S3BucketToAPI

Release 0.0.1

May 18, 2021

Contents

CHAPTER 1

Introduction

There is a need to be able to call an API when an object is placed in an S3 bucket. Currently this process is done manually, which is a labor intensive process. The author has therefore designed and implemented a serverless application using AWS SAM (as well as an accompanying CI/CD pipeline) to achieve this.

1.1 Purpose

This document will provide a detailed description of the components of the project, including the AWS SAM template that forms the core of the application, as well as the resources it creates, such as S3 buckets and Lambda functions.

This information is meant as a reference material for the author himself, as well as any future developers that find themselves working on this project. Some of the technical descriptions may not be particularly accessible to a non-technical audience, but an effort has been made to ensure that this documentation may also provide value to these users.

1.2 Scope

As of the time of writing, the components we have included in this application are the following:

- **Lambda function which is triggered when a file meeting specified criteria is placed in the intended S3 bucket, and:**
  - Checks a configuration file to determine if the data in the s3 file should be processed and passed to the API
  - Calls the API
  - Save the results to a Dynamo DB
  - Send out notifications using SNS to interested parties
- Lambda function and an accompanying API gateway, which will change the Enable setting in the configuration file, allowing administrators to stop the processing of jobs as needed

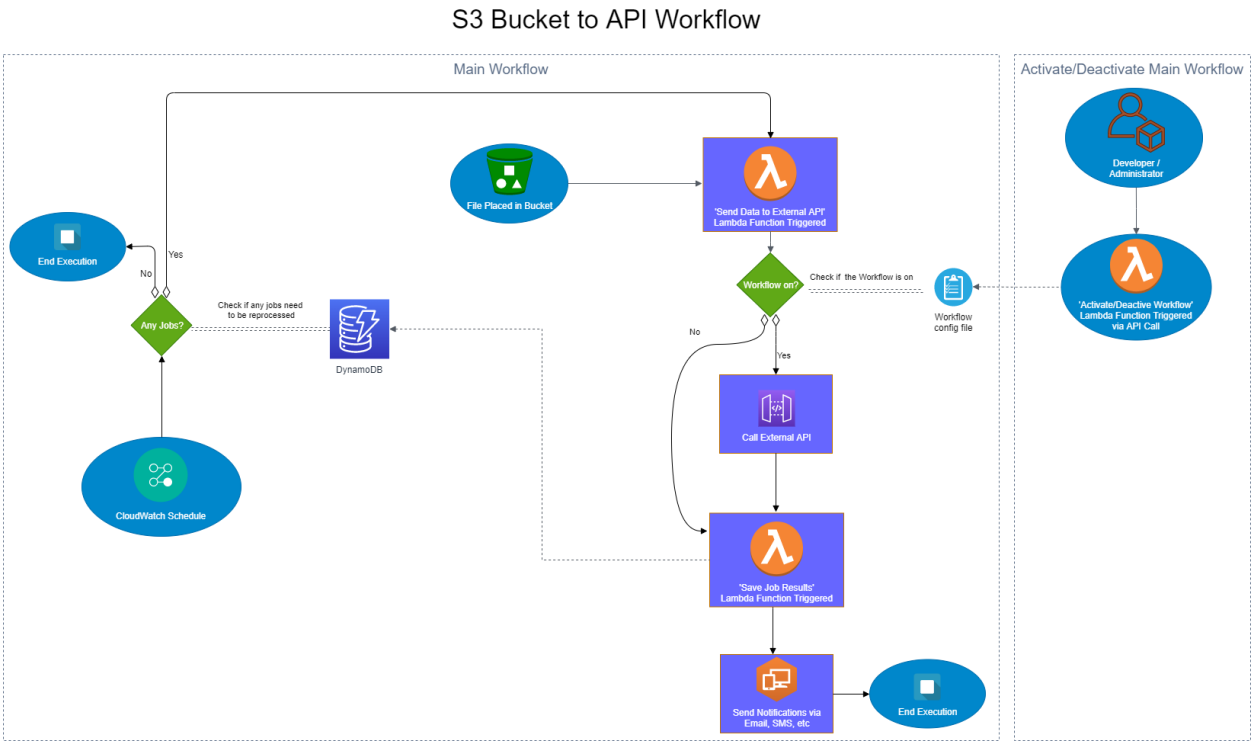
- A CloudWatch schedule that periodically checks the DynamoDB for any files that need to be reprocessed, and sends any such jobs back to the first Lambda function described above.
- Lambda function and an accompanying API gateway which allows an administrator to specify a file to be reprocessed.
- An AWS CodePipeline that provides for separate development, demo, and production environments, and checkpoints requiring developer and administrator approval before deploying to higher environments.
- Unit tests that will be fun as part of the build and deployment process

