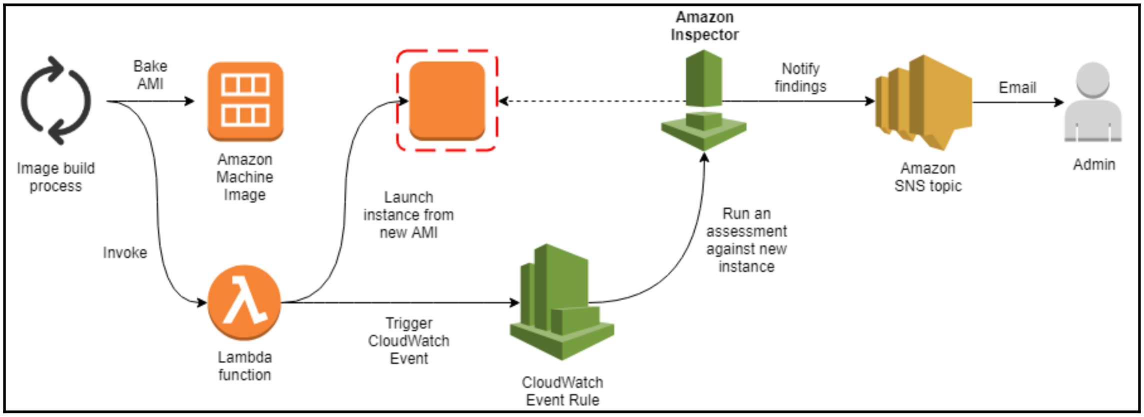
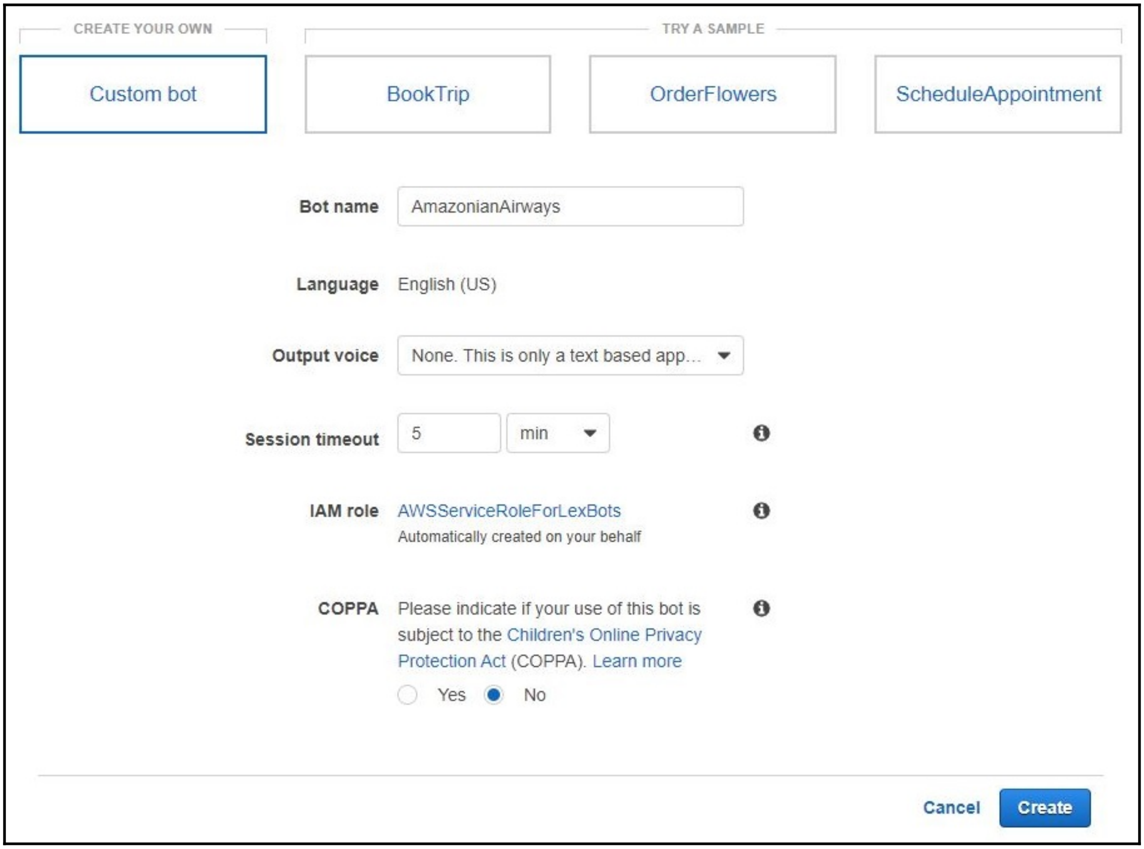


Image build process
Bake AMI
Amazon Machine Image
Invoke
Lambda function
Launch instance from new AMI
Trigger CloudWatch Event
CloudWatch Event Rule
Amazon Inspector
Run an assessment against new instance
Notify findings
Amazon SNS topic
Email
Admin

Chapter 11: Creating Chatbots



Bot name AmazonianAirways

Language English (US)

