

Part 4 - AWS Lambda

We continue to build our Dragon App
and back that API up with code,
which will be hosted using **AWS Lambda**.

AWS Lambda is a **serverless** compute service
that allows you to create functions that run and **scale on demand**,
and **without any server management**.

AWS Lambda is a compute service
that lets you run code without provisioning or managing servers.

- This allows you to execute your code only when needed.
- It has the ability to scale automatically from a few requests per day to thousands per second.
- Lambda runs your code on a highly available compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, capacity provisioning and automatic scaling and code monitoring and logging.

When to use Lambda?

- You can use AWS Lambda to run your code in response to events.
- Integrate it with your REST API through Amazon API Gateway.
- Invoke your code using API calls.

Let's review a few terms that will be beneficial for our discussion of Lambda.

A **Lambda function** is simply
the packaging of the code that processes events.

Runtime in Lambda allows functions
in different languages to run in the same base execution environment.

An **event** is a JSON-formatted document that contains data for a function to process.

The **Lambda runtime** converts the **event** to an object
and passes it to the **function** code.

Concurrency is the number of requests
that your function is serving at any given time.

When your function is invoked,
Lambda provisions an instance of it to process the event.

When the function code finishes running,
it can handle another request.

If the function is invoked again while a request is still processing,
another instance of the function is provisioned,
which will increase the function's concurrency.

A **trigger** is a resource or configuration
that invokes a Lambda function.

Triggers can include

- AWS services that can be configured to invoke a function
- applications that you develop
- event source mapping
 - which is a resource in Lambda that reads items from a stream or queue and invokes a function.

AWS Lambda Execution Environment

Lambda execution environment
is the environment in which your code runs.

It's actually a **micro virtual machine**, but you don't
have access to the operating system behind the scene.

So how can you modify that environment?

We can use the configuration settings that are available for your Lambda function during the creation or by editing it afterward.

Let's see how to modify the settings of an already created Lambda function.



Services ▾

Resource Groups ▾



buildingmodernapps @ 3022-1...

Central ▾



Lambda > Functions



Actions ▾

Create functi

< 1 >

Functions (1)

Filter by tags and attributes or search by keyword



Function name	Description	Runtime	Code size	Last modified
hello-world		Node.js 12.x	304 bytes	1 hour ago

Here is the Lambda console.



Services ▾

Resource Groups ▾



buildingmodernapps @ 3022-1...

Central ▾



Lambda > Functions

Functions (1)



Actions ▾

Create function

< 1 >

Function name	Description	Runtime	Code size	Last modified
hello-world		Node.js 12.x	304 bytes	1 hour ago

Assume we already has created a Lambda function. Click on it.



Services

Resource Groups



buildingmodernapps @ 3022-1...

Central

Lambda > Functions > hello-world

ARN - arn:aws:lambda:ca-central-1:302211264422:function:hello-

hello-world

Throttle

Qualifiers ▾

Actions ▾

Select a test event ▾

Test

Save

Configuration

Permissions

Monitoring

▼ Designer



+ Add trigger

+ Add destination

Function code Info

Actions ▾

File Edit Find View Go Tools Window

Save Test ▾



Feedback English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy



Services

Resource Groups



buildingmodernapps @ 3022-1...

Central

hello-world

Throttle

Qualifiers

Actions

Select a test event

Test

Se

Value

No tags

No tags associated with this function.

[Manage tags](#)

Basic settings Info

[Edit](#)

Description

-

Runtime

Node.js 12.x

Handler Info

index.handler

Memory (MB)

128

Timeout

0 min 3 sec

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](#)

VPC Info

[Edit](#)

No VPC configuration

This function isn't connected to a VPC.

[Edit](#)

In the Basic Setting, you can edit your Lambda Function.



Edit basic settings

Basic settings Info

Description - *optional*

Runtime

Node.js 12.x

Latest supported

.NET Core 3.1 (C#/PowerShell)

Go 1.x

Java 11

Node.js 12.x

Python 3.8

Ruby 2.7

Other supported

.NET Core 2.1 (C#/PowerShell)

Java 8

Node.js 10.x

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.



Feedback



English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

Runtime is the language of the code

Note: If your code relies on an specific OS kernel feature or Library, you need to make sure that the underlying AWS system support it.

For example, it use Amazon Linux for Python. 3.7,
while for Python 3.8, it uses Amazon Linux 2.

If you want to use a different language
like PHP, for example, use AWS Lambda **custom runtime**.

Edit basic settings

Basic settings Info

Description - *optional*

Runtime

Node.js 12.x

Handler Info

index.handler

Memory (MB)

Your function is allocated CPU proportional to the memory configured.



Timeout

0 min 3 sec

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
- Create a new role from AWS policy templates

Existing role

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



Feedback



English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)

CPU Memory to run that Lambda function (note: this is CPU not RAM)

Edit basic settings

Basic settings Info

Description - *optional*

Runtime

 ▾

Handler Info

Memory (MB)

Your function is allocated CPU proportional to the memory configured.



Timeout

 min sec

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
- Create a new role from AWS policy templates

Existing role

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



Feedback



English (US)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy Policy](#)

use timeout since billing is based on execution time (how about an infinite loop error in your code?)

hello-world

Throttle

Qualifiers ▾

Actions ▾

Select a test event ▾

Test

Se

No file systems

To connect to a file system, you must first connect your function to the VPC where your file system runs.

Concurrency

Unreserved account concurrency **1000**

- Use unreserved account concurrency
- Reserve concurrency

Provisioned concurrency

To enable your function to scale without fluctuations in latency, use provisioned concurrency. Provisioned concurrency runs continually and has separate pricing for concurrency and execution duration. [Learn more](#)

Provisioned concurrency configurations (0)

C

Edit

Remove

Add

 Find configuration

Qualifier

▼

Type

▼

Provisioned concurrency

▼

Status

▼

Details

No configurations

Add configuration

how many Lambda functions can run concurrently per region.

hello-world

Throttle

Qualifiers ▾

Actions ▾

Select a test event ▾

Test

S

No file systems

To connect to a file system, you must first connect your function to the VPC where your file system runs.

Concurrency

Unreserved account concurrency 900

 Use unreserved account concurrency Reserve concurrency

100

Provisioned concurrency

To enable your function to scale without fluctuations in latency, use provisioned concurrency. Provisioned concurrency runs continually and has separate pricing for concurrency and execution duration. [Learn more](#)

Provisioned concurrency configurations (0)

Edit

Remove

Add

Find configuration

Qualifier	Type	Provisioned concurrency	Status	Details
No configurations				
Add configuration				

Maximum of concurrency is 100

Environment variables settings enable you to configure your Lambda code based on the environment like a development, QA, or production environment.

You can pass different parameters to your Lambda function without having to modify the code.

A hello-world - Lambda x + ca-central-1.console.aws.amazon.com/lambda/home?region=ca-central-1#/functions/hello-world?tab=configuration

Guest

buildingmodernapps @ 3022-1... Central Support

hello-world

Throttle Qualifiers Actions Select a test event Test Save

1:1 JavaScript Spaces: 4

Environment variables (0)

Key Value

No environment variables

No environment variables associated with this function.

Manage environment variables

Edit

Tags (0)

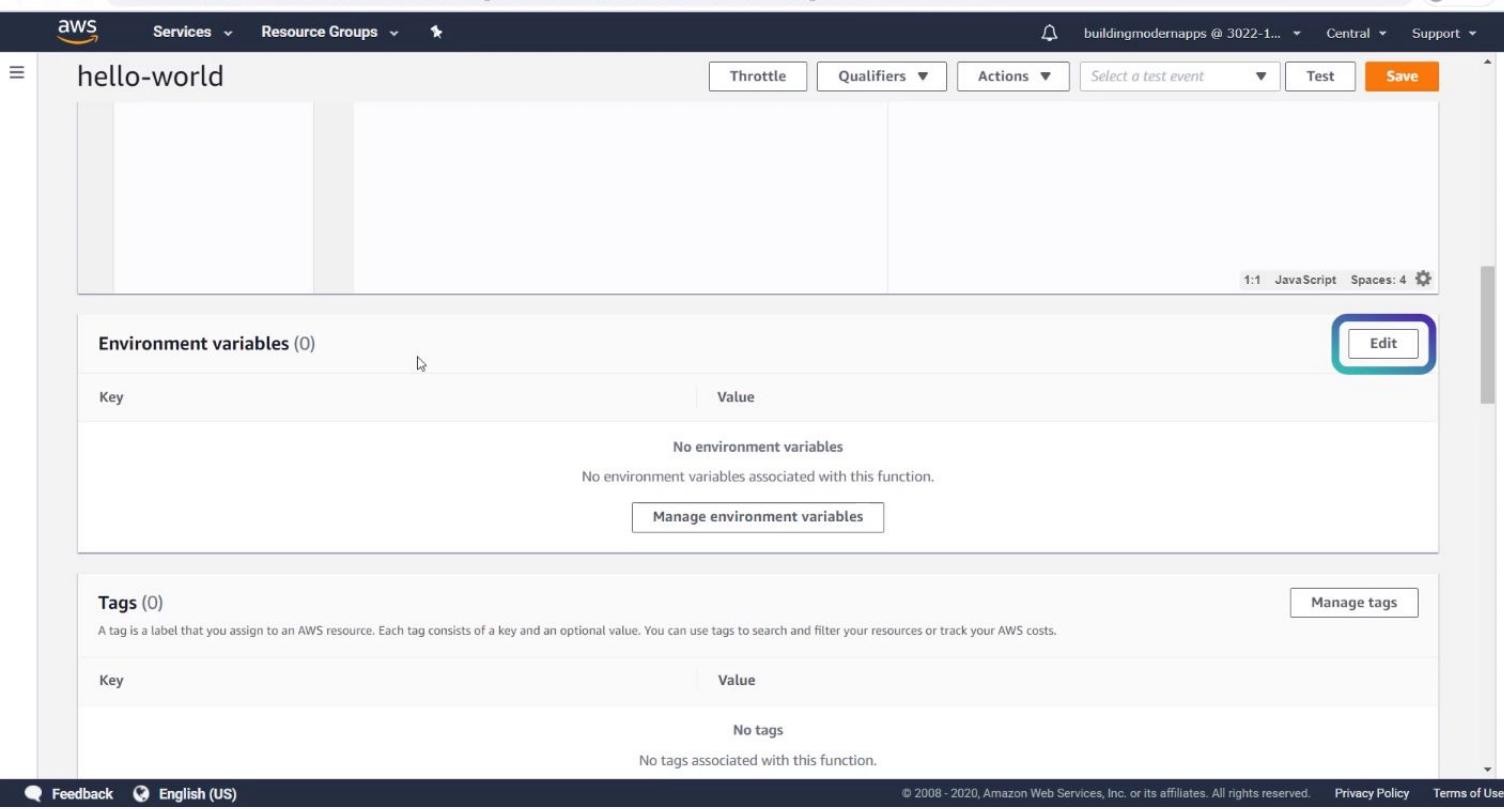
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Manage tags

Key Value

No tags

No tags associated with this function.



Edit environment variables

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](#) ⓘ

Key	Value	
Environment	Production	Remove

[Add environment variable](#)

▼ Encryption configuration

Encryption in transit: [Info](#)

Enable helpers for encryption in transit

AWS KMS key to encrypt at rest

Choose an AWS KMS key to encrypt the environment variables at rest, or simply let Lambda manage the encryption.

(default) aws/lambda

Use a customer master key

[Cancel](#)

[Save](#)

Define Environmental Variables (Note: you need to modify Lambda Code to use Encryption)

Finally, the name of this virtualization technology for creating and managing secure, multi-tenant containers and function-based services is **firecracker**.

AWS Lambda Permissions

Recap:

AWS commonly relies on role-based access.

We create a role in IAM that has a policy associated with it.

The policy allows or denies certain AWS API calls to be made.

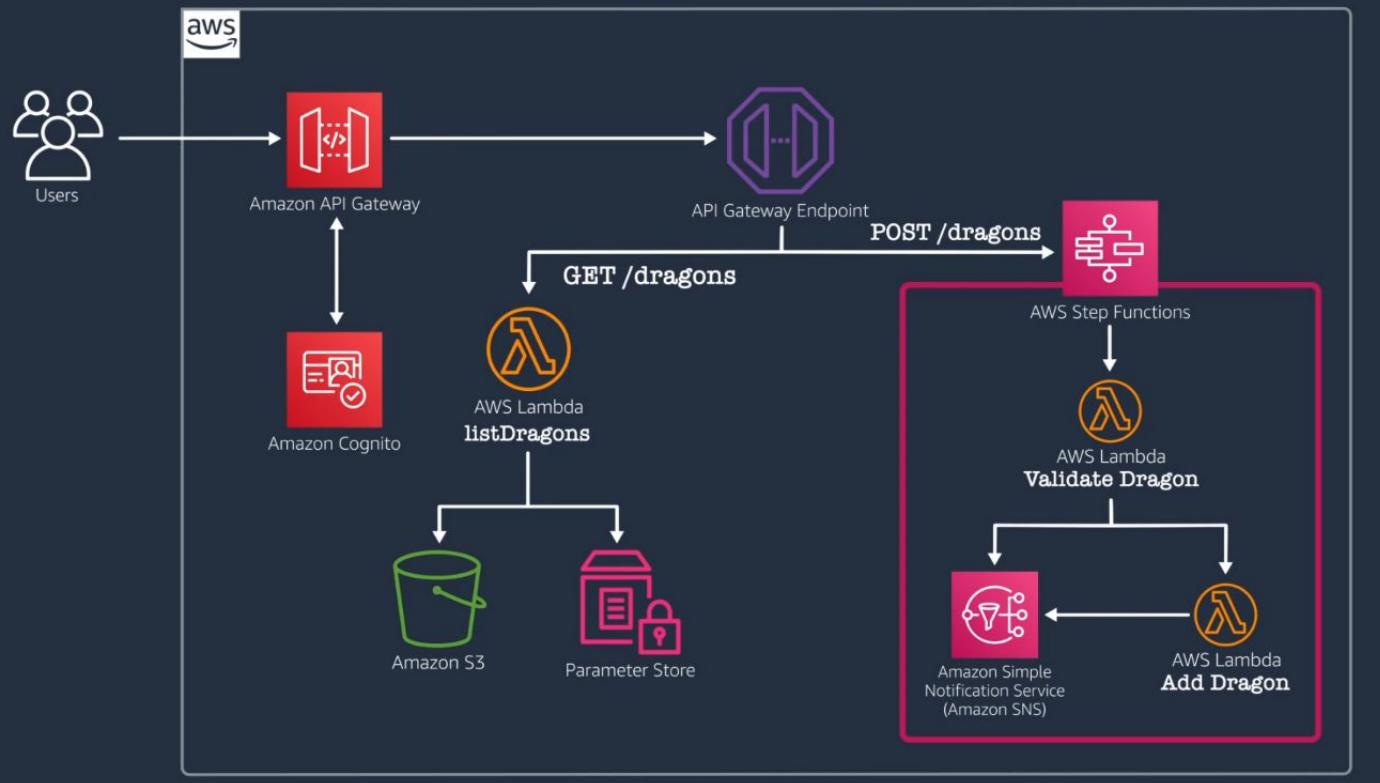
Recap:

AWS services, applications, or identities can then assume the role and gain access to **temporary credentials**, which are supplied by **AWS Security Token Service** and contain the **permissions** defined in the policy associated with the role.