1.3 Overview

The rest of this documentation gives a detailed description of the AWS SAM template and all the components created from it, as well as the CI/CD solution that has been implemented using AWS CodePipeline.

1.4 Diagram

The following is a diagram of the application's workflow as well as administrator access to configurations.



CHAPTER 2

Contents

2.1 Workflow Configuration

2.1.1 Purpose

There may be instances in which an administrator has deemed that the process of files should be suspended. this can be achieved through the use of the API Gateway, as described below, to enable or disable file processing.

API Under Construction. Details to be added

2.2 AWS SAM / CloudFormation

Note: This page is auto-generated from the latest SAM template.

2.2.1 SAM Template

2.2.1.1 Template Description & Transformation Information:

AWSTemplateFormatVersion: "2010-09-09"

Transform: AWS::Serverless-2016-10-31

Description: When a file is saved to S3 Bucket, API is called

2.2.1.2 Parameters:

EnvType: Description: Environment type.

Default: dev

Type: String

AllowedValues: [prod, dev, demo]

ConstraintDescription: must specify prod, dev, or demo.

2.2.1.3 Globals:

Function: Timeout: 60

More info about Globals: https://github.com/awslabs/serverless-application-model/blob/master/docs/globals.rst

2.2.1.4 Resources:

2.2.1.4.1 # DynamoDB Table

 # This table is used to keep track of file processing

Table: Type: AWS::DynamoDB::Table Properties:

TableName: !Sub 'bucket-to-api-table-${EnvType}' ProvisionedThroughput:

ReadCapacityUnits: 5 WriteCapacityUnits: 5

KeySchema: - AttributeName: RequestId

KeyType: HASH

AttributeDefinitions: - AttributeName: RequestId

AttributeType: S

2.2.1.4.2 # Main Helper Bucket

 # This bucket is employed to store files used or created by the application, e.g. the configuration file

WorkflowBucket: Type: AWS::S3::Bucket Properties:

BucketName: !Sub "bucket-to-api-workflow-bucket-${EnvType}"

2.2.1.4.3 # Enabled Parameter

 # This parameter will determine if the workflow processes files (which is stored in the AWS Parameter Store), as well as any other future configurations we may add

Type: AWS::SSM::Parameter Properties:

AllowedPattern: String Description: Whether file data should be sent to api by bucket-to-api app Name: !Sub "bucket-to-api-enabled-${EnvType}" Policies: String Type: String Value: False

2.2.1.4.4 # Testing Bucket

 # This bucket is used in place of the bucket that files will be placed in in production

TestBucket: Type: AWS::S3::Bucket Properties:

BucketName: !Sub "bucket-to-api-test-bucket-${EnvType}"

2.2.1.4.5 # Main Lambda Function

 # this Lambda function is the workhorse of the application, and triggers most other components

WorkflowMainFunction: Type: AWS::Serverless::Function Properties:

FunctionName: !Sub "bucket-to-api-workflow-main-function-${EnvType}" CodeUri: AWS/Services/Lambda/workflow_main_function/ Handler: workflow_main_function.lambda_handler Runtime: python3.8 Environment:

Variables: ENV_TYPE: !Ref EnvType TABLE_NAME: !Sub 'bucket-to-api-table-${EnvType}'

Events:

BucketEvent1: Type: S3 Properties:

Bucket: !Ref TestBucket Events: s3:ObjectCreated:*

Policies: #- S3ReadPolicy: # BucketName: !Ref TestBucket - S3CrudPolicy:

BucketName: !Ref WorkflowBucket

- **DynamoDBCrudPolicy:** TableName: !Sub 'bucket-to-api-table-${EnvType}'
- **SNSPublishMessagePolicy:** TopicName: !Sub "bucket-to-api-sns-topic-${EnvType}"