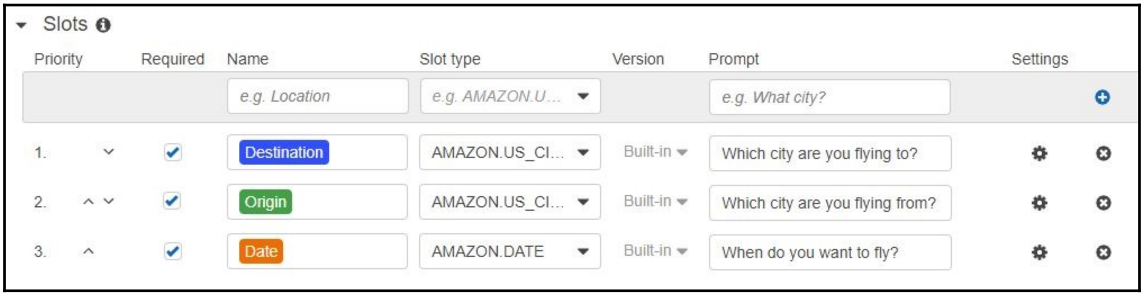
Output voice None. This is only a text based app...

Session timeout 5 min

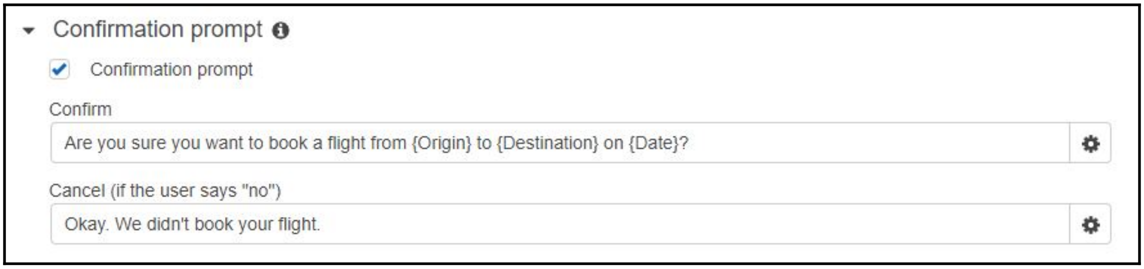
IAM role AWSServiceRoleForLexBots
Automatically created on your behalf

COPPA Please indicate if your use of this bot is subject to the Children's Online Privacy Protection Act (COPPA). Learn more

○ Yes ● No



Priority	Required	Name	Slot type	Version	Prompt	Settings
		e.g. Location	e.g. AMAZON.U...		e.g. What city?	
1.	☑	Destination	AMAZON.US_CI...	Built-in	Which city are you flying to?	
2.	☑	Origin	AMAZON.US_CI...	Built-in	Which city are you flying from?	
3.	☑	Date	AMAZON.DATE	Built-in	When do you want to fly?	



Confirmation prompt

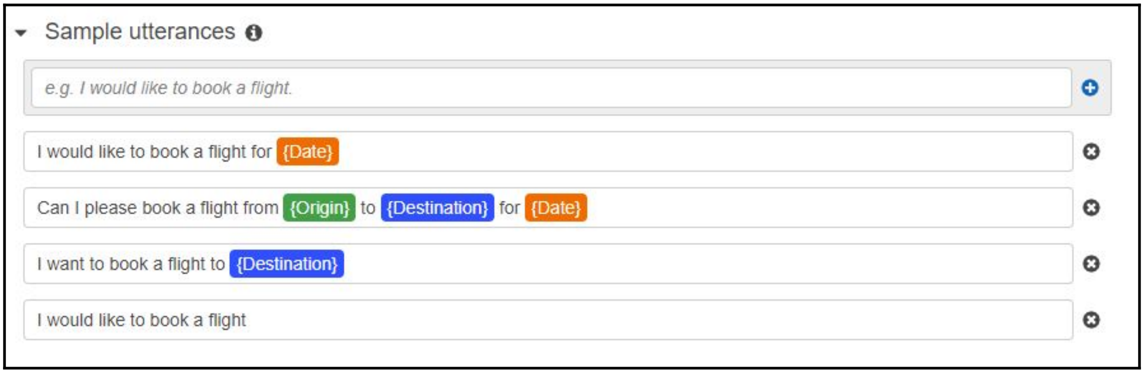
☑ Confirmation prompt

Confirm

Are you sure you want to book a flight from {Origin} to {Destination} on {Date}?

Cancel (if the user says "no")

Okay. We didn't book your flight.



Sample utterances

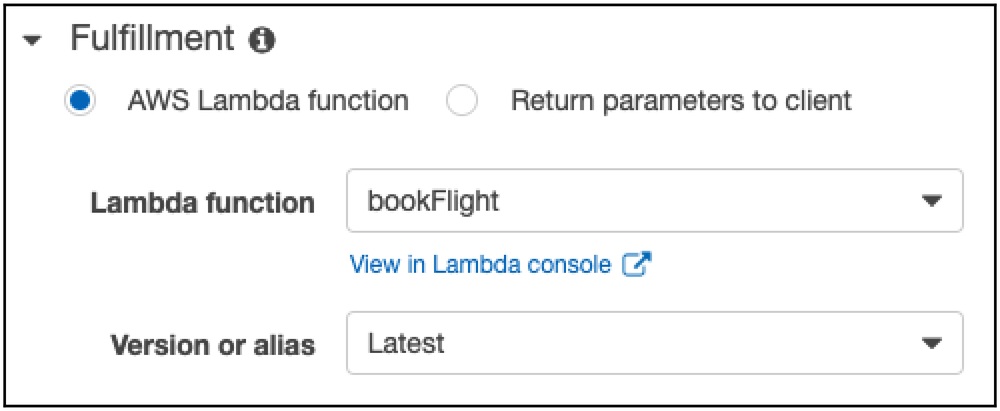
e.g. I would like to book a flight.