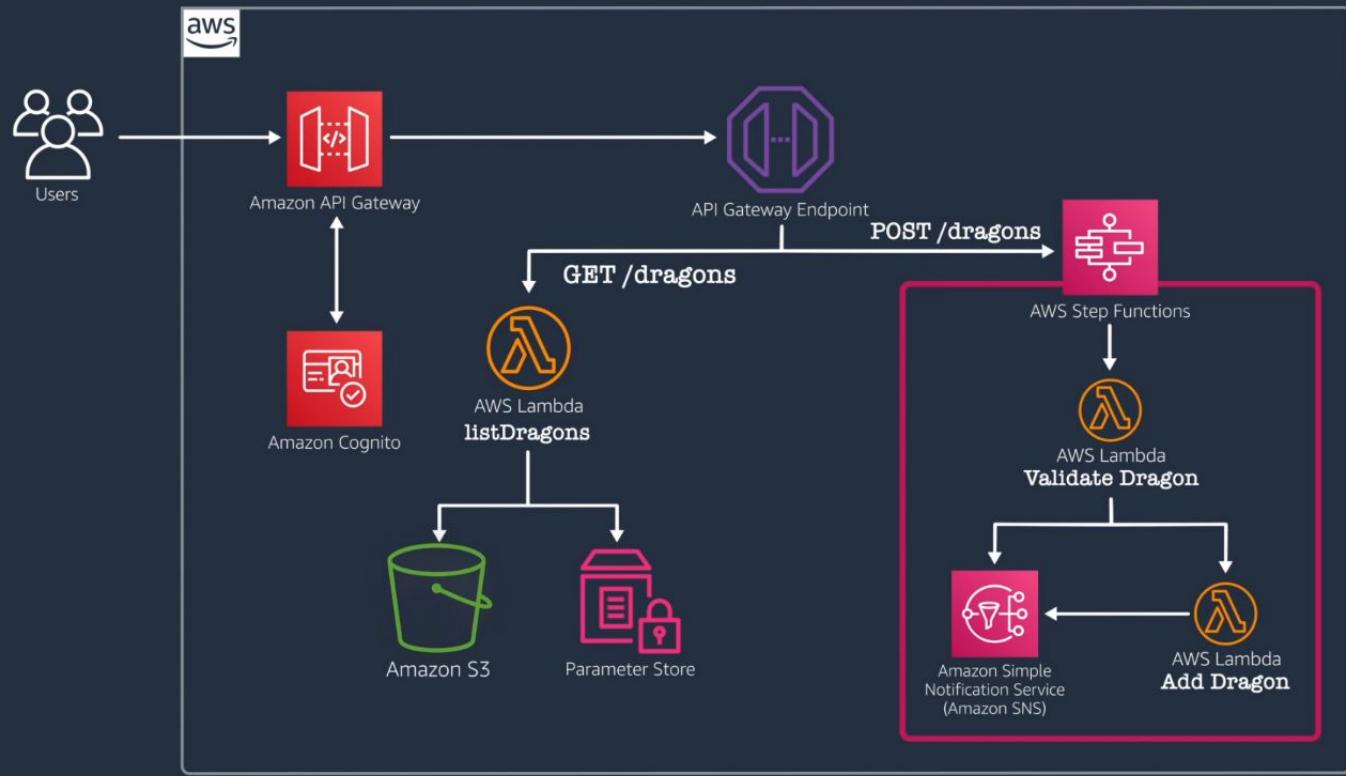
There are two types of permissions Lambda functions have:

Execution permissions and Resource-based policies

First, let's explore Execution permissions



Recap: we are going to have a Lambda function that allows people to get a list of all dragons.



That data is located in an S3 bucket. This means that the Lambda function that queries the data will need access to read that S3 bucket.

Just because the S3 bucket and Lambda function both live in this AWS account, it doesn't mean they are allowed to communicate at all.

We need to explicitly allow all AWS services to communicate with each other, using **role-based access**.

Every API call made in AWS
must be **signed** and **authenticated** before it is allowed to go through,
and we need an **AWS access key ID** and an **AWS secret key** to sign the API call.

We need to rely on **role-based access**,
which will retrieve **temporary credentials** for the Lambda function to use
for a **configurable** amount of time.

In order to do that, you create an IAM role
that contains the permissions your Lambda will need,
associate that role with your Lambda function.

From there, when your code runs,
the AWS SDK will use the credentials from the role
to make authenticated API calls to other AWS services.

Lambda functions need basic execution permissions as well
in order to do things like posting logs to CloudWatch Logs
or sending tracing data to AWS X-Ray.

AWS has a managed policy used for this:

[AWSLambdaBasicExecutionRole](#)

Ok, your execution permissions define what your Lambda has
the authority to do once it's been invoked.

but who can actually invoke your Lambda function in the first place?

Lambda functions have **triggers**.

A trigger is generally another AWS service, account, or AWS entity that invokes your Lambda function under certain event conditions.

Not just any service or entity is allowed to invoke or interact with your Lambda.
Lambda functions **resource-based policies** to control that.

Summary of this discussion:

Execution permissions define what your Lambda function can do.

Resource-based policies define what can invoke or manage your Lambda function.

Lambda functions Invoking

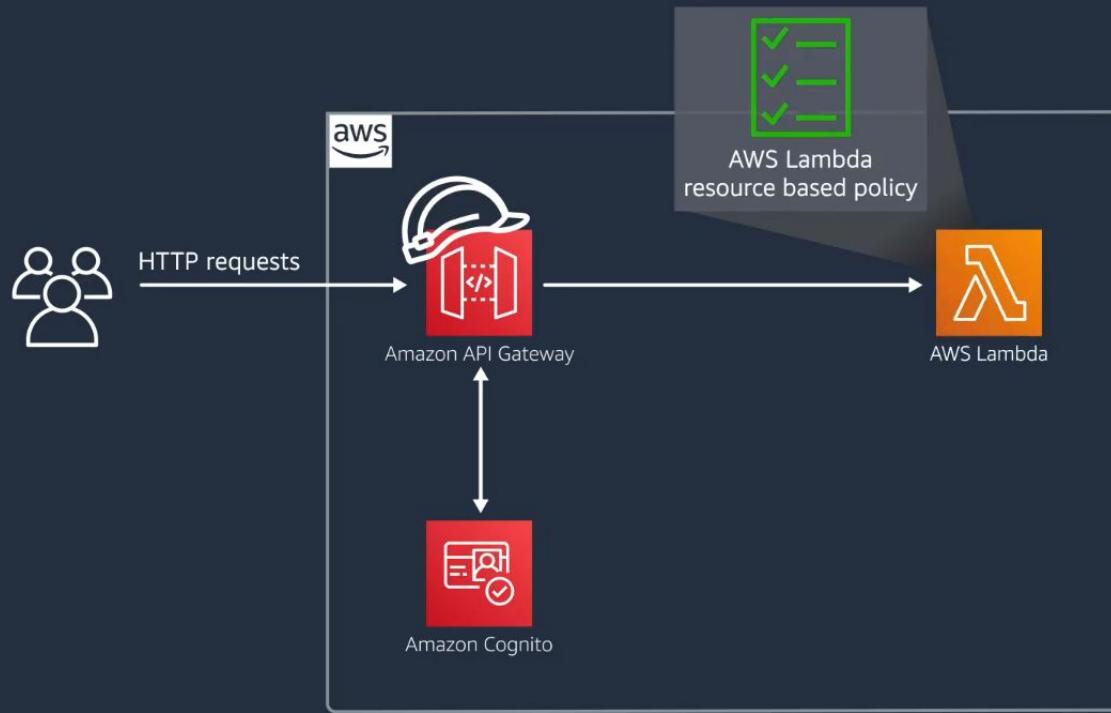
You can configure one or multiple triggers to invoke your function in response to life cycle events, external requests, or on a schedule.

e.g. if a user uploads an object to a bucket,
you can trigger a Lambda function to process that object.

e.g. when someone makes an HTTP request to an API,
a Lambda function is invoked.

Each trigger you configure for your Lambda function acts as a client, invoking your function independently.

Each event that Lambda passes to your function only has data from one client or trigger.



e.g. An HTTP request comes to API Gateway. That request is authenticated and validated. Then API Gateway uses an IAM role to invoke the Lambda function. Lambda then checks the resource-based policy associated with the Lambda function. And if the trigger is allowed to invoke the function, it will execute the Lambda.

In this case, API Gateway is pushing an event to the Lambda function.

Because of this direct push,
the resource-based policy must allow the invocation explicitly.

This is what we call the **push model**.

There are other types of triggers that Lambda **pulls** events from event sources and then invokes your Lambda function.

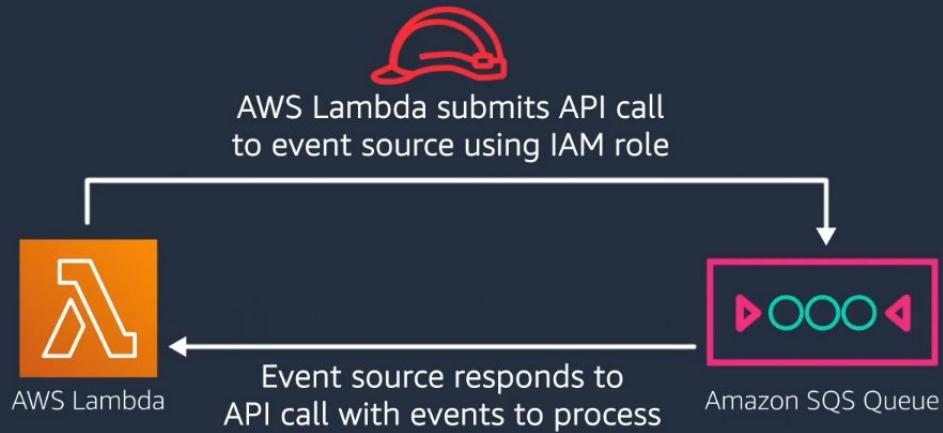


AWS Lambda



Amazon SQS Queue

Let's say you want to use Lambda to process messages from an SQS queue.



Messages get placed into the queue and the Lambda function is executed to process batches of these messages automatically.

Distinction between Lambda function and the Lambda service is important.

Your Lambda function is your code.

The Lambda service is the platform you are running your code on top of.

If you want to use the pull model for events sources like SQS, you don't need to write code to pull the queue in your Lambda function.

Instead, you configure the trigger to be SQS for the Lambda function and create something called an **Event Source Mapping**.

This event source mapping allows you to configure how Lambda pulls the queue and delivers messages to be processed by the Lambda function in batches.

There are other services besides SQS that follow this pull model,
like Amazon Kinesis Streams or Amazon DynamoDB Streams.

Kinesis and DynamoDB don't directly invoke the Lambda function.

Instead, messages are placed into streams,
and then Lambda pulls the stream and delivers the messages to your Lambda function,
just like it does for SQS.

This pull model affects the permissions required for your Lambda function.

In order for your Lambda to pull from these events sources for you, you must include the necessary permissions in your execution permission, or the role you associate with your Lambda function.

Summary

We have the push model and the pull model for triggers with AWS Lambda.

Triggers that push to Lambda require that
the resource-based policy allow that communication.

Whereas triggers use event source mappings require
the execution permissions, or IAM role, to contain the appropriate allowed API calls.

Lambda Execution Context Reuse

what happens to your code, once the micro VM is done executing your code?

what if you had some database connections established
from your code in Lambda at that time?

Let's discuss the concept of **cold starts** and **execution context reuse**.

Micro VM

When your Lambda function gets invoked, it spins up a new virtual machine.

Micro VM

Code

```
import libraries  
global variables definition  
...  
  
handler(...) {  
    db = new DB connection  
    do some work with db  
}
```

Then, it loads the code.

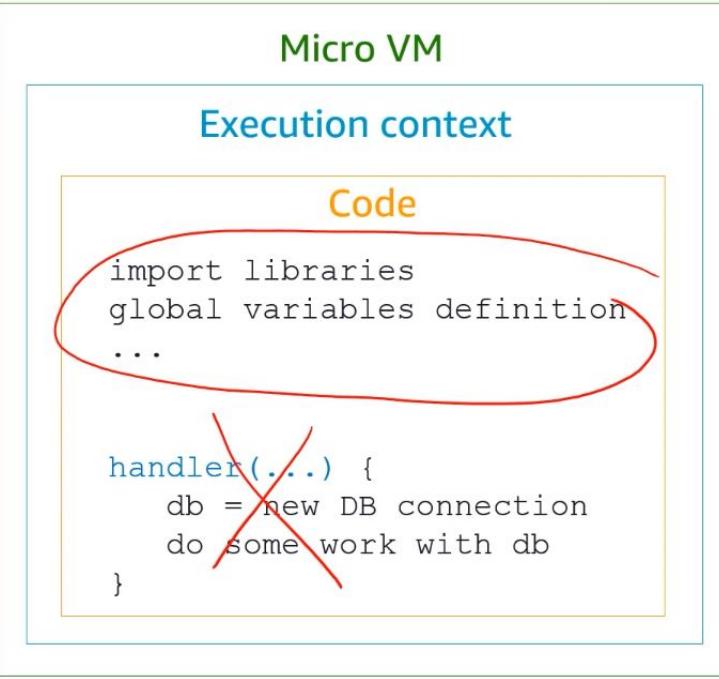
Micro VM

Code

```
import libraries  
global variables definition  
...
```

```
handler(...) {  
    db = new DB connection  
    do some work with db  
}
```

It executes everything in that code except the handler itself.



This is essentially bootstrapping the execution context. This execution context needs to be ready to invoke your handler itself.

That adds latency to invocation.

The term used to define the time spent doing this bootstrap portion, is called,

Cold Start

Does that mean that every time we execute the code,
it has to go through that bootstrap?

It depends.

If it's the first time this Lambda function is executed,
or if it was just modified, then yes.

However, once your code finishes what it has to do,
AWS Lambda freezes the execution context to be ready for **reuse**.
Currently, it keeps the execution context for 15 minutes.

Micro VM

Execution context

Code

```
import libraries
global variables definition
db = new DB connection
...
handler(...) {
    db = new DB connection
    do some work with db
}
```

what if we establish the database connection outside of the handler?

Remember, you need to develop your code in Lambda assuming that it's stateless.

However, with context reuse, there may be a state!

ex., check to see if that database connection was established first.

And if it was, you are in a **context reuse** scenario.

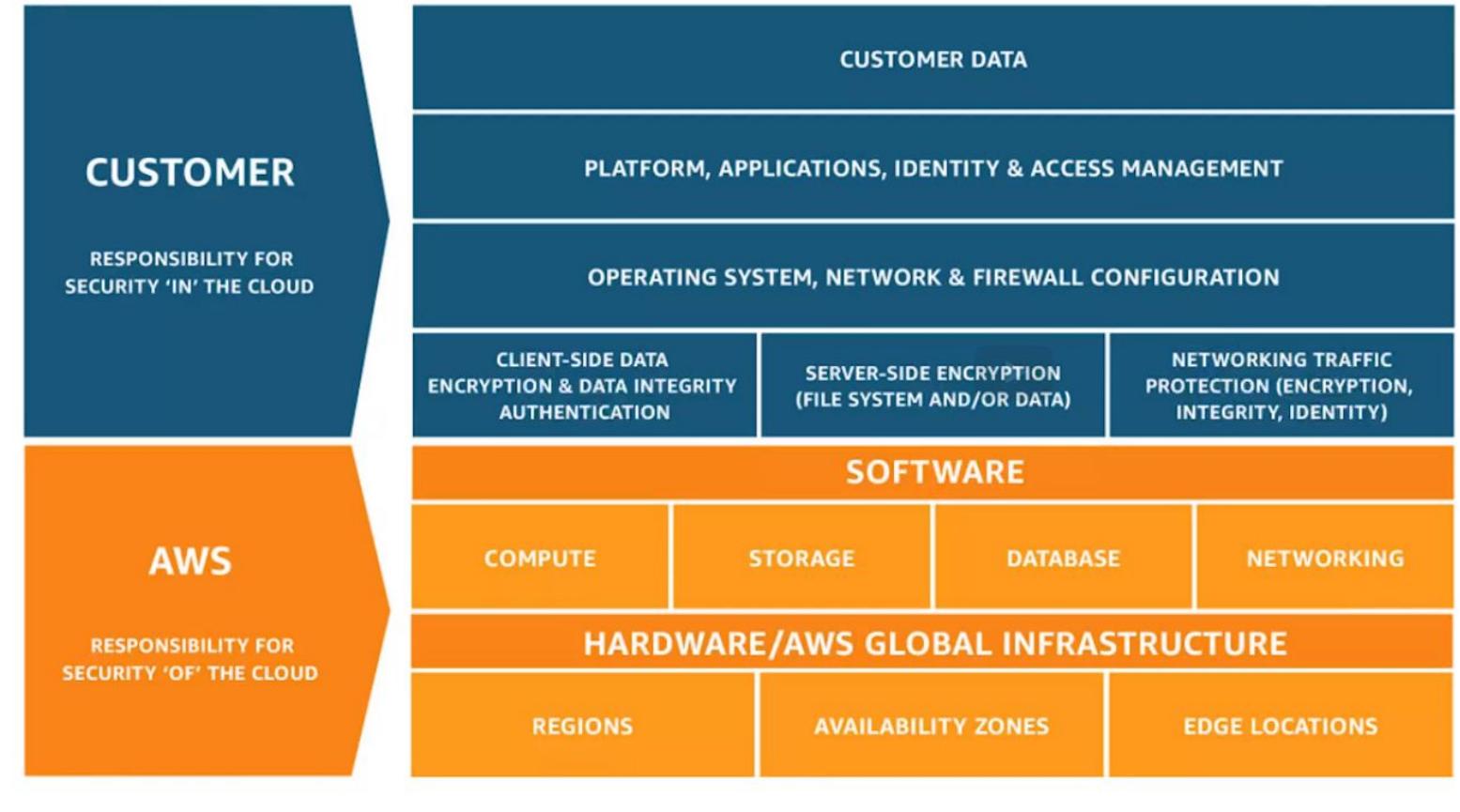
Even though Lambda functions shall be treated as stateless,
check if there is a state that was saved using the execution context reuse feature.