 # Code Snippets, retained as reference:

2.2.1.4.6 # Enable/Disable Function

 # this Lambda function is used to enable or disable the processing of files in the application by editing the config file

ToggleFunction:

Type: AWS::Serverless::Function

Properties:

CodeUri: AWS/Services/Lambda/toggle_function/

Handler: toggle_function.lambda_handler

Runtime: python3.8

Events: # ActivateDeactivate:

Type: Api

Properties:

Path: /toggle

Method: ANY

Policies:

- S3CrudPolicy:

BucketName: !Ref WorkflowBucket

2.2.1.5 # Outputs:

2.2.1.5.1 # Enable/Disable API

 # this API allows administrators to trigger the lambda function that enables or disables the processing of files by the application
ToggleApi:

Description: "Allows you to activate, deactivate, and check the current status of the workflow"

Value: !Sub "https://${ServerlessRestApi}.execute-api.${AWS::Region}.amazonaws.com/${EnvType}/activatedeactivate/"

ServerlessRestApi is an implicit API created out of Events key under Serverless::Function

Find out more about other implicit resources you can reference within SAM

https://github.com/awslabs/serverless-application-model/blob/master/docs/internals/generated_resources.rst#api

2.2.1.5.2 # Sample API and Function

 # these are retained for reference # HelloWorldApi:

Description: "API Gateway endpoint URL for Prod stage for Hello World function"

Value: !Sub "https://${ServerlessRestApi}.execute-api.${AWS::Region}.amazonaws.com/Prod/hello/"

HelloWorldFunction:

Description: "Hello World Lambda Function ARN"

Value: !GetAtt HelloWorldFunction.Arn

HelloWorldFunctionIamRole:

Description: "Implicit IAM Role created for Hello World function"

Value: !GetAtt HelloWorldFunctionRole.Arn

2.3 Lambda Modules

2.3.1 workflow_main_funtion.py

The main module of the workflow

`AWS.Services.Lambda.workflow_main_function.workflow_main_function.`**`CheckIfEnabled`**`(`*ENV_TYPE*`)`
Function to check config to see if files should be processed

Parameters **ENV_TYPE** (*string, required*) – dev, demo or prod

Returns True if files should be proccessed, false otherwise

Return type bool

AWS.Services.Lambda.workflow_main_function.workflow_main_function.**lambda_handler**(*event*, *context*)

Function to pass jobs to external api

Steps:

1. Check if the config file to see if we call the api
2. Make api call, depending on the config file in the previous step
3. Save the job results to DynamoDB table
4. Send notifications on job results

Parameters

- **event** (*dict, required*) – file is deposited in S3 bucket

  Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format

- **context** (*object, required*) – Lambda Context runtime methods and attributes

  Context doc: https://docs.aws.amazon.com/lambda/latest/dg/python-context-object.html

Returns Success or failure of function

Return type int

2.3.2 toggle_function.py

Module to allow administrators to turn the main workflow on or off

The config file, which will live in a workflow S3 bucket, will determine if the workflow processes the jobs. This function will edit the config file to set switch this to on or off, and an API gateway will give us access to this lambda function.

AWS.Services.Lambda.toggle_function.toggle_function.**lambda_handler**(*event*, *context*)

Function to change the configuration parameter that determines whether the application sends file data to the external API

Parameters

- **event** (*dict, required*) – api gateway is used to call this function

  Event doc: https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-lambda-proxy-integrations.html#api-gateway-simple-proxy-for-lambda-input-format

- **context** (*object, required*) – Lambda Context runtime methods and attributes

  Context doc: https://docs.aws.amazon.com/lambda/latest/dg/python-context-object.html