I would like to book a flight for {Date}

Can I please book a flight from {Origin} to {Destination} for {Date}

I want to book a flight to {Destination}

I would like to book a flight



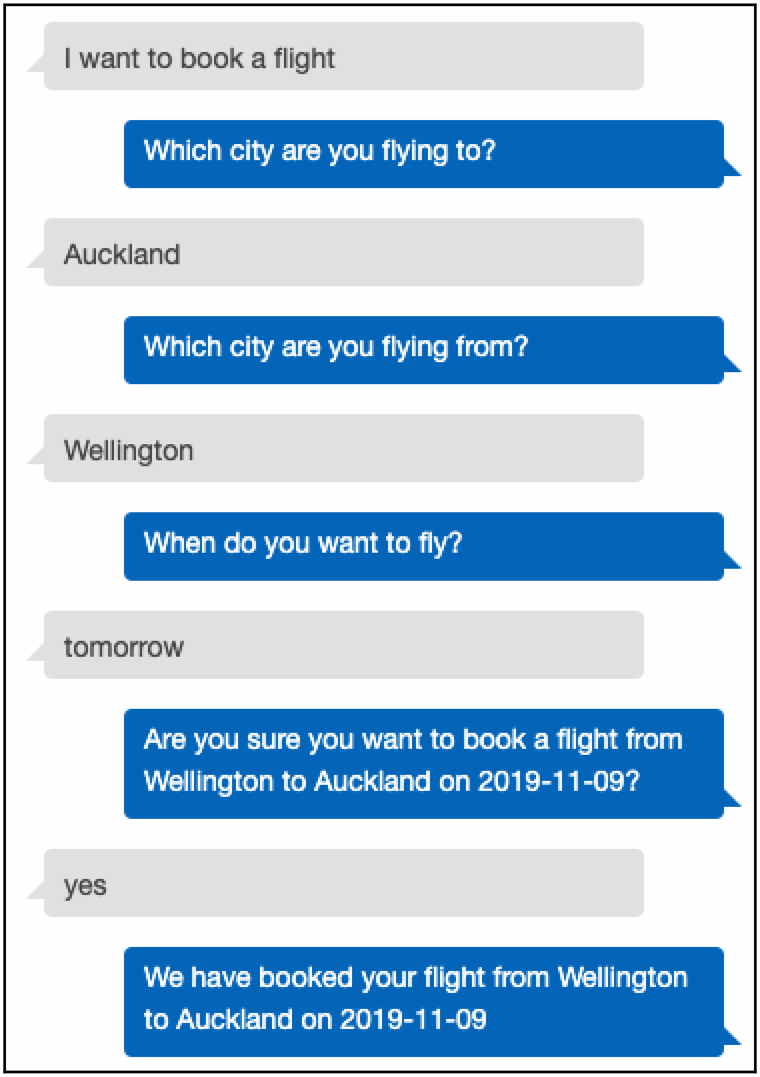
Fulfillment

● AWS Lambda function ○ Return parameters to client

Lambda function bookFlight

View in Lambda console

Version or alias Latest



I want to book a flight

Which city are you flying to?

Auckland

Which city are you flying from?

Wellington

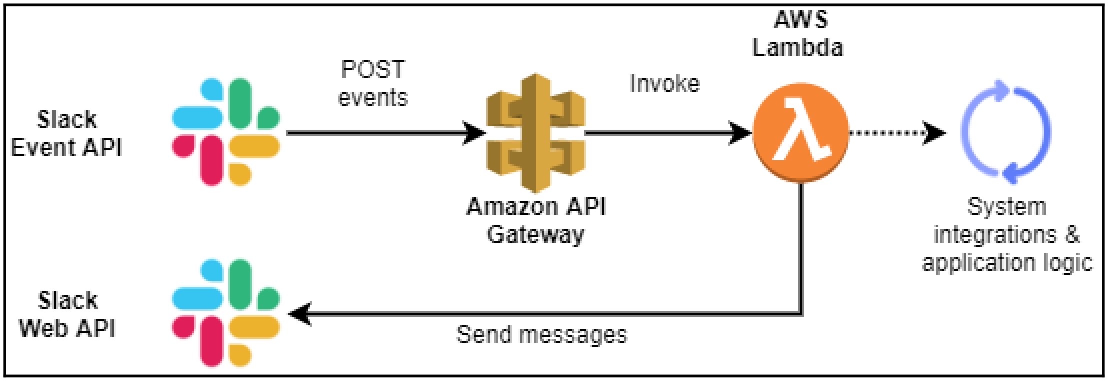
When do you want to fly?