Context reuse will minimize costs and cold starts.

for example: loading a file in memory or making an API call that could be cached.

And second, if you don't want any cold start,
then you can use the provision concurrency feature (has cost).

Compliance with AWS Lambda



AWS shared responsibility model

Per the AWS shared responsibility model
it is **your responsibility** to **protect any personal data**,
whether from you or your customers, that you've put in the AWS cloud.

Using multi-factor authentication, logging, and taking advantage of the various AWS encryption solutions are all very helpful in protecting your data.

For encryption in transit,
Lambda API endpoints only support secure connections over HTTPS.

For encryption at rest, you can use environment variables to store secrets securely for use with Lambda functions.

Lambda always **encrypts environment variables** at rest.

Additionally, you can use features such as
key configuration and **encryption helpers**
to customize how environment variables are encrypted.

Key configurations work on a per function basis and allow you to use an encryption key that you create and manage in the [AWS Key Management Service](#).

Encryption helpers lets you encrypt environment variable values **client-side** before sending them to Lambda.

Asynchronous vs Synchronous Responses

Recap

Different types of triggers for Lambda can either use the **push** or **pull** model.

In the **pull model**, Lambda (the services) will trigger your Lambda function based on data it finds in a queue or data stream.

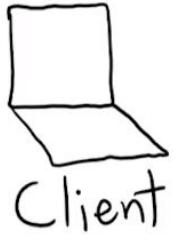
In the **push model**, the Lambda function is being invoked directly.

In this section, we cover the push model.

Invoking a Lambda function can be done with
either a **synchronous** or **asynchronous** call.

It depends, if we want to see the response from the code or not.

Let's take an example of a synchronous call.



Client

API
GW



Lambda

Client / API Gateway / Lambda



GET /dragons → API
GW



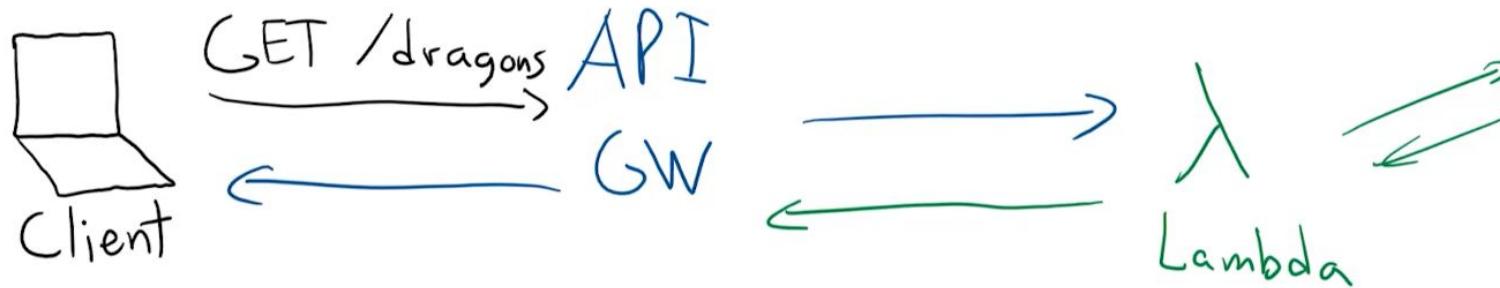
A client sends a request to API Gateway. Let's say a Get/dragons request



then API Gateway will invoke a Lambda function that it has



which the lambda function will execute some code. For ex. pull the list of dragons from somewhere (e.g. a s3 bucket)



Then it return data back towards API Gateway, which will then send that data back towards the client.

This is the default type of integration from API Gateway.

Because the desired behavior for most rest API operations is to see a result.

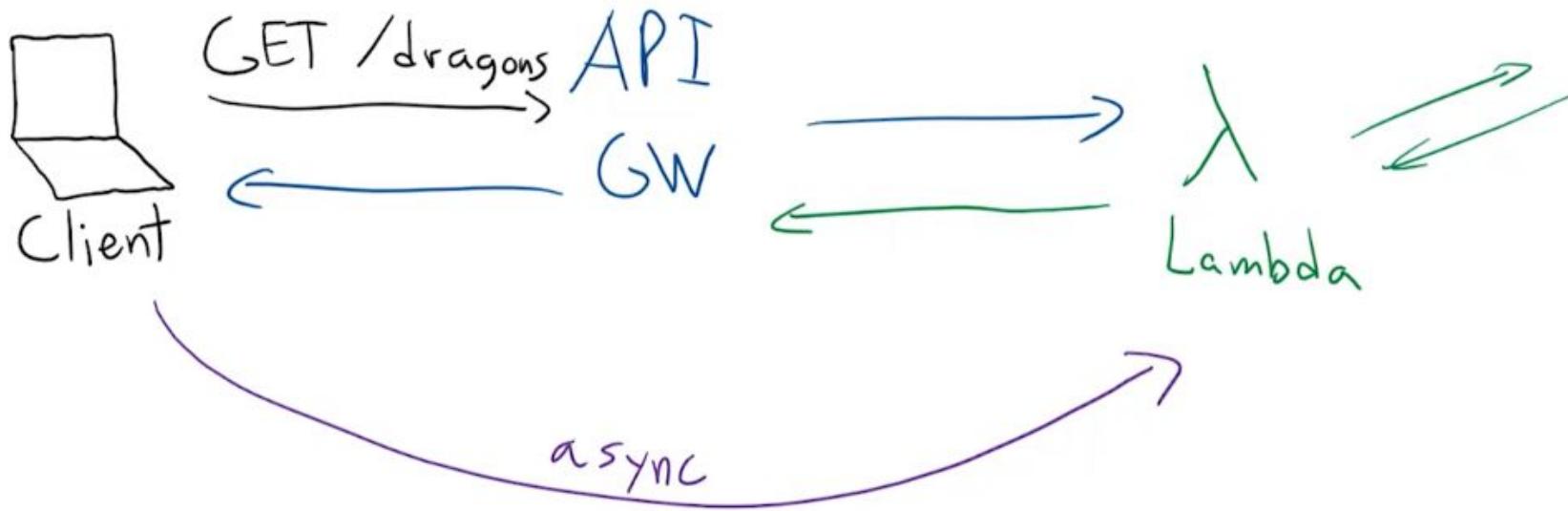
In a synchronous call,

the client is leaving its connection open and is eagerly waiting for reply back.

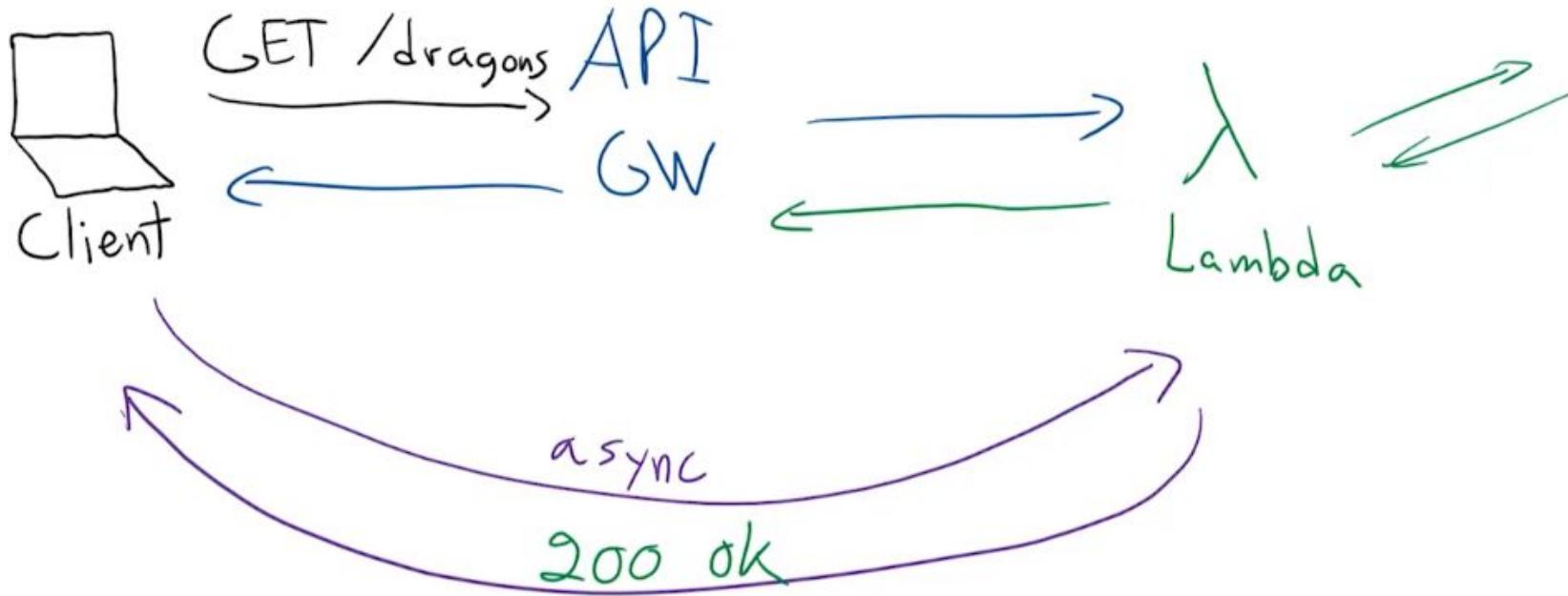
What if the client didn't care about the response?

Or to invoke a Lambda function for a batch job operation or a long latency operation.

That's what an **asynchronous call** to a Lambda function does.



an asynchronous call



Lambda Service returns 200. it means: "I've received your event for me to start a Lambda function."
But the client don't receive any other data.

Many services of AWS invoke Lambda function in an asynchronous way.

That's a great way to **scalability**.

There is one major issue with asynchronous calls.

What is if there was an error during the execution?

This is where the concept of **Retries in Lambda** becomes important to understand.

Functions (1)



Actions ▾

Create func



<

1

>

Function name	Description	Runtime	Code size	Last modified
hello-world		Node.js 12.x	304 bytes	3 hours ago

A "hello world" function in the lambda Function



hello-world

Throttle

Qualifiers ▾

Actions ▾

Select a test event ▾

Test



Qualifier



Type



Provisioned concurrency



Status

Details

No configurations

Add configuration

Asynchronous invocation Info

Edi

Maximum age of event
6 h 0 min 0 secRetry attempts
2Dead-letter queue service
None

Database proxies

Add database prox

Proxy identifier

Status

Engine compatibility

IAM Authentication

No proxies

You don't have any proxies.

Add database proxy

In the settings, you can find the asynchronous invocation setting box



Edit asynchronous configuration

Retries Info

Maximum age of event

The maximum amount of time to keep unprocessed events in the queue.

 h min sec

Retry attempts

The maximum number of times to retry when the function returns an error.

 ▼

Dead-letter queue Info

Dead-letter queue service

You can send unprocessed events from an asynchronous invocation to an Amazon SQS queue or an Amazon SNS topic.

 ▼CancelSave

A Lambda function's asynchronous configuration

When you invoke a Lambda function asynchronously, incoming events are placed in the queue before being sent to the actual function.

If the function returns an error, Lambda (the service) retries your Lambda function up to a configurable Retry Attempt setting (two by default).

If the function is troubled or Lambda returns an error
the event is kept in the queue for up to the maximum age of an event (default 6h).

This is a very important distinction between synchronous and asynchronous.

With an asynchronous invocation,
if this Retry Attempt settings is set to one or two,
your Lambda code could get executed more than once if there is an error.

That's why you need to make sure that your code is **idempotent**
so that if it gets executed more than once it produces the same result.

If it's **not idempotent** then change that maximum Retry setting and make it a zero.

If the Maximum Retry number or maximum age is reached,
then you can configure a dead letter queue
either to an [Amazon Simple Queue Service](#), or an [Amazon Simple Notification Service](#).
We will discuss these two services later in the course.

There is another new option instead of dead-letter queue.

Lambda destination



Services ▾

Resource Groups ▾



buildingmodernapps @ 3022-1... ▾ Central ▾



Lambda > Functions > hello-world

ARN - arn:aws:lambda:ca-central-1:302211264422:function:hell

hello-world

[Throttle](#) [Qualifiers ▾](#) [Actions ▾](#) [Select a test event ▾](#) [Test](#) [?](#)[Configuration](#) [Permissions](#) [Monitoring](#)

▼ Designer

[+ Add trigger](#)[+ Add destination](#)

Function code Info

[Actions](#)[File](#) [Edit](#) [Find](#) [View](#) [Go](#) [Tools](#) [Window](#)[Save](#) [Test](#) ▾[Feedback](#) [Report a bug](#)© 2008–2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy](#)

the Lambda Function dashboard



Services

Resource Groups



buildingmodernapps @ 3022-1... Central

Lambda > Functions > hello-world

ARN - arn:aws:lambda:ca-central-1:302211264422:function:hello

hello-world

Throttle

Qualifiers ▾

Actions ▾

Select a test event ▾

Test

S...

Configuration

Permissions

Monitoring

▼ Designer

Function code [Info](#)

Actions ▾

File Edit Find View Go Tools Window

Save Test ▾

trigger -> lambda function -> destination

Add destination

Destination configuration

Send invocation records to a destination when your function is invoked asynchronously, or if your function processes records from a stream.

Source

The type of invocation that maps to the destination.

- Asynchronous invocation
- Stream invocation

Condition

The condition for using the destination.

- On failure
- On success

Destination type

An SQS queue, SNS topic, Lambda function, or EventBridge event bus.

SNS topic



Destination



Cancel

Save

add destination setting



Services ▾

Resource Groups ▾



buildingmodernapps @ 3022-1...

[Lambda](#) > [Functions](#) > [hello-world](#) > Add destination

Add destination

Destination configuration
Send invocation records to a destination when your function is invoked asynchronously, or if your function processes records from a stream.

Source
The type of invocation that maps to the destination.

Asynchronous invocation ✓
 Stream invocation ✓

Condition
The condition for using the destination.

On failure ✓
 On success ✓

Destination type
An SQS queue, SNS topic, Lambda function, or EventBridge event bus.

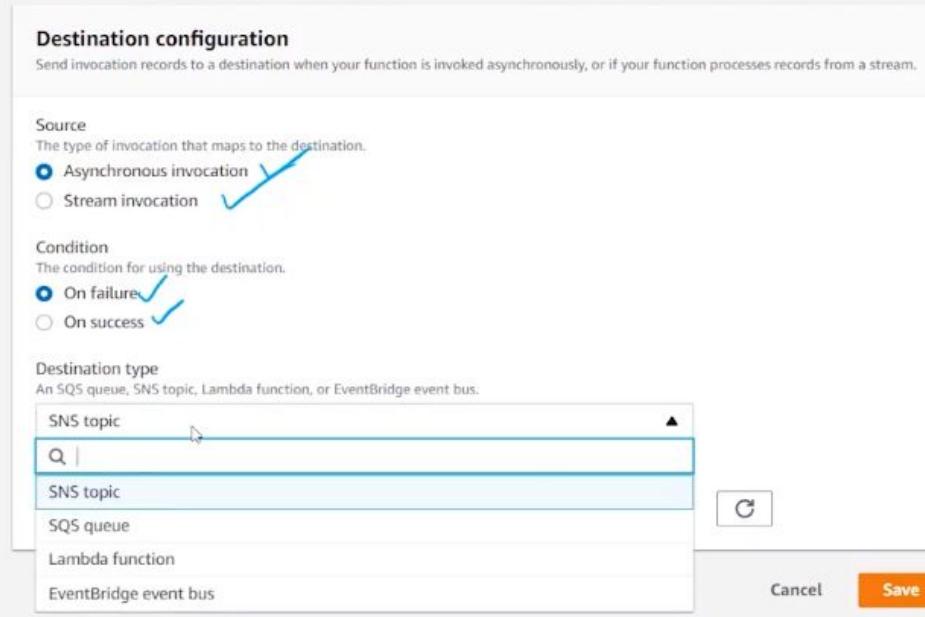
SNS topic

SNS topic

SQS queue

Lambda function

EventBridge event bus



You can use another Lambda Function as the destination

invoking another Lambda function as a destination!