Returns Success or failure of function

Return type int

2.3.3 Modules (Tree Map View)

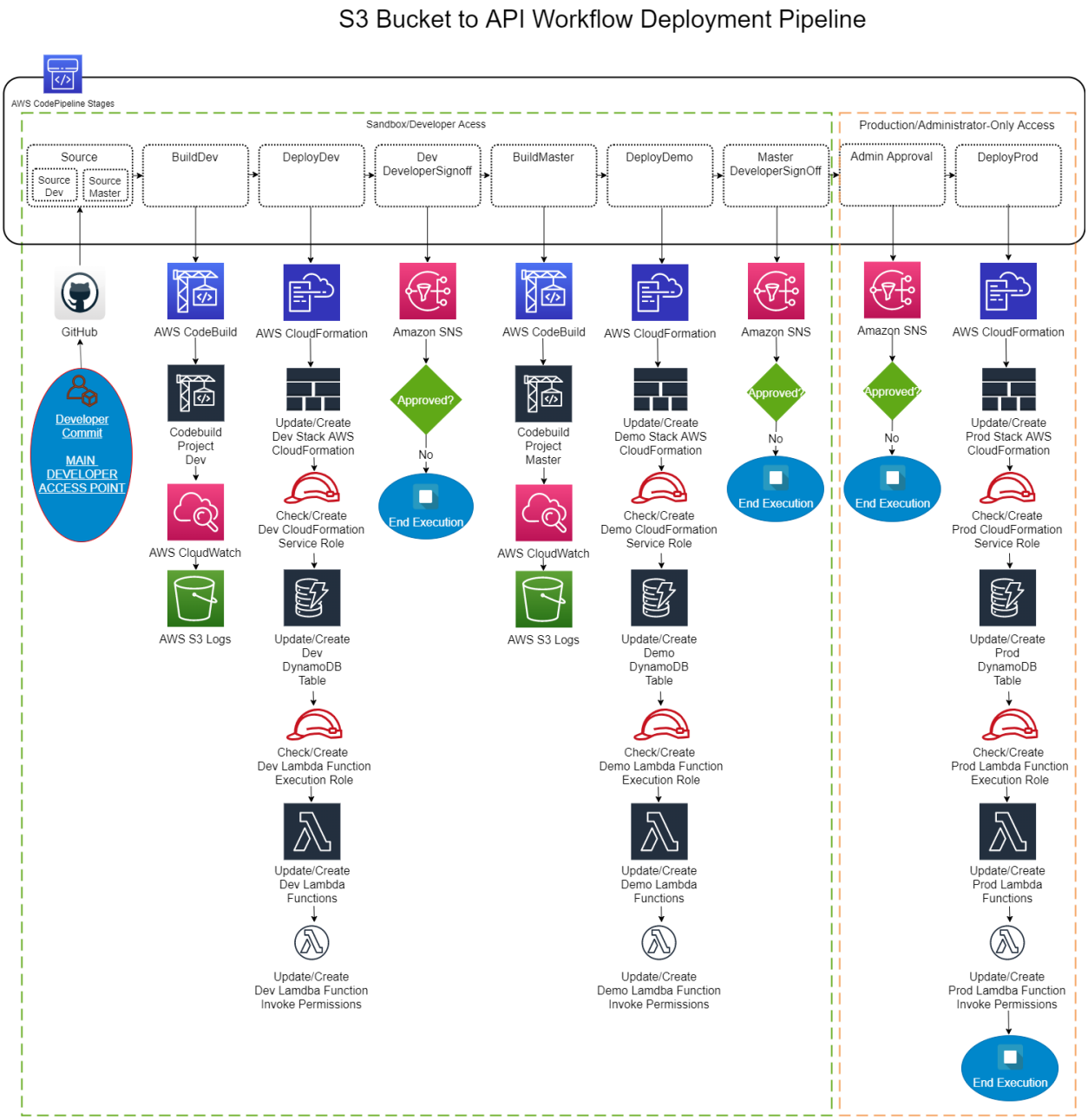
2.4 Continuous Integration & Deployment

2.4.1 Purpose

In order to follow industry best practices, the author has implemented a CI/CD pipeline. Given that this project is a serverless application built around AWS services, we have chosen to build this pipeline using AWS CodePipeline

2.4.2 Diagram

The following is a diagram of the CodePipeline along with the services it employs.



2.4.3 Template

The CodePipeline template below is an export of the latest version of the pipeline. This template may be used to create the skeleton of the pipeline in other environments. Please note that although this will create the pipeline and the steps within it, the resources it uses such as CodeBuild projects or service roles.

Under Construction. Template to be added

2.5 Contact

For questions or support requests, please contact the author at max.albrecht@rhsps.com, or at max.albrecht100@gmail.com

CHAPTER 3

Indices and tables

- modindex

Python Module Index

a

Index

A

C

L