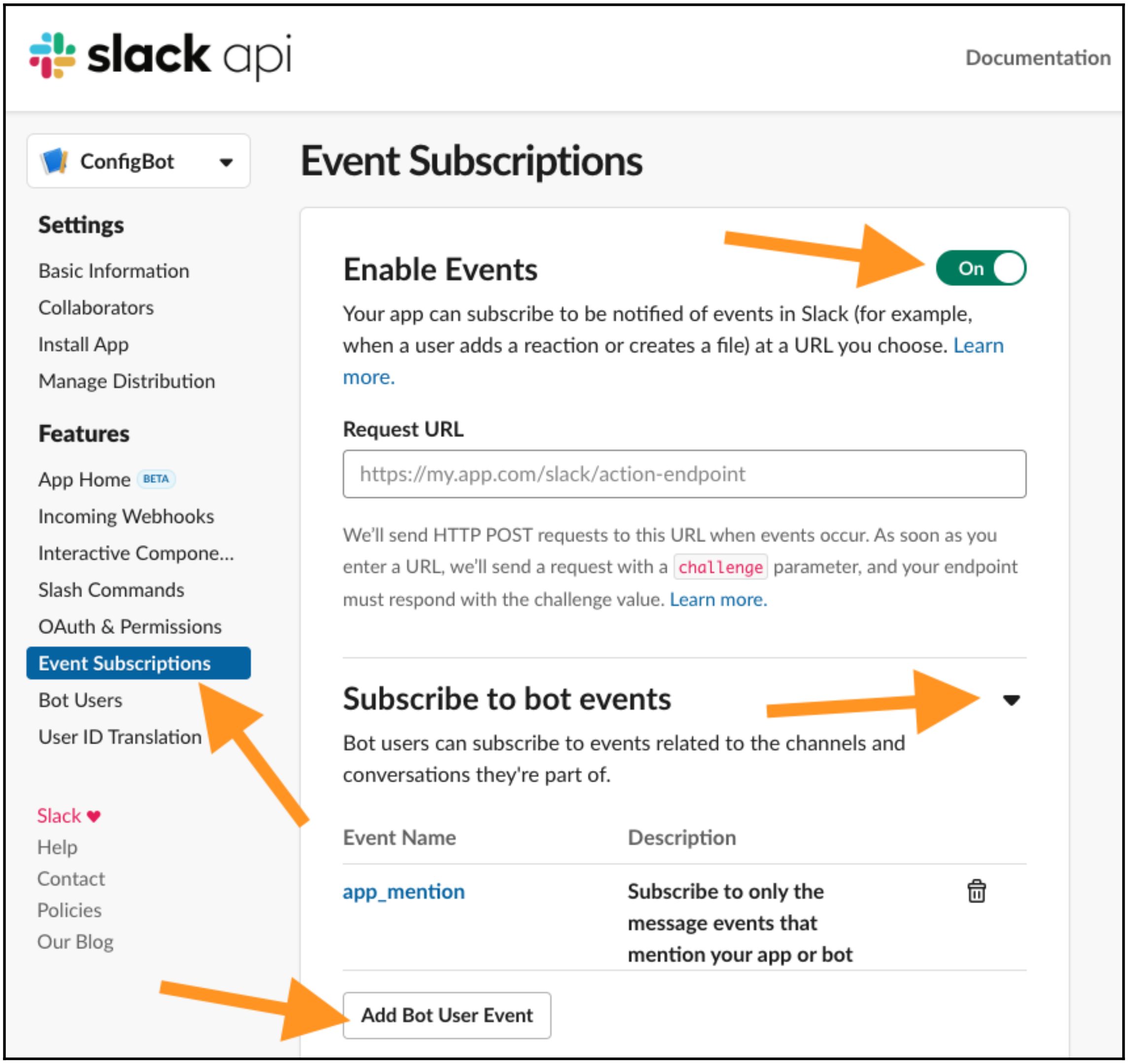
tomorrow

Are you sure you want to book a flight from Wellington to Auckland on 2019-11-09?

yes

We have booked your flight from Wellington to Auckland on 2019-11-09


AWS Lambda
POST events
Invoke
Slack Event API
Amazon API Gateway
System integrations & application logic
Slack Web API
Send messages



Event Subscriptions

Enable Events On

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. Learn more.