This is a great way to decouple microservices.

We are able to execute something else
after the code in a Lambda Function executed successfully or not.

Useful resources:

[Using AWS Lambda with other services](#)

[Lambda event source mappings](#)

[Set up asynchronous invocation of the backend Lambda function - Amazon API Gateway](#)

[Security in AWS Lambda](#)

Creating an AWS Lambda Function

Note that all codes using in this section
is already uploaded in the Moodle as DemoCode.zip file

We're going to build out the back end for our dragon API.

We need a Lambda function for our get method on the dragon resource.

Once we create this Lambda function,
we will change our API in API Gateway to trigger the newly created Lambda function.

Then we can test it out.

The other functionality is the POST method to the dragon resource.

This will need a few different Lambda functions that
will be orchestrated by a tool called AWS Step Functions.

We will talk about it later.

A handler for a Python function generally looks like this:

```
def handler_name (event, context)  
    # application logic  
    return some_value
```

Event is the information coming in from a request.
in our case information coming from API Gateway.

This **event object** will include the payload of the request,
as well as any query parameters that might be on that request

In our dragon example, we will be using query parameters
to filter dragons based on family or dragon name.

The **context object** is passing information about the specific invocation of your Lambda function.

such as:

the AWS request ID, Lambda version that was invoked, ARN of the invoked function, information about execution environment (like the amount of allocated memory), identity information (potentially coming from Cognito), etc.

The **event** is the information on the request.

The **context** is the information about the invocation or the execution.

return:

You can optionally return a value
depending on the type of invocation that you chose for your Lambda.

If we wanted to have a request response **synchronous invocation**,
we need to return valid JSON.

If the return value cannot be serialized as JSON, it will cause an error.

If we choose an **asynchronous** invocation type,
the return value will be discarded.

If there is an error, the function invocation would be retried.

In the case of the list, dragons function,
we want it to be synchronous, and to return a list of dragons.

The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes tabs for 'AWS Management Console' and 'BuildingModernAppsPython'. The address bar shows the URL 'console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558'. The menu bar has options like File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run. On the left, there's an 'Environment' sidebar with a file tree containing 'BuildingModernAppsPython' (with 'SDK Exploration' expanded), 'add-dragon', 'list-dragons' (containing 'Commands.txt', 'listDragons.py'), and 'validate-dragon'. There are also 'app.py', 'client.py', and 'resource.py' files under 'SDK Exploration'. A 'README.md' file is also listed. The main workspace shows the 'app.py' file with the following code:

```
import boto3
s3 = boto3.resource('s3', 'us-east-1').meta.client
ssm = boto3.client('ssm', 'us-east-1')
bucket_name = ssm.get_parameter(Name='dragon_data_bucket_name', WithDecryption=False)['Parameter']['Value']
file_name = ssm.get_parameter(Name='dragon_data_file_name', WithDecryption=False)['Parameter']['Value']

def listDragons():
    expression = "select * from s3object s"
    result = s3.select_object_content(
        Bucket=bucket_name,
        Key=file_name,
        ExpressionType='SQL',
        Expression=expression,
        InputSerialization={'JSON': {'Type': 'Document'}},
        OutputSerialization={'JSON': {}}
    )
```

The bottom left shows a terminal window titled 'bash - ip-172-31' with the command 'buildingmodernapps:~/environment \$'. The bottom right of the code editor shows '1:1 Python Spaces: 4'.

Cloud9 IDE. we have the code we went over earlier in the course in the app.py code.

```
import boto3

s3 = boto3.resource('s3', 'us-east-1').meta.client
ssm = boto3.client('ssm', 'us-east-1')
bucket_name = ssm.get_parameter( Name='dragon_data_bucket_name',WithDecryption=False)['Parameter']['Value']
file_name = ssm.get_parameter( Name='dragon_data_file_name',WithDecryption=False)['Parameter']['Value']

def listDragons():

    expression = "select * from s3object s"

    result = s3.select_object_content(
        Bucket=bucket_name,
        Key=file_name,
        ExpressionType='SQL',
        Expression=expression,
        InputSerialization={'JSON': {'Type': 'Document'}},
        OutputSerialization={'JSON': {}}
    )

    for event in result['Payload']:
        if 'Records' in event:
            print(event['Records']['Payload'].decode('utf-8'))


listDragons()
```

recap from previous lectures.

We want to change this code from a regular Python script
into something that can run inside of a Lambda function.

The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes tabs for 'AWS Management Console' and 'BuildingModernAppsPython'. The main workspace has two tabs open: 'app.py' and 'listDragons.py'. The left sidebar displays a file tree with projects like 'BuildingModernAppsPython', 'add-dragon', 'list-dragons', 'Commands.txt', 'SDK Exploration', 'app.py', 'client.py', 'resource.py', 'validate-dragon', and 'README.md'. The 'listDragons.py' tab contains the following Python code:

```
1 import boto3
2 import json
3
4 s3 = boto3.client('s3', 'us-east-1')
5 ssm = boto3.client('ssm', 'us-east-1')
6 bucket_name = ssm.get_parameter( Name='dragon_data_bucket_name', WithDecryption=False)['Parameter']['Value']
7 file_name = ssm.get_parameter( Name='dragon_data_file_name', WithDecryption=False)['Parameter']['Value']
8
9 def listDragons(event, context):
10
11     expression = "select * from s3object s"
12
13     if 'queryStringParameters' in event and event['queryStringParameters'] is not None:
14         if 'dragonName' in event['queryStringParameters']:
15             expression = "select * from S3Object[*][*] s where s.dragon_name_str = '" + event["queryStringParameters"]['dragonName']
16         if 'family' in event['queryStringParameters']:
17             expression = "select * from S3Object[*][*] s where s.family_str = '" + event["queryStringParameters"]['family'] +
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
209
210
211
212
213
214
215
216
217
218
219
219
220
221
222
223
224
225
226
227
227
228
229
229
230
231
232
233
234
235
236
237
237
238
239
239
240
241
242
243
244
245
245
246
247
247
248
248
249
249
250
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
150
```

```
import boto3
import json
from aws_xray_sdk.core import patch_all
patch_all()

s3 = boto3.client('s3','us-east-1')
ssm = boto3.client('ssm', 'us-east-1')
bucket_name = ssm.get_parameter( Name='dragon_data_bucket_name',WithDecryption=False)[ 'Parameter'][ 'Value']
file_name = ssm.get_parameter( Name='dragon_data_file_name',WithDecryption=False)[ 'Parameter'][ 'Value']

def listDragons(event, context):

    expression = "select * from s3object s"

    if 'queryStringParameters' in event and event['queryStringParameters'] is not None:
        if 'dragonName' in event['queryStringParameters']:
            expression = "select * from S3Object[*][*] s where s.dragon_name_str = '" + \
                         event["queryStringParameters"]['dragonName'] + "'"
        if 'family' in event['queryStringParameters']:
            expression = "select * from S3Object[*][*] s where s.family_str = '" + event["queryStringParameters"]['family'] + "'"

    result = s3.select_object_content(
        Bucket=bucket_name,
        Key=file_name,
        ExpressionType='SQL',
        Expression=expression,
        InputSerialization={'JSON': {'Type': 'Document'}},
        OutputSerialization={'JSON': {}}
    )

    records = ''
    for event in result['Payload']:
        if 'Records' in event:
            records = event['Records']['Payload'].decode('utf-8')

    return {
        "statusCode": 200,
        "body": json.dumps(records)
    }
```

```
import boto3
import json
from aws_xray_sdk.core import patch_all
patch_all()

s3 = boto3.client('s3', 'us-east-1')
ssm = boto3.client('ssm', 'us-east-1')
bucket_name = ssm.get_parameter(
    Name='dragon_data_bucket_name', WithDecryption=False)[ 'Parameter'][ 'Value']
file_name = ssm.get_parameter(
    Name='dragon_data_file_name', WithDecryption=False)[ 'Parameter'][ 'Value']

def listDragons(event, context):
```

Note that the variable are declared out of the function to take advantage of execution context reuse.
If you are using a database, like DynamoDB, for example, create the client for DynamoDB outside of the handler to take advantage of that execution context reuse.

```
def listDragons(event, context):

    expression = "select * from s3object s"

    if 'queryStringParameters' in event and event['queryStringParameters'] is not None:
        if 'dragonName' in event['queryStringParameters']:
            expression = "select * from S3Object[*][*] s where s.dragon_name_str = '" +
                event["queryStringParameters"]['dragonName'] + "'"
        if 'family' in event['queryStringParameters']:
            expression = "select * from S3Object[*][*] s where s.family_str = '" +
                event["queryStringParameters"]['family'] + "'"

    result = s3.select_object_content(
        Bucket=bucket_name,
        Key=file_name,
        ExpressionType='SQL',
        Expression=expression,
        InputSerialization={'JSON': {'Type': 'Document'}},
        OutputSerialization={'JSON': {}}
    )
```

filter what's coming back from S3 using expression variable

```
records = ''  
for event in result['Payload']:   
    if 'Records' in event:  
        records = event['Records']['Payload'].decode('utf-8')  
  
return {  
    "statusCode": 200,  
    "body": json.dumps(records)  
}
```

- print the records and return back valid json using python json library
- response is the body of a request

That's all that it takes to get the old code
to be turned into "Lambda code".

It's tweaking the original code to run inside a Lambda execution environment.

Then we need to package up this code
into the Lambda deployment package.

which for Python needs to be a zip,
containing the Lambda code as well as all of its dependencies.

list of commands that we need to run in order to create a Lambda function:

- Download dependencies into folder
- `sudo pip install --target ./list-dragons-package boto3`
- Zip up your code (with dependencies)
- `zip -r9 ${OLDPWD}/ pythonlistDragonsFunction.zip .`
- Add python script to zip
- `zip -g pythonlistDragonsFunction.zip listDragons.py`
- Create Lambda Function
- `aws lambda create-function --function-name list-dragons --runtime python3.6 --role <IAM ROLE ARN> --handler listDragons.listDragons --publish --zip-file fileb://pythonlistDragonsFunction.zip`
- Invoke Lambda Function
- `aws lambda invoke --function-name list-dragons output.txt`
- Update Lambda Code
- `aws lambda update-function-code --function-name list-dragons --zip-file fileb://pythonlistDragonsFunction.zip --publish`

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run Share

Environment

Go to Anything (⌘ P)

BuildingModernAppsPython - *
add-dragon
list-dragons
Commands.txt
listDragons.py
SDK Exploration
app.py
client.py
resource.py
validate-dragon
README.md

app.py x listDragons.py x Commands.txt x

```
1 Download dependencies into folder
2 sudo pip install --target ./list-dragons-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r ${OLDPWD}/ pythonlistDragonsFunction.zip .
6
7 Add python script to zip
8 zip -g pythonlistDragonsFunction.zip listDragons.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name list-dragons --runtime python3.6 --role arn:aws:iam::302211264422:role/ListDragonsFunction
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name list-dragons output.txt
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name list-dragons --zip-file file:///pythonlistDragonsFunction.zip --publish
```

(54 Bytes) 2:59 Text Spaces: 4

sudo - "ip-172-31" x Immediate x

```
buildingmodernapps:~/environment $ cd list-dragons/
buildingmodernapps:~/environment/list-dragons $ sudo pip install --target ./list-dragons-package boto3
Collecting boto3
  Downloading https://files.pythonhosted.org/packages/a8/70/608e4a4b328c574630c81f9c0fbfdea660ad8bf3f7184280793f502073c8/boto3-1.13.14-py2.py3-none-any.whl (128kB)
    100% |██████████| 133kB 5.9MB/s
Collecting botocore<1.17.0,>=1.16.14 (from boto3)
  Downloading https://files.pythonhosted.org/packages/d8/87/f0bc2300be3a4ca7f910478c2037e706313e169484a6b082d59783bd557e/botocore-1.16.14-py2.py3-none-any.whl (6.2MB)
```

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run Share

Environment

Q Go to Anything (% P)

app.py x listDragons.py x Commands.txt x

BuildingModernAppsPython - *
add-dragon
list-dragons
list-dragons-package
Commands.txt
listDragons.py
SDK Exploration
app.py
client.py
resource.py
validate-dragon
README.md

1 Download dependencies into folder
2 sudo pip install --target ./list-dragons-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonlistDragonsFunction.zip .
6
7 Add python script to zip
8 zip -g pythonlistDragonsFunction.zip listDragons.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name list-dragons --runtime python3.6 --role arn:aws:iam::302211264422:role/ListDragonsFunction
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name list-dragons output.txt
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name list-dragons --zip-file file:///pythonlistDragonsFunction.zip --publish
(51 Bytes) 5:55 Text Spaces: 4

bash - "ip-172-31 x Immediate x +

```
Collecting six>=1.5 (from python-dateutil<2.8.0,>=2.1>dateutil<1.17.0,>=1.10.14>dateutil)
  Using cached https://files.pythonhosted.org/packages/65/eb/1f97cb97bfc2390a276969c6fae16075da282f505
  8082d4cb10c6c5c1dba/six-1.14.0-py2.py3-none-any.whl
Installing collected packages: urllib3, six, python-dateutil, jmespath, docutils, botocore, s3transfer, boto3
Successfully installed boto3-1.13.14 botocore-1.16.14 docutils-0.15.2 jmespath-0.10.0 python-dateutil-2.8.1 s3transfer-0.3.3 six-1.14.0 urllib3-1.25.9
You are using pip version 9.0.3, however version 20.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

buildingmodernapps:~/environment/list-dragons \$ cd list-dragons-package/
buildingmodernapps:~/environment/list-dragons/list-dragons-package \$ zip -r9 \${OLDPWD}/ pythonlistDragonsFunction.zip .

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

Environment

BuildingModernAppsPython - * app.py listDragons.py Commands.txt

1 Download dependencies into folder
2 sudo pip install --target ./list-dragons-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonlistDragonsFunction.zip .
6
7 Add python script to zip
8 zip -g pythonlistDragonsFunction.zip listDragons.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name list-dragons --runtime python3.6 --role arn:aws:iam::302211264422:role/ListDragonsFunction
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name list-dragons output.txt
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name list-dragons --zip-file file:///pythonlistDragonsFunction.zip --publish
(52 Bytes) 8:57 Text Spaces: 4

bash - "ip-172-31 x Immediate x +

```
buildingmoderndrags:~/environment/list-dragons/list-dragons-package$ ls
botocore-1.16.14.dist-info/WHEEL
botocore-1.16.14.dist-info/metadata.json
botocore-1.16.14.dist-info/INSTALLER
botocore-1.16.14.dist-info/METADATA
botocore-1.16.14.dist-info/RECORD
botocore-1.16.14.dist-info/DESCRIPTION.rst
buildingmoderndrags:~/environment/list-dragons/list-dragons-package$ cd ..
buildingmoderndrags:~/environment/list-dragons$ zip -g pythonlistDragonsFunction.zip listDragons.py
      zip warning: pythonlistDragonsFunction.zip not found or empty
      adding: listDragons.py (deflated 61%)
buildingmoderndrags:~/environment/list-dragons$
```

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run Share

Environment

Go to Anything (% P)

app.py listDragons.py Commands.txt

BuildingModernAppsPython - *
add-dragon
list-dragons
list-dragons-package
Commands.txt
listDragons.py
pythonlistDragonsFunction.zip
SDK Exploration
app.py
client.py
resource.py
validate-dragon
README.md

1 download dependencies into folder
2 sudo pip install --target ./list-dragons-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonlistDragonsFunction.zip .
6
7 Add python script to zip
8 zip -g pythonlistDragonsFunction.zip listDragons.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name list-dragons --runtime python3.6 --role arn:aws:iam::302211264422:role/ListDragonsLambdaExecutionRole
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name list-dragons output.txt
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name list-dragons --zip-file fileb://pythonlistDragonsFunction.zip --publish

(243 Bytes) 12:1 Text Spaces: 4

python2.7 - "ip-1" Immediate

```
adding: listDragons.py (deflated 61%)
buildingmodernapps:~/environment/list-dragons $ aws lambda create-function --function-name list-dragons --runtime python3.6 --role arn:aws:iam::302211264422:role/ListDragonsLambdaExecutionRole --handler listDragons.listDragons --publish --zip-file fileb://pythonlistDragonsFunction.zip
{
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "3a4rmkR2pRfnz/yuEhIQsSSJjUV8tBLfcKcQytql+To=",
    "FunctionName": "list-dragons",
    "CodeSize": 720,
```

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

Environment

BuildingModernAppsPython - * app.py listDragons.py Commands.txt

1 download dependencies into folder
2 sudo pip install --target ./list-dragons-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonlistDragonsFunction.zip .
6
7 Add python script to zip
8 zip -g pythonlistDragonsFunction.zip listDragons.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name list-dragons --runtime python3.6 --role arn:aws:iam::302211264422:role/ListDragonsLambdaExecutionRole
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name list-dragons output.txt
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name list-dragons --zip-file file:///pythonlistDragonsFunction.zip --publish

(243 Bytes) 12:1 Text Spaces: 4

python2.7 - "ip-1" Immediate

```
functionName : list-dragons ,  
"CodeSize": 720,  
"RevisionId": "e514b3bb-6174-40a3-a0f5-f7d7bdfc0ca1",  
"MemorySize": 128,  
"FunctionArn": "arn:aws:lambda:us-east-1:302211264422:function:list-dragons",  
"Version": "3",  
"Role": "arn:aws:iam::302211264422:role/ListDragonsLambdaExecutionRole",  
"Timeout": 3,  
"LastModified": "2020-05-20T19:39:40.053+0000",  
"Handler": "listDragons.listDragons",  
"Runtime": "python3.6",  
"Description": ""
```

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run Share

Environment

Go to Anything (% P)

BuildingModernAppsPython - * app.py listDragons.py Commands.txt

1 Download dependencies into folder
2 sudo pip install --target ./list-dragons-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonlistDragonsFunction.zip .
6
7 Add python script to zip
8 zip -g pythonlistDragonsFunction.zip listDragons.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name list-dragons --runtime python3.6 --role arn:aws:iam::302211264422:role/ListDragonsFunction
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name list-dragons output.txt
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name list-dragons --zip-file file:///pythonlistDragonsFunction.zip --publish

14:52 Text Spaces: 4

python2.7 - "ip-1" Immediate

```
def handler(event, context):
    print("Received event: " + event)
    print("Context: " + context)
    return {
        "statusCode": 200,
        "body": "Hello from Lambda"
    }
```

buildingmodernapps:~/environment/list-dragons \$ aws lambda invoke --function-name list-dragons output.txt

```
{ "statusCode": 200, "body": "Hello from Lambda" }
```

buildingmodernapps:~/environment/list-dragons \$

AWS Management Console | BuildingModernAppsPython - | list-dragons - Lambda

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

Services Resource Groups M

Lambda > Functions > list-dragons

ARN - arn:aws:lambda:us-east-1:302211264422:function:list-dragons

Throttle Qualifiers Actions Select a test event Test Save

Configuration Permissions Monitoring

Designer

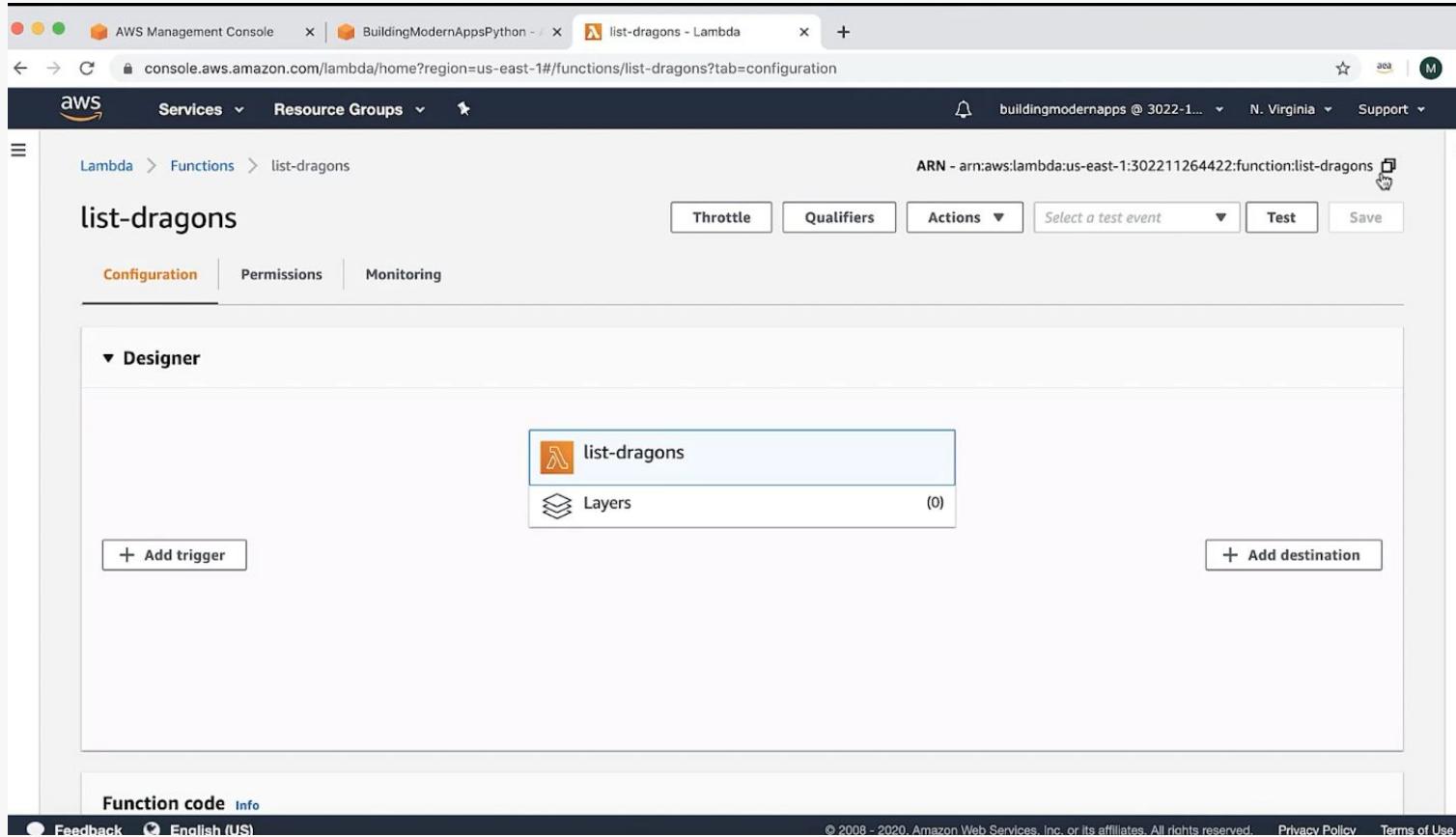
list-dragons

Layers (0)

+ Add trigger + Add destination

Function code Info

Feedback English (US) © 2008 – 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



Note that Lambda has a new ARN (Amazon Resource Name)

AWS Management Console | BuildingModernAppsPython - | list-dragons - Lambda

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

Services Resource Groups

buildingmodernapps @ 302211264422 N. Virginia Support

Lambda > Functions > list-dragons

ARN - arn:aws:lambda:us-east-1:302211264422:function:list-dragons

list-dragons

Throttle Qualifiers Actions ▾ Select a test event Test Save

Configuration Permissions Monitoring

Designer

list-dragons

Layers (0)

+ Add trigger + Add destination

Function code Info

Feedback English (US) © 2008 – 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

we can test AWS Lambda from AWS Lambda console

AWS Management Console | BuildingModernAppsPython - | list-dragons - Lambda

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

Services | Resources

Lambda > Functions > list-dragons

list-dragons

Configuration | Permissions

Designer

+ Add trigger

Function code | Info

Code | Handler | Lambda URL

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

Event template

hello-world

Event name

test

1 - {
2 |
3 }

+ Add destination

11264422:function:list-dragons

Test | Save

12-1... N. Virginia Support :

test configuration

AWS Management Console | BuildingModernAppsPython - | list-dragons - Lambda | console.aws.amazon.com/lambda/home?region=us-east-1#/functions/list-dragons?tab=configuration

Services Resource Groups M :

Lambda > Functions > list-dragons

ARN - arn:aws:lambda:us-east-1:302211264422:function:list-dragons

list-dragons

Throttle Qualifiers Actions ▾ test Test Save

Execution result: succeeded (logs) Details

Configuration Permissions Monitoring

Designer

list-dragons

Layers (0)

+ Add trigger + Add destination

Feedback English (US) © 2008 – 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

this is after the execution

AWS Management Console | BuildingModernAppsPython - | list-dragons - Lambda | console.aws.amazon.com/lambda/home?region=us-east-1#/functions/list-dragons?tab=configuration

Services Resource Groups

Lambda > Functions > list-dragons

ARN - arn:aws:lambda:us-east-1:302211264422:function:list-dragons

list-dragons

Throttle Qualifiers Actions ▾ test Test Save

Execution result: succeeded (logs)

Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
vegas\\\",\\\\\"location_country_str\\\":\\\\\"usa\\\\\",\\\\\"location_neighborhood_str\\\":\\\\\"westcliff  
dr\\\\\",\\\\\"location_state_str\\\":\\\\\"nevada\\\\\",\\\\\"protection_int\\\":3}, {\\\\\"damage_int\\\":8,\\\\\"description_str\\\":\\\\\"Midnight has  
a stealth attack mode. He can disappear and ghost for sneak  
attacks.\\\\\",\\\\\"dragon_name_str\\\":\\\\\"Midnight\\\\\",\\\\\"family_str\\\":\\\\\"black\\\\\",\\\\\"location_city_str\\\":\\\\\"cupertino\\\\\",\\\\\"loc  
ation_country_str\\\":\\\\\"usa\\\\\",\\\\\"location_neighborhood_str\\\":\\\\\"pear tree  
ln\\\\\",\\\\\"location_state_str\\\":\\\\\"california\\\\\",\\\\\"protection_int\\\":8}, {\\\\\"damage_int\\\":5,\\\\\"description_str\\\":\\\\\"Longlu is  
a healer in the water tribe. She heals warriors who have known  
battle.\\\\\",\\\\\"dragon_name_str\\\":\\\\\"Longlu\\\\\",\\\\\"family_str\\\":\\\\\"blue\\\\\",\\\\\"location_city_str\\\":\\\\\"hilo\\\\\",\\\\\"location_cu  
ntry_str\\\":\\\\\"usa\\\\\",\\\\\"location_neighborhood_str\\\":\\\\\"mona  
loop\\\\\",\\\\\"location_state_str\\\":\\\\\"hawaii\\\\\",\\\\\"protection_int\\\":8}, {\\\\\"damage_int\\\":4,\\\\\"description_str\\\":\\\\\"Dexler is a
```

Summary

Code SHA-256	Request ID
3a4rmkR2pRfnz/yuEhIQsSSJjUV8tBLfcKcQytql+To=	d6b21cdb-2b43-48b5-834a-a4ed7506c58c
Duration	Billed duration
199.81 ms	200 ms
Resources configured	Max memory used
128 MB	74 MB

Feedback English (US) © 2008–2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

execution log

Let's wire this up with API Gateway.

AWS Management Console | BuildingModernAppsPython - | list-dragons - Lambda | console.aws.amazon.com/lambda/home?region=us-east-1#/functions/list-dragons?tab=configuration

Services Resource Groups

Lambda > Functions > list-dragons

ARN - arn:aws:lambda:us-east-1:302211264422:function:list-dragons Copied

list-dragons

Throttle Qualifiers Actions ▾ test Test Save

Execution result: succeeded (logs)

Details

The area below shows the result returned by your function execution. [Learn more](#) about returning results from your function.

```
vegas\\\",\\\"location_country_str\\\":\\\"usa\\\",\\\"location_neighborhood_str\\\":\\\"westcliff  
dr\\\",\\\"location_state_str\\\":\\\"nevada\\\",\\\"protection_int\\\":3},\\\"damage_int\\\":8,\\\"description_str\\\":\\\"Midnight has  
a stealth attack mode. He can disappear and ghost for sneak  
attacks.\\\",\\\"dragon_name_str\\\":\\\"Midnight\\\",\\\"family_str\\\":\\\"black\\\",\\\"location_city_str\\\":\\\"cupertino\\\",\\\"loc  
ation_country_str\\\":\\\"usa\\\",\\\"location_neighborhood_str\\\":\\\"pear tree  
ln\\\",\\\"location_state_str\\\":\\\"california\\\",\\\"protection_int\\\":8},\\\"damage_int\\\":5,\\\"description_str\\\":\\\"Longlu is  
a healer in the water tribe. She heals warriors who have known  
battle.\\\",\\\"dragon_name_str\\\":\\\"Longlu\\\",\\\"family_str\\\":\\\"blue\\\",\\\"location_city_str\\\":\\\"hilo\\\",\\\"location_cu  
ntry_str\\\":\\\"usa\\\",\\\"location_neighborhood_str\\\":\\\"mona  
loop\\\",\\\"location_state_str\\\":\\\"hawaii\\\",\\\"protection_int\\\":8},\\\"damage_int\\\":4,\\\"description_str\\\":\\\"Dexler is a
```

Summary

Code SHA-256	Request ID
3a4rmkR2pRfnz/yuEhIQsSSJjUV8tBLfcKcQytql+To=	d6b21cdb-2b43-48b5-834a-a4ed7506c58c
Duration	Billed duration
199.81 ms	200 ms
Resources configured	Max memory used
128 MB	74 MB

Feedback English (US) © 2008–2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