Request URL

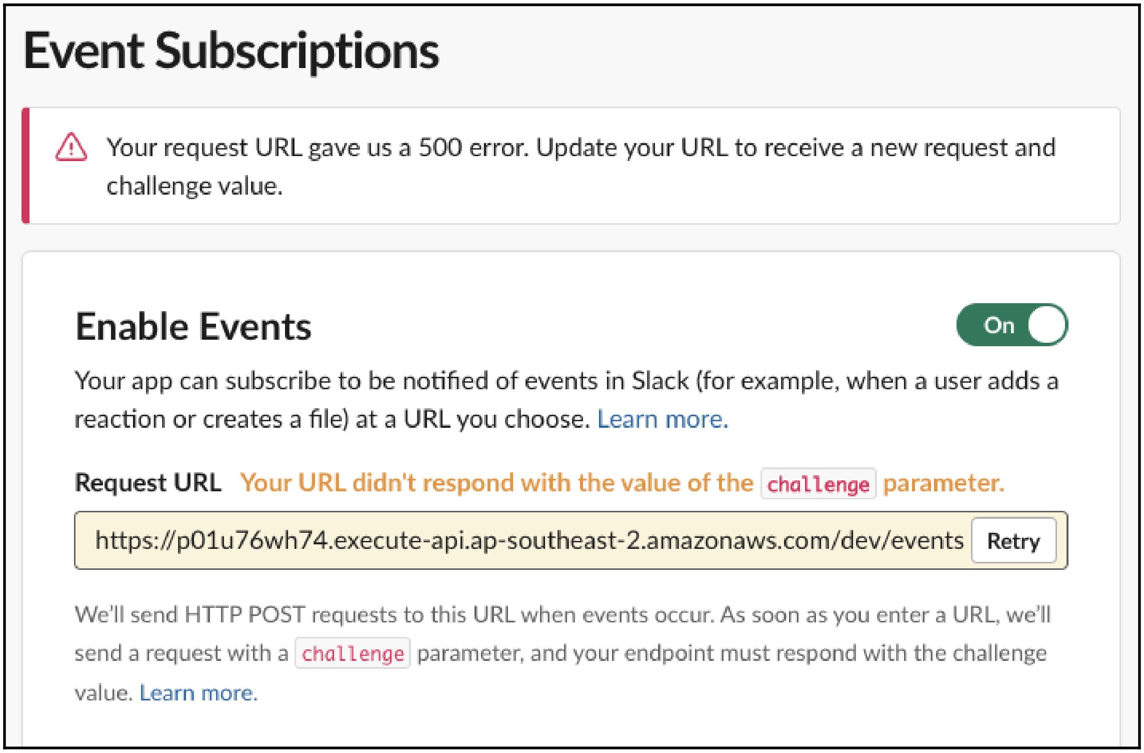
https://my.app.com/slack/action-endpoint

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a `challenge` parameter, and your endpoint must respond with the challenge value. Learn more.

Subscribe to bot events

Bot users can subscribe to events related to the channels and conversations they're part of.

Event Name	Description
app_mention	Subscribe to only the message events that mention your app or bot



Event Subscriptions

Your request URL gave us a 500 error. Update your URL to receive a new request and challenge value.

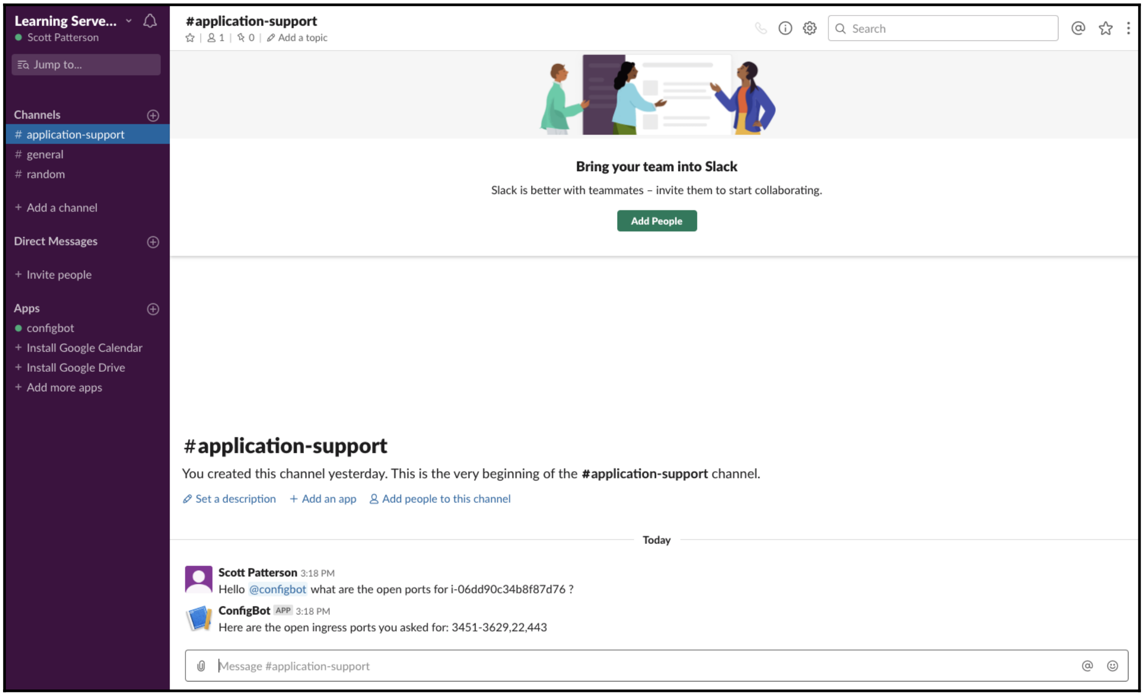
Enable Events

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. Learn more.

Request URL Your URL didn't respond with the value of the `challenge` parameter.

https://p01u76wh74.execute-api.ap-southeast-2.amazonaws.com/dev/events

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a `challenge` parameter, and your endpoint must respond with the challenge value. Learn more.



Bring your team into Slack

Slack is better with teammates – invite them to start collaborating.

#application-support

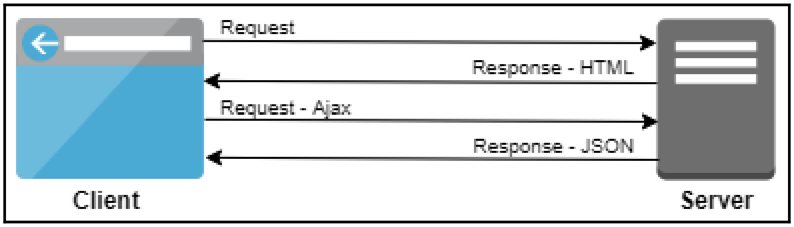
You created this channel yesterday. This is the very beginning of the **#application-support** channel.