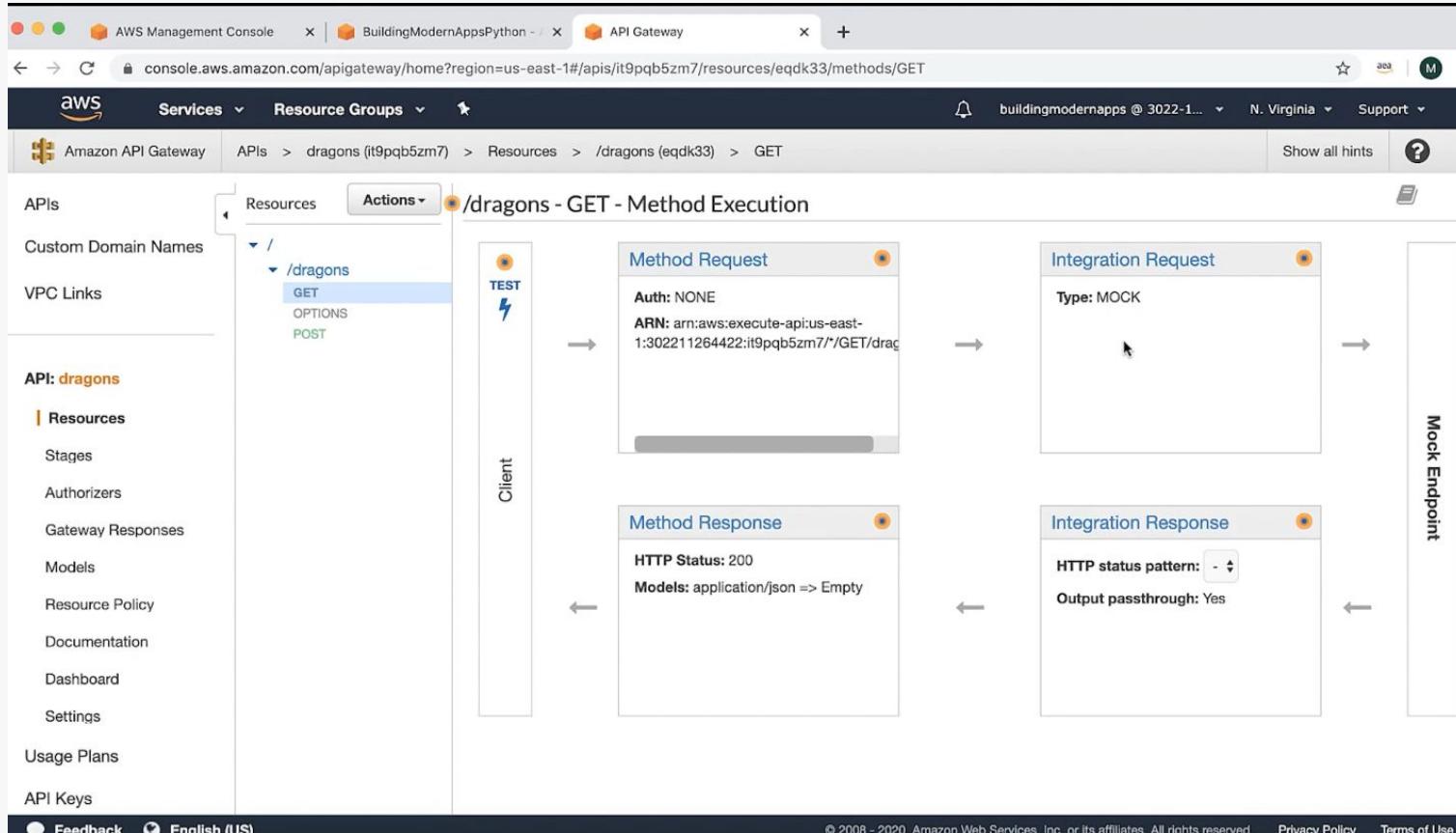
Copy the ARN

The screenshot shows the AWS Management Console with three tabs open: 'AWS Management Console', 'BuildingModernAppsPython -', and 'API Gateway'. The 'API Gateway' tab is active, displaying the 'APIs' list. The left sidebar shows navigation options: 'APIs', 'Custom domain names', and 'VPC links'. The main content area has a header 'APIs (1)' with a search bar, an 'Actions' dropdown, and a 'Create API' button. A table lists one API entry:

Name	Description	ID	Protocol	Endpoint type	Created
dragons		it9pqb5zm7	REST	Regional	2020-05-15

A cursor is hovering over the 'dragons' link in the Name column.

then navigate the API Gateway, and the dragons api



change the integration request from Mock to our lambda function

AWS Management Console | BuildingModernAppsPython | API Gateway

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/it9pqb5zm7/resources/eqdk33/methods/GET

Services Resource Groups

Amazon API Gateway APIs > dragons (it9pqb5zm7) > Resources > /dragons (eqdk33) > GET Show all hints ?

APIs Custom Domain Names VPC Links

API: dragons

Resources Stages Authorizers Gateway Responses Models Resource Policy Documentation Dashboard Settings Usage Plans API Keys

Actions Resources Actions

Method Execution /dragons - 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

Lambda Function

Use Default Timeout

Save

Mapping Templates

Feedback English (US)

© 2008–2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

The screenshot shows the AWS Management Console with the API Gateway service selected. The URL in the browser is `console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/it9pqb5zm7/resources/eqdk33/methods/GET`. The left sidebar shows the navigation path: APIs > dragons (it9pqb5zm7) > Resources > /dragons (eqdk33) > GET. The main content area is titled "Method Execution /dragons - GET - Integration Request". It displays the configuration for this specific method. The "Integration type" is set to "Lambda Function". The "Use Lambda Proxy integration" checkbox is checked. The "Lambda Region" dropdown is empty, and the "Lambda Function" input field is also empty. Below these fields are "Use Default Timeout" and "Save" buttons. A "Mapping Templates" section is visible at the bottom.

select use proxy integration. it takes the input and just passes it straight on back to the back end.

AWS Management Console | BuildingModernAppsPython | API Gateway

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/it9pqb5zm7/resources/eqdk33/methods/GET

Services Resource Groups

Amazon API Gateway APIs > dragons (it9pqb5zm7) > Resources > /dragons (eqdk33) > GET Show all hints ?

APIs / Resources Actions

Method Execution /dragons - 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 us-east-1

Lambda Function arn:aws:lambda:us-east-1:302211264422:function:list-dragons

Use Default Timeout list-dragons

Save

Feedback English (US)

© 2008–2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

paste the lambda function ARN here

AWS Management Console | BuildingModernAppsPython | API Gateway

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/it9pqb5zm7/resources/eqdk33/methods/GET

Services Resource Groups

Amazon API Gateway APIs > dragons (it9pqb5zm7) > Resources > /dragons (eqdk33) > GET Show all hints ?

APIs Resources Actions / Method Execution /dragons - 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 us-east-1 Lambda Function list-dragons

Use Default Timeout

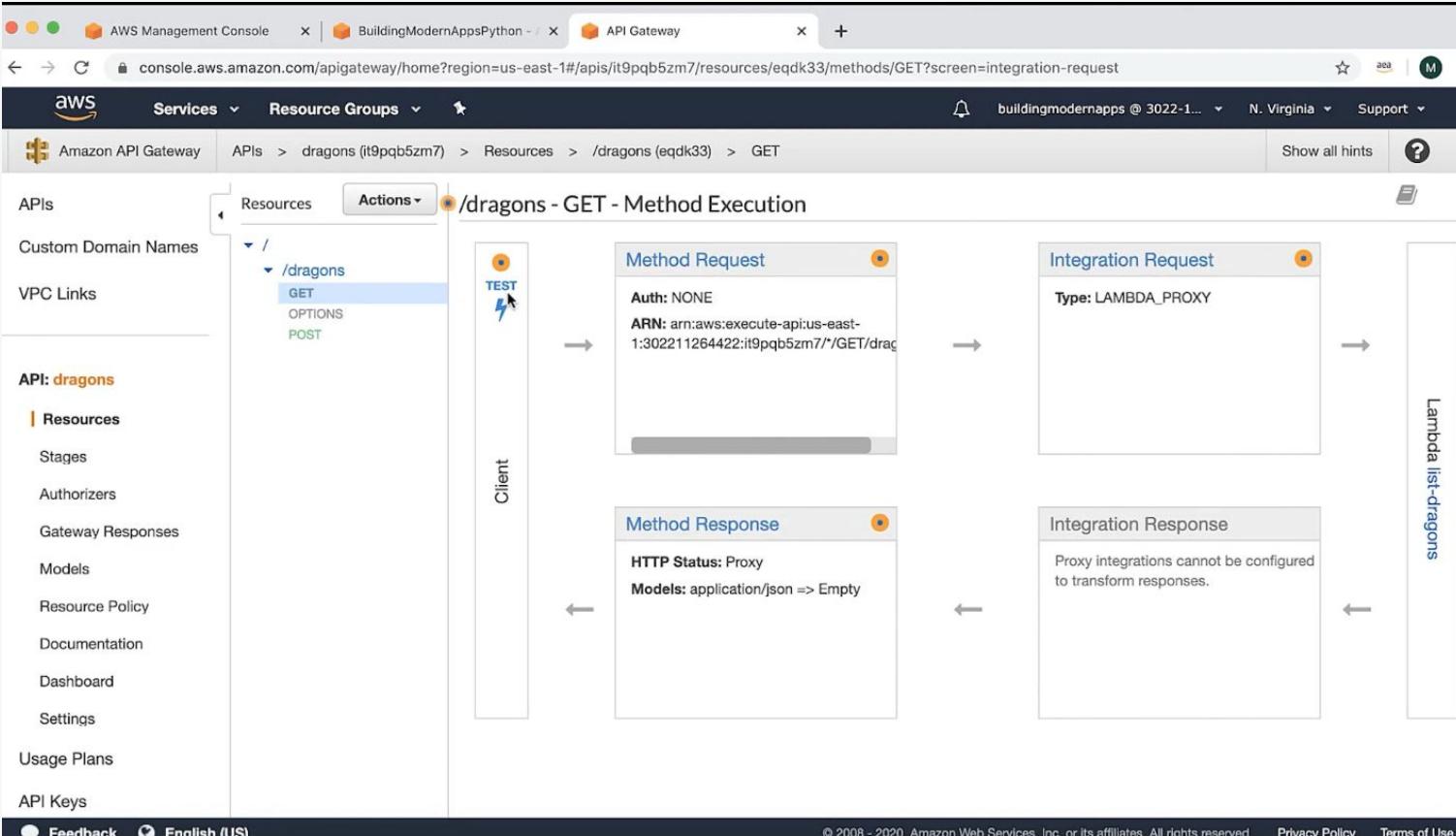
Mapping Templates

Save

Feedback English (US)

The screenshot shows the AWS Management Console with three tabs open: 'AWS Management Console', 'BuildingModernAppsPython - X', and 'API Gateway'. The 'API Gateway' tab is active, displaying the URL: 'console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/it9pqb5zm7/resources/eqdk33/methods/GET'. The left sidebar shows navigation options like 'Services', 'Resource Groups', 'APIs', 'Custom Domain Names', 'VPC Links', and a detailed section for the 'API: dragons' resource. The main content area is titled 'Add Permission to Lambda Function' and contains the message: 'You are about to give API Gateway permission to invoke your Lambda function: arn:aws:lambda:us-east-1:302211264422:function:list-dragons'. Below this, there are three radio button options: 'Mock', 'AWS Service', and 'VPC Link'. A large 'OK' button is visible at the bottom right of the dialog. In the background, the API configuration interface shows fields for 'Lambda Region' (set to 'us-east-1'), 'Lambda Function' ('list-dragons'), and 'Use Default Timeout' (checked). A 'Save' button is located at the bottom right of the main form.

here we give permission to AWS Gateway to invoke the Lambda function



AWS Management Console X BuildingModernAppsPython - X API Gateway X +

← → C console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/it9pqb5zm7/resources/eqdk33/methods/GET?screen=integration-request

Resources Stages Authorizers Gateway Responses Models Resource Policy Documentation Dashboard Settings Usage Plans API Keys Client Certificates Settings

Query Strings
{dragons}
param1=value1¶m2=value2

Headers
{dragons}
Use a colon (:) to separate header name and value, and new lines to declare multiple headers. e.g.
Accept:application/json.

Stage Variables
No stage variables exist for this method.

Client Certificate
No client certificates have been generated.

Request Body
Request Body is not supported for GET methods.

Test

"{\\"_1\":[{\\"damage_int\":9,\"description_str\":\"From the northern fire tribe, Atlas was born from the ashes of his fallen father in combat. He is fearless and does not fear battle.\",\"dragon_name_str\":\"Atlas\",\"family_str\":\"red\",\"location_city_str\":\"anchorage\",\"location_country_str\":\"usa\",\"location_neighborhood_str\":\"w fireweed ln\",\"location_state_str\":\"alaska\",\"protection_int\":7},{\"damage_int\":4,\"description_str\":\"Bahamethut is an immortal dragon. Cruel and ruthless, he comes from the undead realm and is a judge among his dragon tribe.\",\"dragon_name_str\":\"Bahamethut\",\"family_str\":\"black\",\"location_city_str\":\"las vegas\",\"location_country_str\":\"usa\",\"location_neighborhood_str\":\"shore breeze dr\",\"location_state_str\":\"nevada\",\"protection_int\":3},{\"damage_int\":6,\"description_str\":\"Frealeu has an ice breath that can freeze her enemies into a paralyzed state. She is from the souther water tribe.\",\"dragon_name_str\":\"Frealeu\",\"family_str\":\"blue\",\"location_city_str\":\"mesa\",\"location_country_str\":\"usa\",\"location_neighborhood_str\":\"e adobe st\",\"location_state_str\":\"arizona\",\"protection_int\":6},{\"damage_int\":2,\"description_str\":\"Padumo is a sage from the easter wood s. She practices healing medicine.\",\"dragon_name_str\":\"Padumo\",\"family_str\":\"green\",\"location_city_str\":\"sterling\",\"location_country_str\":\"usa\",\"location_neighborhood_str\":\"broadway\",\"location_state_str\":\"colorado\",\"protection_int\":4},{\"damage_int\":9,\"description_str\":\"Tagnaurak's breathe melts and refines precious gems. He covets money and loves shiny things that he obsessively collects.\",\"dragon_name_str\":\"Tagnaurak\",\"family_str\":\"red\",\"location_city_str\":\"las vegas\",\"location_country_str\":\"usa\",\"location_neighborhood_str\":\"westcliff dr\"},\"location_state_str\":\"nev

test

AWS Management Console X BuildingModernAppsPython - X API Gateway X +

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/it9pqb5zm7/resources/eqdk33/methods/GET?screen=integration-request

VPC Links

API: dragons

Resources

- Stages
- Authorizers
- Gateway Responses
- Models
- Resource Policy
- Documentation
- Dashboard
- Settings
- Usage Plans
- API Keys
- Client Certificates
- Settings

Path

No path parameters exist for this resource. You can define path parameters by using the syntax `{myPathParam}` in a resource path.

Query Strings

[dragons]

dragonName=Dexler

Headers

[dragons]

Use a colon (:) to separate header name and value, and new lines to declare multiple headers. e.g.
Accept:application/json.

Stage Variables

No stage variables exist for this method.

Client Certificate

No client certificates have been generated.

Request Body

Request Body is not supported for GET methods.

Request: /dragons?dragonName=Dexler

Status: 200

Latency: 234 ms

Response Body

```
{"\"damage_int\":4,\"description_str\":\"Dexler is a protector of the earth and forests. He is as green as the earth and burrows into the ground for protection and extra defense.\",\"dragon_name_str\":\"Dexler\",\"family_str\":\"green\",\"location_city_str\":\"lexington\",\"location_country_str\":\"usa\",\"location_neighborhood_str\":\"bellcastle rd\",\"location_state_str\":\"kentucky\",\"protection_int\":2}\n"
```

Response Headers

```
{"X-Amzn-Trace-Id":"Root=1-sec58922-1673d2a516274a23b838d22b;Sampled=0"}
```

Logs

```
Execution log for request e0710925-9087-428c-ae2a-063e4471f6c8
Wed May 20 19:46:42 UTC 2020 : Starting execution for request: e0710925-9087-428c-ae2a-063e4471f6c8
Wed May 20 19:46:42 UTC 2020 : HTTP Method: GET, Resource Path: /dragons
Wed May 20 19:46:42 UTC 2020 : Method request path: {}
Wed May 20 19:46:42 UTC 2020 : Method request query string: {dragonName=Dexler}
Wed May 20 19:46:42 UTC 2020 : Method request headers: {}
Wed May 20 19:46:42 UTC 2020 : Method request body before
```

Everything is fully set up for this one method.

We've created our first Lambda function.

This Lambda function is wired up.

API Gateway is set up as the trigger for that Lambda function.

We set up the API to be a Lambda proxy integration with that back end.

But we still have two more Lambda functions to create:
the validate dragon and add dragon functions using the POST method.

To report a new dragon successfully, we need to:

first, validate that the dragon is not already in the system

second, add the dragon to our JSON file being hosted in S3

Then once that's done,
the Step Functions will alert the user
that their dragon was added, or it was not added.

The screenshot shows the AWS Cloud9 IDE interface. The top navigation bar includes tabs for 'AWS Management Console' and 'BuildingModernAppsPython'. The main window displays a file named 'validateDragon.py' with the following Python code:

```
import boto3
import json

s3 = boto3.client('s3', 'us-east-1')
ssm = boto3.client('ssm', 'us-east-1')
bucket_name = ssm.get_parameter(Name='dragon_data_bucket_name', WithDecryption=False)['Parameter']['Value']
file_name = ssm.get_parameter(Name='dragon_data_file_name', WithDecryption=False)['Parameter']['Value']

def validate(event, context):
    result = s3.select_object_content(
        Bucket=bucket_name,
        Key=file_name,
        ExpressionType='SQL',
        Expression="select * from S3Object[*][*] s where s.dragon_name_str = '" + event['dragon_name_str'] + "'",
        InputSerialization={'JSON': {'Type': 'Document'}},
        OutputSerialization={'JSON': {}}
    )
```

The code uses the Boto3 library to interact with S3 and SSM. It retrieves the bucket name and file name from SSM parameters and then uses the S3 select object content operation to query the S3 bucket for objects where the dragon_name_str matches the value provided in the event payload.