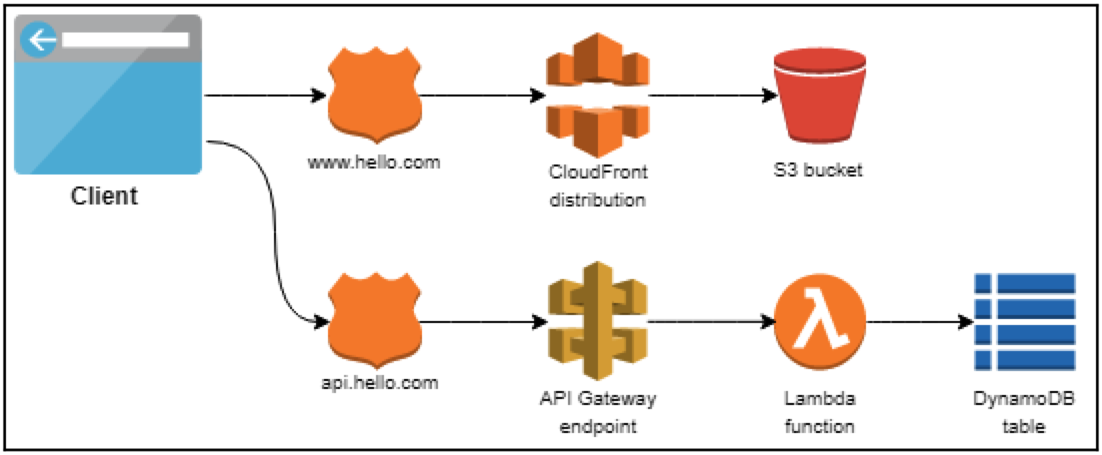
Set a description + Add an app Add people to this channel

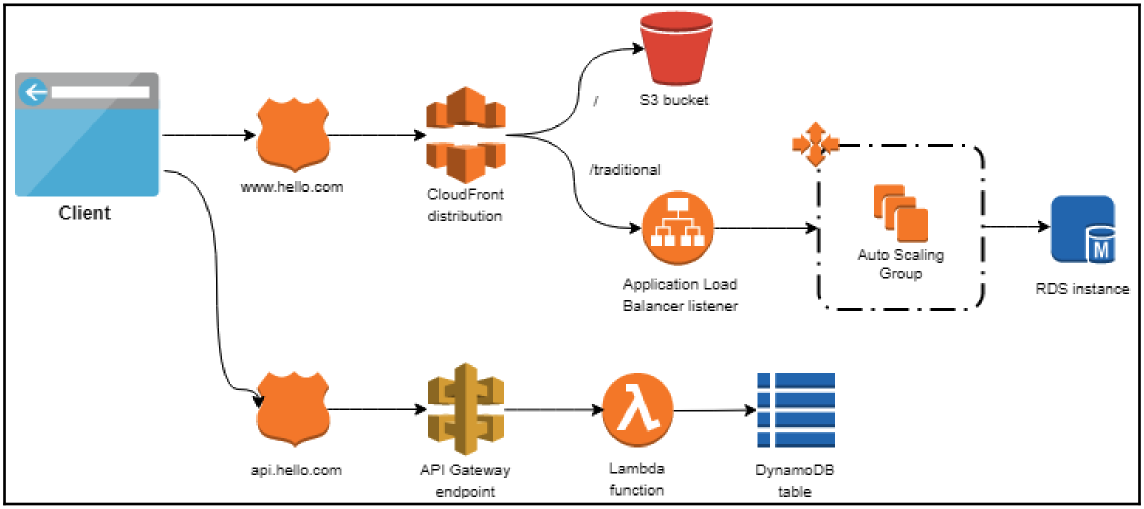
Scott Patterson 3:18 PM
Hello @configbot what are the open ports for i-06dd90c34b8f87d76 ?