The left sidebar shows the project environment structure, including folders like 'BuildingModernAppsPython', 'add-dragon', 'list-dragons', 'SDK Exploration', and 'validate-dragon', along with files such as 'Commands.txt', 'newDragonPayload.json', and 'validateDragon.py'. A 'README.md' file is also listed.

The bottom terminal window shows the command 'buildingmodernapps:~/environment/list-dragons \$'.

validate a dragon

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (% P)

validateDragon.py

Environment

BuildingModernAppsPython

- add-dragon
- Commands.txt
- newDragonPayload.json

list-dragons

SDK Exploration

validate-dragon

- Commands.txt
- duplicateDragonPayload.json
- newDragonPayload.json
- validateDragon.py

README.md

15 Expression="select * from S3Object[*][*] s where s.dragon_name_str = '" + event['dragon_name_str'] + "'",
16 InputSerialization={'JSON': {'Type': 'Document'}},
17 OutputSerialization={'JSON': {}}

18)

19

20 for records in result['Payload']:
21 if 'Records' in records:
22 | raise DragonValidationException("Duplicate dragon reported")
23 return 'Dragon Validated'

24

25 class DragonValidationException(Exception):
26 def __init__(self, *args):
27 if args:
28 self.message = args[0]
else:
30 self.message = None
31

32 def __str__(self):

20:38 Python Spaces: 4

bash - ip-172-31 x Immediate x +

buildingmodernapps:~/environment/list-dragons \$

The screenshot shows the AWS Management Console IDE interface. The top bar includes tabs for 'AWS Cloud9' and 'BuildingModernAppsPython'. The main area displays a Python script named 'validateDragon.py' with the following code:

```
20     for records in result['Payload']:
21         if 'Records' in records:
22             raise DragonValidationException("Duplicate Dragon reported")
23         return 'Dragon Validated'
24
25 class DragonValidationException(Exception):
26     def __init__(self, *args):
27         if args:
28             self.message = args[0]
29         else:
30             self.message = None
31
32     def __str__(self):
33         if self.message:
34             return 'DragonValidationException, {0}'.format(self.message)
35         else:
36             return 'DragonValidationException has been raised'
37
```

The code implements a custom exception class 'DragonValidationException' that raises an error if a record is found in the payload. The environment sidebar on the left lists project files: 'add-dragon', 'Commands.txt', 'newDragonPayload.json', 'list-dragons', 'SDK Exploration', 'validate-dragon', 'Commands.txt', 'duplicateDragonPayload.json', 'newDragonPayload.json', and 'validateDragon.py'. The bottom terminal window shows the command 'buildingmodernapps:~/environment/list-dragons \$'.

custom exception if there is a duplicated record

list of commands that we need to run in order to this Lambda function:

- Download dependencies into folder
`sudo pip install --target ./validate-dragon-package boto3`
- Zip up your code (with dependencies)
`zip -r9 ${OLDPWD}/ pythonValidateDragonFunction.zip .`
- Add python script to zip
`zip -g pythonValidateDragonFunction.zip validateDragon.py`
- Create Lambda Function
`aws lambda create-function --function-name validate-dragon --runtime python3.6 --role <IAM ROLE ARN> --handler validateDragon.validate --publish --zip-file fileb://pythonValidateDragonFunction.zip`
- Invoke Lambda Function
`aws lambda invoke --function-name validate-dragon output.txt --payload file://duplicateDragonPayload.json`
- Update Lambda Code
`aws lambda update-function-code --function-name validate-dragon --zip-file fileb://pythonValidateDragonFunction.zip --publish`

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (⌘ P)

validateDragon.p Commands.txt

BuildingModernAppsPython add-dragon Commands.txt newDragonPayload.json list-dragons SDK Exploration validate-dragon Commands.txt duplicateDragonPayload.json newDragonPayload.json validateDragon.py README.md

1 Download dependencies into folder
2 sudo pip install --target ./validate-dragon-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonValidateDragonFunction.zip .
6
7 Add python script to zip
8 zip -g pythonValidateDragonFunction.zip validateDragon.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name validate-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/Validat...
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name validate-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name validate-dragon --zip-file file:///pythonValidateDragonFunction.zip --publis...
(57 Bytes) 2:62 Text Spaces: 4

sudo - "ip-172-31" Immediate

buildingmodernapps:~/environment \$ cd validate-dragon/
buildingmodernapps:~/environment/validate-dragon \$ sudo pip install --target ./validate-dragon-package
boto3
Collecting boto3
Using cached https://files.pythonhosted.org/packages/a8/70/608e4a4b328c574630c81f9c0fbfdea660ad8bf3f
7184280793f502073c8/boto3-1.13.14-py2.py3-none-any.whl
Collecting jmespath<1.0.0,>=0.7.1 (from boto3)
Using cached https://files.pythonhosted.org/packages/07/cb/5f001272b6faeb23c1c9e0acc04d48eaaf5c862c1
7709d20e3469c6e0139/jmespath-0.10.0-py2.py3-none-any.whl
Collecting botocore<1.17.0,>=1.16.14 (from boto3)

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (⌘ P)

validateDragon.p Commands.txt

BuildingModernAppsPython add-dragon Commands.txt newDragonPayload.json list-dragons SDK Exploration validate-dragon validate-dragon-package Commands.txt duplicateDragonPayload.json newDragonPayload.json validateDragon.py README.md

1 Download dependencies into folder
2 sudo pip install --target ./validate-dragon-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonValidateDragonFunction.zip .
6
7 Add python script to zip
8 zip -g pythonValidateDragonFunction.zip validateDragon.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name validate-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/Validat...
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name validate-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name validate-dragon --zip-file file:///pythonValidateDragonFunction.zip --publis...
(53 Bytes) 5:5 Text Spaces: 4

bash - "ip-172-31-10-10" x Immediate x

Collecting six>=1.5.2 (from https://files.pythonhosted.org/packages/65/eb/1f97cb97bfc2390a276969c6fae16075da282f5058082d4cb10c6c5c1dba/six-1.14.0-py2.py3-none-any.whl)

Using cached https://files.pythonhosted.org/packages/65/eb/1f97cb97bfc2390a276969c6fae16075da282f5058082d4cb10c6c5c1dba/six-1.14.0-py2.py3-none-any.whl

Installing collected packages: jmespath, six, python-dateutil, docutils, urllib3, botocore, s3transfer, boto3

Successfully installed boto3-1.13.14 botocore-1.16.14 docutils-0.15.2 jmespath-0.10.0 python-dateutil-2.8.1 s3transfer-0.3.3 six-1.14.0 urllib3-1.25.9

You are using pip version 9.0.3, however version 20.1.1 is available.

You should consider upgrading via the 'pip install --upgrade pip' command.

buildingmodernapps:~/environment/validate-dragon \$ cd validate-dragon-package/
buildingmodernapps:~/environment/validate-dragon/validate-dragon-package \$ zip -r9 \${OLDPWD}/ pythonValidateDragonFunction.zip .

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

validateDragon.p Commands.txt

BuildingModernAppsPython add-dragon Commands.txt newDragonPayload.json list-dragons SDK Exploration validate-dragon validate-dragon-package Commands.txt duplicateDragonPayload.json newDragonPayload.json pythonValidateDragonFunction.zip validateDragon.py README.md

```
1 Download dependencies into folder
2     sudo pip install --target ./validate-dragon-package boto3
3
4 Zip up your code (with dependencies)
5     zip -r9 ${OLDPWD}/ pythonValidateDragonFunction.zip .
6
7 Add python script to zip
8     zip -g pythonValidateDragonFunction.zip validateDragon.py
9
10 Create Lambda Function
11     aws lambda create-function --function-name validate-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/Validat
12
13 Invoke Lambda Function
14     aws lambda invoke --function-name validate-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17     aws lambda update-function-code --function-name validate-dragon --zip-file file:///pythonValidateDragonFunction.zip --publis
(59 Bytes) 8:63 Text Spaces: 4
```

bash - "ip-172-31 x Immediate

```
adding: jmespath/_pycache__parser.cpython-36.pyc (deflated 0%)
adding: jmespath/_pycache__functions.cpython-36.pyc (deflated 58%)
adding: jmespath/exceptions.py (deflated 76%)
adding: jmespath/compat.py (deflated 67%)
adding: jmespath/ast.py (deflated 76%)
adding: six.py (deflated 76%)
buildingmodernapps:~/environment/validate-dragon/validate-dragon-package $ cd ../
buildingmodernapps:~/environment/validate-dragon $ zip -g pythonValidateDragonFunction.zip validateDragon.py
      zip warning: pythonValidateDragonFunction.zip not found or empty
      adding: validateDragon.py (deflated 57%)
buildingmodernapps:~/environment/validate-dragon $
```

AWS Management Console X BuildingModernAppsPython - +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P) validateDragon.p Commands.txt

Environment

BuildingModernAppsPython /h

- add-dragon
 - addDragon.py
 - Commands.txt
 - newDragonPayload.json
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - pythonValidateDragonFunction.zip
 - validateDragon.py
 - README.md

1
2
3
4
5
6
7
8
9
10
11 runtime python3.6 --role arn:aws:iam::302211264422:role/ValidateDragonLambdaExecutionRole --handler validateDragon.validate --
12
13
14 --payload file://newDragonPayload.json
15
16
17 in --zip-file fileb://pythonValidateDragonFunction.zip --publish

12:1 Text Spaces: 4

bash - "ip-172-31" Immediate

```
buildingmodernapps:~/environment/validate-dragon/validate-dragon-package $ cd ../
buildingmodernapps:~/environment/validate-dragon $ zip -g pythonValidateDragonFunction.zip validateDragon.py
      zip warning: pythonValidateDragonFunction.zip not found or empty
      adding: validateDragon.py (deflated 57%)
buildingmodernapps:~/environment/validate-dragon $
```

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P) validateDragon.p Commands.txt

Environment

BuildingModernAppsPython - h

- add-dragon
 - addDragon.py
 - Commands.txt
 - newDragonPayload.json
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - pythonValidateDragonFunction.zip
 - validateDragon.py
 - README.md

```
1 Download dependencies into folder
2     sudo pip install --target ./validate-dragon-package boto3
3
4 Zip up your code (with dependencies)
5     zip -r9 ${OLDPWD}/ pythonValidateDragonFunction.zip .
6
7 Add python script to zip
8     zip -g pythonValidateDragonFunction.zip validateDragon.py
9
10 Create Lambda Function
11     aws lambda create-function --function-name validate-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/ValidateDragonLambdaExecutionRole --handler validateDragon.validate --publish --zip-file fileb://pythonValidateDragonFunction.zip
12
13 Invoke Lambda Function
14     aws lambda invoke --function-name validate-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17     aws lambda update-function-code --function-name validate-dragon --zip-file fileb://pythonValidateDragonFunction.zip --publish
```

12:1 Text Spaces: 4

bash - "ip-172-31 x Immediate x +

```
buildingmodernapps:~/environment/validate-dragon/validate-dragon-package $ cd ../
buildingmodernapps:~/environment/validate-dragon $ zip -g pythonValidateDragonFunction.zip validateDragon.py
    zip warning: pythonValidateDragonFunction.zip not found or empty
    adding: validateDragon.py (deflated 57%)
buildingmodernapps:~/environment/validate-dragon $ aws lambda create-function --function-name validate-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/ValidateDragonLambdaExecutionRole --handler validateDragon.validate --publish --zip-file fileb://pythonValidateDragonFunction.zip
```

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P) validateDragon.p Commands.txt

Environment

BuildingModernAppsPython - /h

- add-dragon
 - addDragon.py
 - Commands.txt
 - newDragonPayload.json
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - pythonValidateDragonFunction.zip
 - validateDragon.py
 - README.md

1 Download dependencies into folder
2 sudo pip install --target ./validate-dragon-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonValidateDragonFunction.zip .
6
7 Add python script to zip
8 zip -g pythonValidateDragonFunction.zip validateDragon.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name validate-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/ValidateDragonLambdaExecutionRole
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name validate-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name validate-dragon --zip-file file:///pythonValidateDragonFunction.zip --publish
(38 Bytes) 14:56 Text Spaces: 4

python2.7 - "ip-1" x Immediate x +

```
REVISIONID : c0996910150104-0100-4100-cc40-040510909500 ,  
"MemorySize": 128,  
"FunctionArn": "arn:aws:lambda:us-east-1:302211264422:function:validate-dragon",  
"Version": "1",  
"Role": "arn:aws:iam::302211264422:role/ValidateDragonLambdaExecutionRole",  
"Timeout": 3,  
"LastModified": "2020-05-20T20:11:45.106+0000",  
"Handler": "validateDragon.validate",  
"Runtime": "python3.6",  
"Description": ""  
}
```

buildingmodernapps:~/environment/validate-dragon \$

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (⌘ P)

validateDragon.p Commands.txt newDragonPaylc

BuildingModernAppsPython /h

- add-dragon
 - addDragon.py
 - Commands.txt
 - newDragonPayload.json
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - pythonValidateDragonFunction.zip
 - validateDragon.py
 - README.md

1 {
2 "description_str": "George is a new dragon, we don't know much about them yet.",
3 "dragon_name_str": "George",
4 "family_str": "green",
5 "location_city_str": "seattle",
6 "location_country_str": "usa",
7 "location_neighborhood_str": "4th st",
8 "location_state_str": "washington"
9 }

9:2 JSON Spaces: 5

bash - ip-172-31-1-111 : 2020-05-20T20:11:45.106+0000 , ca0969015004-cc4d-0100-4100-4100-a74cfc
"MemorySize": 128,
"FunctionArn": "arn:aws:lambda:us-east-1:302211264422:function:validate-dragon",
"Version": "1",
"Role": "arn:aws:iam::302211264422:role/ValidateDragonLambdaExecutionRole",
"Timeout": 3,
"LastModified": "2020-05-20T20:11:45.106+0000",
"Handler": "validateDragon.validate",
"Runtime": "python3.6",
"Description": ""
}
buildingmodernapps:~/environment/validate-dragon \$

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P) validateDragon.p Commands.txt

Environment

BuildingModernAppsPython - /h

- add-dragon
 - addDragon.py
 - Commands.txt
 - newDragonPayload.json
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - output.txt
 - pythonValidateDragonFunction.zip
 - validateDragon.py
- README.md

1 Download dependencies into folder
2 sudo pip install --target ./validate-dragon-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 \${OLDPWD}/ pythonValidateDragonFunction.zip .
6
7 Add python script to zip
8 zip -g pythonValidateDragonFunction.zip validateDragon.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name validate-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/Vali
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name validate-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name validate-dragon --zip-file file:///pythonValidateDragonFunction.zip --publ
(104 Bytes) 15:1 Text Spaces: 4

python2.7 - "ip-1" × Immediate × +

```
{
    "Handler": "validateDragon.validate",
    "Runtime": "python3.6",
    "Description": ""
}
buildingmodernapps:~/environment/validate-dragon $ aws lambda invoke --function-name validate-dragon output.txt --payload file://newDragonPayload.json
{
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
buildingmodernapps:~/environment/validate-dragon $
```

AWS Management Console < BuildingModernAppsPython - console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

validateDragon.p Commands.txt output.txt

1 "Dragon Validated"

Environment

BuildingModernAppsPython /h

- add-dragon
 - addDragon.py
 - Commands.txt
 - newDragonPayload.json
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - output.txt
 - pythonValidateDragonFunction.zip
 - validateDragon.py
- README.md

1:1 Text Spaces: 4

python2.7 - "ip-1" Immediate

```
    "Handler": "validateDragon.validate",
    "Runtime": "python3.6",
    "Description": ""
}
buildingmodernapps:~/environment/validate-dragon $ aws lambda invoke --function-name validate-dragon output.txt --payload file://newDragonPayload.json
{
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
buildingmodernapps:~/environment/validate-dragon $
```

Now, let's go over the addDragon.py

AWS Management Console < BuildingModernAppsPython -> +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit Find View Go Run Tools Window Support Preview Run

Go to Anything (% P)

Environment

addDragon.py

```
bucket_name = ssm.get_parameter( Name='dragon_data_bucket_name',WithDecryption=False)['Parameter']['Value']
file_name = ssm.get_parameter( Name='dragon_data_file_name',WithDecryption=False)['Parameter']['Value']

def addDragonToFile(event, context):
    dragon_data = {
        "description_str":event['description_str'],
        "dragon_name_str":event['dragon_name_str'],
        "family_str":event['family_str'],
        "location_city_str":event['location_city_str'],
        "location_country_str":event['location_country_str'],
        "location_neighborhood_str":event['location_neighborhood_str'],
        "location_state_str":event['location_state_str']
    }

    resp=s3.get_object(Bucket=bucket_name, Key=file_name)
    data=resp.get('Body').read()

    # Add logic here to process the data and update the file
    # ...

    return {
        "FunctionError": "Unhandled",
        "ExecutedVersion": "$LATEST",
        "StatusCode": 200
    }

```

bash - "ip-172-31-1-1" Immediate

```
"ExecutedVersion": "$LATEST",
"StatusCode": 200
}
buildingmodernapps:~/environment/validate-dragon $ aws lambda invoke --function-name validate-dragon output.txt --payload file://duplicateDragonPayload.json
{
    "FunctionError": "Unhandled",
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
buildingmodernapps:~/environment/validate-dragon $
```

addDragon.py

The screenshot shows the AWS Management Console interface for building modern AWS Lambda functions. The top navigation bar includes tabs for 'AWS Cloud9' and 'BuildingModernAppsPython'. The main area displays a code editor for 'addDragon.py' and a terminal window.