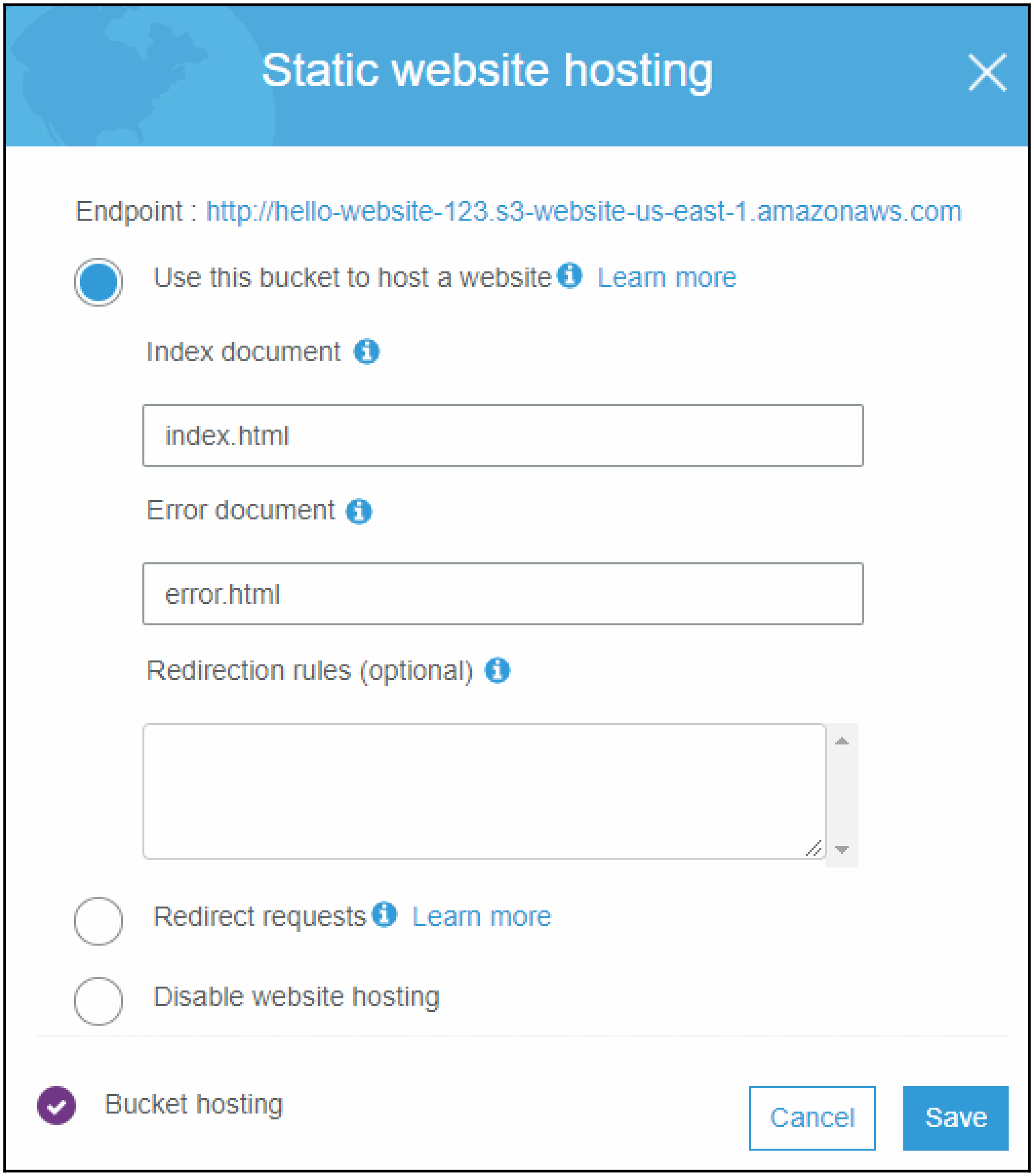
ConfigBot APP 3:18 PM
Here are the open ingress ports you asked for: 3451-3629,22,443

Chapter 12: Hosting Single-Page Web Applications






Client
www.hello.com
CloudFront distribution
/
S3 bucket
/traditional
Application Load Balancer listener
Auto Scaling Group
RDS instance
api.hello.com
API Gateway endpoint
Lambda function
DynamoDB table



Endpoint : http://hello-website-123.s3-website-us-east-1.amazonaws.com

● Use this bucket to host a website ℹ Learn more

Index document ℹ

index.html

Error document ℹ

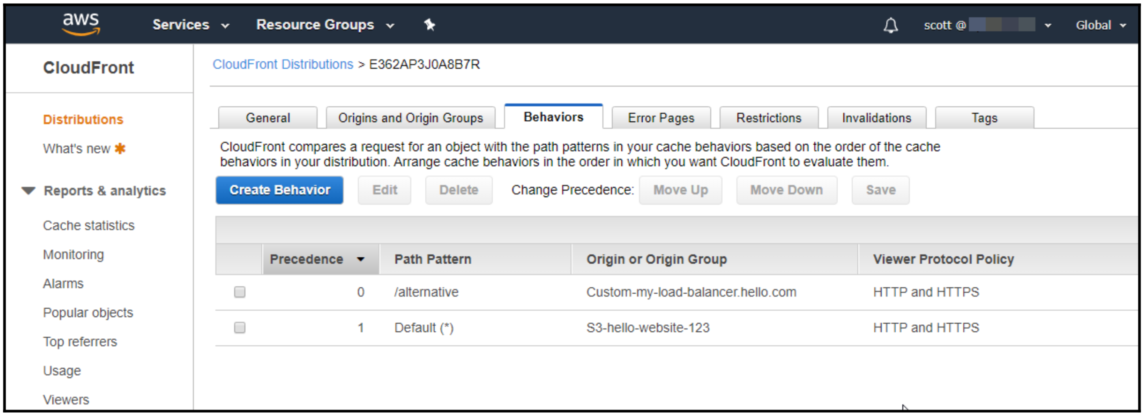
error.html

Redirection rules (optional) ℹ

○ Redirect requests ℹ Learn more

○ Disable website hosting

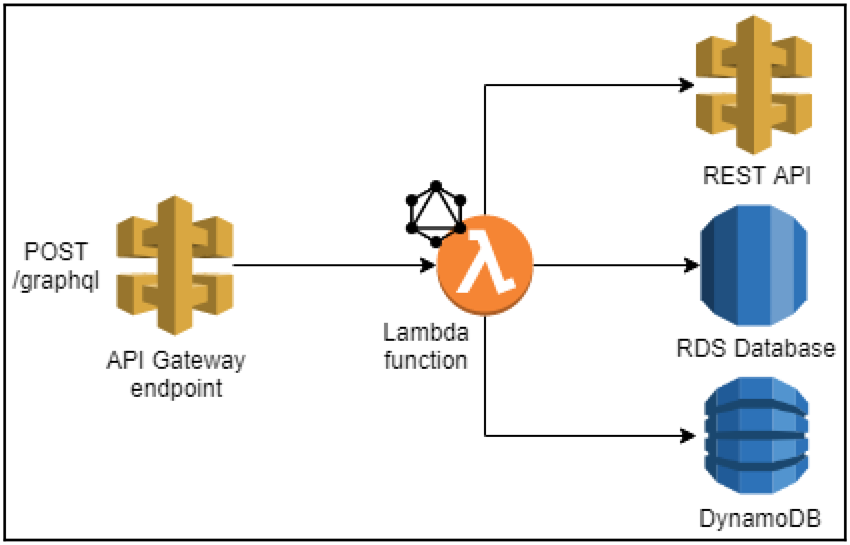
✓ Bucket hosting

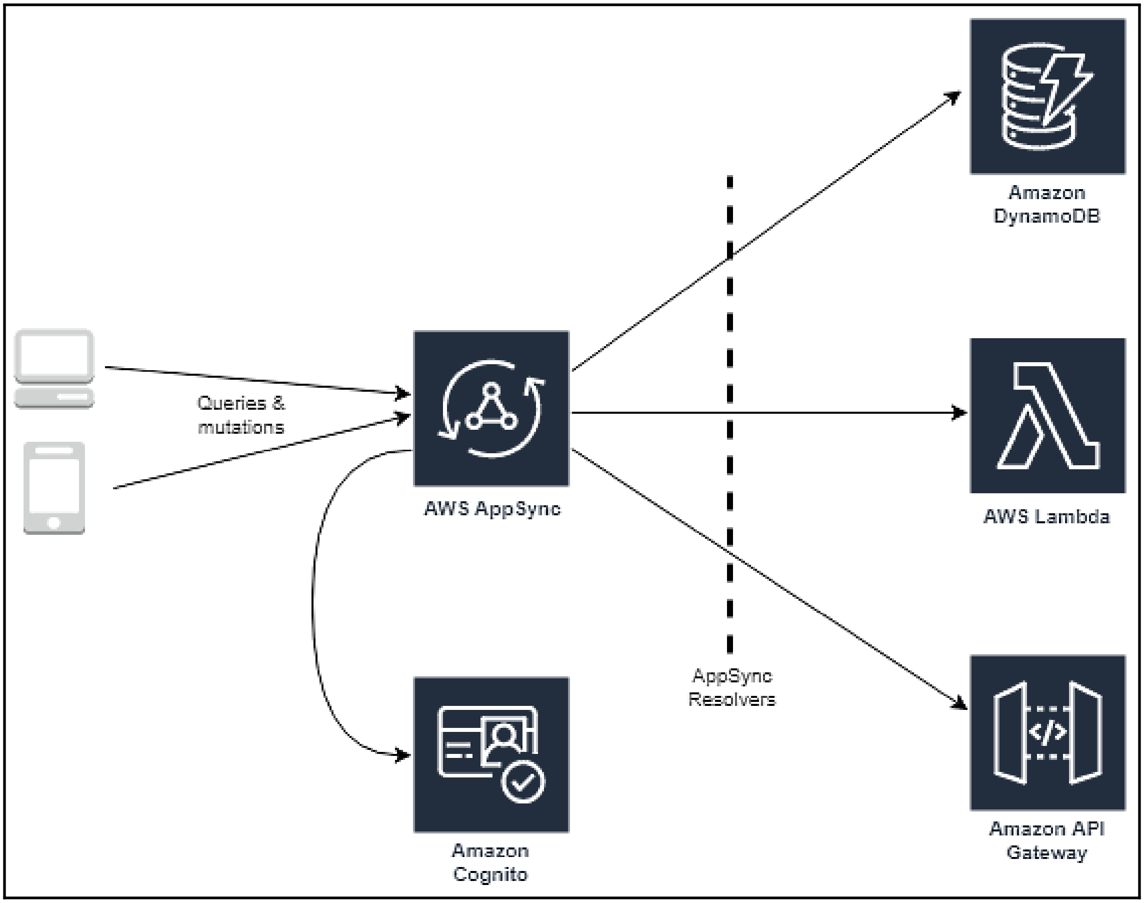


CloudFront compares a request for an object with the path patterns in your cache behaviors based on the order of the cache behaviors in your distribution. Arrange cache behaviors in the order in which you want CloudFront to evaluate them.

	Precedence	Path Pattern	Origin or Origin Group	Viewer Protocol Policy
□	0	/alternative	Custom-my-load-balancer.hello.com	HTTP and HTTPS
□	1	Default (*)	S3-hello-website-123	HTTP and HTTPS

Chapter 13: GraphQL APIs




Queries &
mutations
AWS AppSync
Amazon
Cognito
AppSync
Resolvers
Amazon
DynamoDB
AWS Lambda
Amazon API
Gateway