Code Editor:

```
14     "family_str":event['family_str'],
15     "location_city_str":event['location_city_str'],
16     "location_country_str":event['location_country_str'],
17     "location_neighborhood_str":event['location_neighborhood_str'],
18     "location_state_str":event['location_state_str']
19 }
20
21 resp=s3.get_object(Bucket=bucket_name, Key=file_name)
22 data=resp.get('Body').read()
23
24 json_data = json.loads(data)
25 json_data.append(dragon_data)
26 s3.put_object(Bucket=bucket_name, Key=file_name, Body=json.dumps(json_data).encode())
27
```

Terminal:

```
bash - "ip-172-31 x" Immediate x +
1
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
buildingmodernapps:~/environment/validate-dragon $ aws lambda invoke --function-name validate-dragon output.txt --payload file://duplicateDragonPayload.json
{
    "FunctionError": "Unhandled",
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
buildingmodernapps:~/environment/validate-dragon $
```

reading the body of a response, appending it to the json data, and overwriting the entire object using `s3.put_object`

using S3, we can't add a line.

we need to override the entire object.

list of commands that we need to run in order to this Lambda function:

- Download dependencies into folder
- ```
sudo pip install --target ./add-dragon-package boto3
```
- Zip up your code (with dependencies)
- ```
zip -r9 ${OLDPWD}/ pythonAddDragonFunction.zip .
```
- Add python script to zip
- ```
zip -g pythonAddDragonFunction.zip addDragon.py
```
- Create Lambda Function
- ```
aws lambda create-function --function-name add-dragon --runtime python3.6 --role <IAM  
ROLE ARN> --handler addDragon.addDragonToFile --publish --zip-file  
fileb://pythonAddDragonFunction.zip
```
- Invoke Lambda Function
- ```
aws lambda invoke --function-name add-dragon output.txt --payload
file://newDragonPayload.json
```
- Update Lambda Code
- ```
aws lambda update-function-code --function-name add-dragon --zip-file  
fileb://pythonAddDragonFunction.zip --publish
```

AWS Management Console < BuildingModernAppsPython -

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

Environment

BuildingModernAppsPython - /h

- add-dragon
 - addDragon.py
 - Commands.txt
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - output.txt
 - pythonValidateDragonFunction.zip
 - validateDragon.py
 - README.md

addDragon.py Commands.txt

```
1 Download dependencies into folder
2 sudo pip install --target ./add-dragon-package boto3
3
4 Zip up your code (with dependencies)
5 zip -r9 ${OLDPWD}/ pythonAddDragonFunction.zip .
6
7 Add python script to zip
8 zip -g pythonAddDragonFunction.zip addDragon.py
9
10 Create Lambda Function
11 aws lambda create-function --function-name add-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/AddDragon
12
13 Invoke Lambda Function
14 aws lambda invoke --function-name add-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17 aws lambda update-function-code --function-name add-dragon --zip-file fileb://pythonAddDragonFunction.zip --publish
```

(57 Bytes) 3:1 Text Spaces: 4

sudo - "ip-172-31" Immediate

```
buildingmodernapps:~/environment/add-dragon $ sudo pip install --target ./add-dragon-package boto3
Collecting boto3
  Using cached https://files.pythonhosted.org/packages/a8/70/608e4a4b328c574630c81f9c0fbfdea660ad8bf3f7184280793f502073c8/boto3-1.13.14-py2.py3-none-any.whl
Collecting botocore<1.17.0,>=1.16.14 (from boto3)
```

AWS Management Console X BuildingModernAppsPython - / +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

BuildingModernAppsPython - /h

- add-dragon
- add-dragon-package
- addDragon.py
- Commands.txt
- newDragonPayload.json
- list-dragons
- SDK Exploration
- validate-dragon
- validate-dragon-package
- Commands.txt
- duplicateDragonPayload.json
- newDragonPayload.json
- output.txt
- pythonValidateDragonFunction.zip
- validateDragon.py

Commands.txt

```
1 Download dependencies into folder
2     sudo pip install --target ./add-dragon-package boto3
3
4 Zip up your code (with dependencies)
5     zip -r9 ${OLDPWD}/ pythonAddDragonFunction.zip .
6
7 Add python script to zip
8     zip -g pythonAddDragonFunction.zip addDragon.py
9
10 Create Lambda Function
11     aws lambda create-function --function-name add-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/AddDragon
12
13 Invoke Lambda Function
14     aws lambda invoke --function-name add-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17     aws lambda update-function-code --function-name add-dragon --zip-file file://pythonAddDragonFunction.zip --publish
```

(53 Bytes) 6:1 Text Spaces: 4

bash - "ip-172-31" x Immediate x

```
0aa29227bf58149dc72f/urllib3-1.25.9-py2.py3-none-any.whl
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore<1.17.0,>=1.16.14->boto3)
    Using cached https://files.pythonhosted.org/packages/65/eb/1f97cb97bfc2390a276969c6fae16075da282f50
58082d4cb10c6c5c1dba/six-1.14.0-py2.py3-none-any.whl
Installing collected packages: jmespath, docutils, six, python-dateutil, urllib3, botocore, s3transfer, boto3
Successfully installed boto3-1.13.14 botocore-1.16.14 docutils-0.15.2 jmespath-0.10.0 python-dateutil-2.8.1 s3transfer-0.3.3 six-1.14.0 urllib3-1.25.9
You are using pip version 9.0.3, however version 20.1.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

buildingmodernapps:~/environment/add-dragon \$ cd add-dragon-package/

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

Environment

BuildingModernAppsPython - /h

- add-dragon
- add-dragon-package
- Commands.txt
- newDragonPayload.json
- pythonAddDragonFunction.zip
- list-dragons
- SDK Exploration
- validate-dragon
- validate-dragon-package
- Commands.txt
- duplicateDragonPayload.json
- newDragonPayload.json
- output.txt
- pythonValidateDragonFunction.zip
- validateDragon.py

addDragon.py Commands.txt

- Download dependencies into folder
sudo pip install --target ./add-dragon-package boto3
- Zip up your code (with dependencies)
zip -r9 \${OLDPWD}/ pythonAddDragonFunction.zip .
- Add python script to zip
zip -g pythonAddDragonFunction.zip addDragon.py
- Create Lambda Function
aws lambda create-function --function-name add-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/AddDragon
- Invoke Lambda Function
aws lambda invoke --function-name add-dragon output.txt --payload file://newDragonPayload.json
- Update Lambda Code
aws lambda update-function-code --function-name add-dragon --zip-file file://pythonAddDragonFunction.zip --publish

(53 Bytes) 9:1 Text Spaces: 4

bash - "ip-172-31" x Immediate x +

```
buildingmodernapps:~/environment/add-dragon$ cd ..
buildingmodernapps:~/environment/add-dragon$ zip -g pythonAddDragonFunction.zip addDragon.py
zip warning: pythonAddDragonFunction.zip not found or empty
adding: addDragon.py (deflated 61%)
buildingmodernapps:~/environment/add-dragon$
```

AWS Management Console < BuildingModernAppsPython -

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run

Go to Anything (% P)

Commands.txt

Environment

BuildingModernAppsPython - /h

- add-dragon
 - add-dragon-package
 - addDragon.py
 - Commands.txt
 - newDragonPayload.json
 - pythonAddDragonFunction.zip
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - output.txt
 - pythonValidateDragonFunction.zip
 - validateDragon.py
- README.md

1
2
3
4
5
6
7
8
9
10
11 -runtime python3.6 --role arn:aws:iam::302211264422:role/AddDragonLambdaExecutionRole --handler addDragon.addDragonToFile --
12
13
14 --payload file://newDragonPayload.json
15
16
17 gon --zip-file fileb://pythonAddDragonFunction.zip --publish

12:1 Text Spaces: 4

bash - "ip-172-31-0-104 ~" Immediate

```
adding: jmespath/_pycache_/compat.cpython-36.pyc (deflated 52%)  
adding: jmespath/_pycache_/parser.cpython-36.pyc (deflated 60%)  
adding: jmespath/_pycache_/functions.cpython-36.pyc (deflated 58%)  
adding: jmespath/exceptions.py (deflated 76%)  
adding: jmespath/compat.py (deflated 67%)  
adding: jmespath/ast.py (deflated 76%)  
adding: six.py (deflated 76%)
```

buildingmodernapps:~/environment/add-dragon/add-dragon-package \$ cd ..
buildingmodernapps:~/environment/add-dragon \$ zip -g pythonAddDragonFunction.zip addDragon.py
zip warning: pythonAddDragonFunction.zip not found or empty
adding: addDragon.py (deflated 61%)
buildingmodernapps:~/environment/add-dragon \$

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

Environment

BuildingModernAppsPython - /h

- add-dragon
 - add-dragon-package
 - addDragon.py
 - Commands.txt
 - newDragonPayload.json
 - pythonAddDragonFunction.zip
- list-dragons
- SDK Exploration
- validate-dragon
 - validate-dragon-package
 - Commands.txt
 - duplicateDragonPayload.json
 - newDragonPayload.json
 - output.txt
 - pythonValidateDragonFunction.zip
 - validateDragon.py
- README.md

addDragon.py x Commands.txt x +

```
1 Download dependencies into folder
2     sudo pip install --target ./add-dragon-package boto3
3
4 Zip up your code (with dependencies)
5     zip -r9 ${OLDPWD}/ pythonAddDragonFunction.zip .
6
7 Add python script to zip
8     zip -g pythonAddDragonFunction.zip addDragon.py
9
10 Create Lambda Function
11     aws lambda create-function --function-name add-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/AddDragonLambdaExecutionRole
12
13 Invoke Lambda Function
14     aws lambda invoke --function-name add-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17     aws lambda update-function-code --function-name add-dragon --zip-file fileb://pythonAddDragonFunction.zip --publish
```

12:1 Text Spaces: 4

python2.7 - "ip-1" x Immediate x +

```
REVISITATION : 2020-05-20T20:18:02.281+0000 ,
"MemorySize": 128,
"FunctionArn": "arn:aws:lambda:us-east-1:302211264422:function:add-dragon",
"Version": "1",
"Role": "arn:aws:iam::302211264422:role/AddDragonLambdaExecutionRole",
"Timeout": 3,
"LastModified": "2020-05-20T20:18:02.281+0000",
"Handler": "addDragon.addDragonToFile",
"Runtime": "python3.6",
"Description": ""
```

}

buildingmodernapps:~/environment/add-dragon \$

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run

Go to Anything (⌘ P)

addDragon.py Commands.txt newDragonPayload

BuildingModernAppsPython - /h

- add-dragon
- add-dragon-package
- addDragon.py
- Commands.txt
- newDragonPayload.json
- pythonAddDragonFunction.zip
- list-dragons
- SDK Exploration
- validate-dragon
- validate-dragon-package
- Commands.txt
- duplicateDragonPayload.json
- newDragonPayload.json
- output.txt
- pythonValidateDragonFunction.zip
- validateDragon.py

README.md

1 {
2 "description_str": "George is a new dragon, we don't know much about them yet.",
3 "dragon_name_str": "George",
4 "family_str": "green",
5 "location_city_str": "seattle",
6 "location_country_str": "usa",
7 "location_neighborhood_str": "4th st",
8 "location_state_str": "washington"
9 }

8:39 JSON Spaces: 5

bash - ip-172-31-1-111 : 8000

```
RESULTS : {"FunctionArn": "arn:aws:lambda:us-east-1:302211264422:function:add-dragon", "MemorySize": 128, "Role": "arn:aws:iam::302211264422:role/AddDragonLambdaExecutionRole", "Runtime": "python3.6", "Handler": "addDragon.addDragonToFile", "Version": "1", "LastModified": "2020-05-20T20:18:02.281+0000", "Description": ""}
```

buildingmodernapps:~/environment/add-dragon \$

AWS Management Console X BuildingModernAppsPython - X +

console.aws.amazon.com/cloud9/ide/e6cb398c0bf04e50aa8e917f22e07558

AWS Cloud9 File Edit View Go Run Tools Window Support Preview Run Share

Go to Anything (% P)

Environment

addDragon.py Commands.txt

```
1 Download dependencies into folder
2     sudo pip install --target ./add-dragon-package boto3
3
4 Zip up your code (with dependencies)
5     zip -r9 ${OLDPWD}/ pythonAddDragonFunction.zip .
6
7 Add python script to zip
8     zip -g pythonAddDragonFunction.zip addDragon.py
9
10 Create Lambda Function
11     aws lambda create-function --function-name add-dragon --runtime python3.6 --role arn:aws:iam::302211264422:role/AddDragon
12
13 Invoke Lambda Function
14     aws lambda invoke --function-name add-dragon output.txt --payload file://newDragonPayload.json
15
16 Update Lambda Code
17     aws lambda update-function-code --function-name add-dragon --zip-file file:///pythonAddDragonFunction.zip --publish
```

15:5 Text Spaces: 4

python2.7 - "ip-1" × Immediate × +

```
    "Handler": "addDragon.addDragonToFile",
    "Runtime": "python3.6",
    "Description": ""
}
buildingmodernapps:~/environment/add-dragon $ aws lambda invoke --function-name add-dragon output.txt --payload file://newDragonPayload.json
{
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
buildingmodernapps:~/environment/add-dragon $
```

The screenshot shows the AWS Management Console Cloud9 IDE interface. The left sidebar displays the project structure under 'BuildingModernAppsPython - /'. The main workspace contains three tabs: 'addDragon.py', 'Commands.txt', and 'output.txt'. The 'output.txt' tab shows the command 'aws lambda invoke --function-name add-dragon output.txt --payload file://newDragonPayload.json' and its execution results. The results show a JSON object with fields like 'Handler', 'Runtime', 'Description', and 'StatusCode'. The status bar at the bottom indicates 'buildingmodernapps:~/environment/add-dragon \$'.

```
python2.7 -> "ip-1" x Immediate x +  
    "Handler": "addDragon.addDragonToFile",  
    "Runtime": "python3.6",  
    "Description": ""  
}  
buildingmodernapps:~/environment/add-dragon $ aws lambda invoke --function-name add-dragon output  
.txt --payload file://newDragonPayload.json  
{  
    "ExecutedVersion": "$LATEST",  
    "StatusCode": 200  
}  
buildingmodernapps:~/environment/add-dragon $
```

it worked. Null is because our function doesn't have a return.

Lab 4: AWS Lambda Functions

<https://aws-tc-largeobjects.s3.amazonaws.com/DEV-AWS-MO-BuildingRedux/exercise-4-lambda.html>

Lab 4 - in this lab you will work directly with AWS Lambda.

First, you will deploy
the ListDragon, AddDragon, and ValidateDragon AWS Lambda functions.

After you implement the functions, you will then update the GET method so that the REST application programming interface (API) can use the Lambda functions.

I strongly recommend that you explore the source code to
view how the code and the Lambda handlers work.