Serverless/Faas

Introduction to running a serverless application using AWS and the Serverless framework

Contents

Severless /
Function as a Service (FaaS)

Serverless Computing

- Abstract server management/low-level infrastructure decisions away from developers.
- Allocation of resources managed by cloud provider (not application architect)

Pros/Cons

Pros

- Less focus on server mgnt
- Automatically scales
- Pay only when used
- Modular

Cons

- Someone else is managing your infrastructure
- Potentially tough to debug
- Keeping track of 100s/1000s functions
-

Summary

- Upload chunks of functionality to cloud
- Functions get executed independently.
- No idling monolithic REST server to handle potential load,
- Split server into individual functions
- Scales automatically and independently
- Only runs when called (Lower potential costs)

AWS Lambda

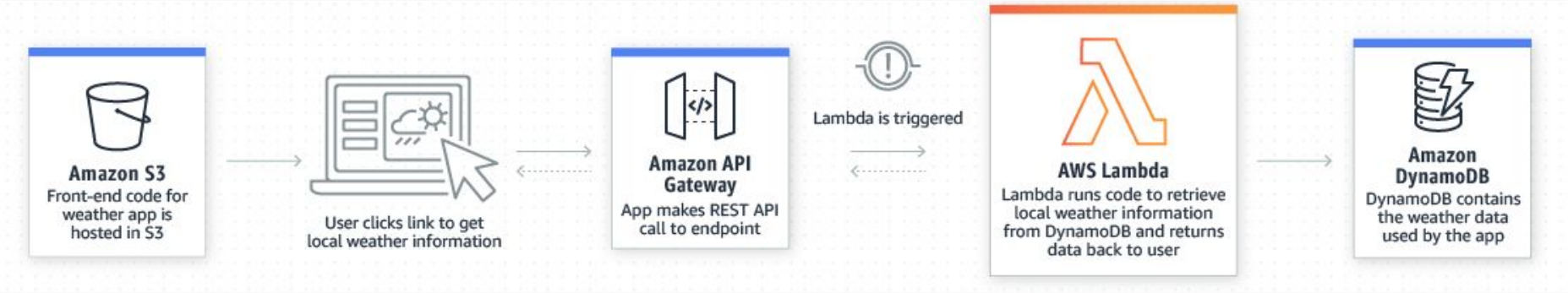
AWS Lambda

- Amazon’s serverless (FaaS) platform
- Simplifies building on-demand applications
- Event driven (API, time, storage/dB events, IoT)
- Supports numerous languages (JS, Py, Java, C#, etc…)
- Billed by hour / metered by millisec

Real-time File Processing Example

Mobile Backend Example

Web Application Example



AWS Lambda Walk-Thru

Severless Framework

Serverless Framework

- Free/Open source
- Written in Node
- Automates many of the AWS Lambda build processes
- No need to log into AWS console

Serverless Framework CLI Commands

- Install
  - `npm i serverless -g`
- Start from a template
  - `Serverless create --template aws-nodejs --path-to-dir my-service-name`
- Deploy
  - `Serverless deploy`
- Remove from cloud
  - `Serverless remove`

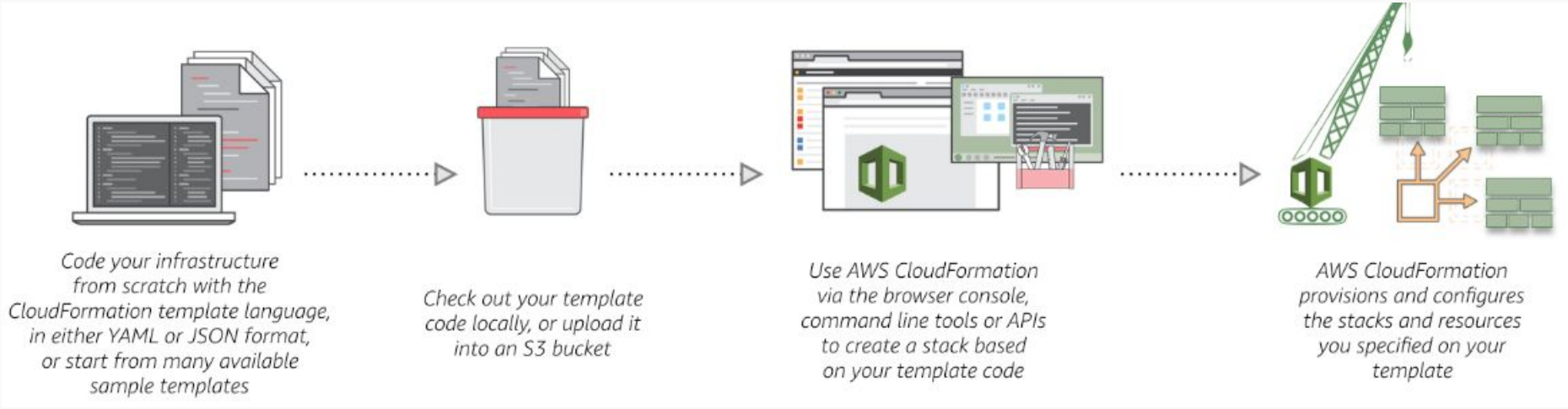
Serverless Framework - How it works

- CloudFormation template created from serverless.yml.
- If no Stack, it is created with no resources except an S3 Bucket, to store zip files of your Function.
- your code then packaged into zip files.
- Serverless fetches the hashes for all files of the previous deployment (if any) and compares them against the hashes of the local files.
- Terminates deployment if files are same.
- Zip files of your Functions' code are uploaded to your Code S3 Bucket.
- Any IAM Roles, Functions, Events and Resources are added to the AWS CloudFormation template.
- The CloudFormation Stack is updated with the new CloudFormation template.
- Each deployment publishes a new version for each function in your service.

Brief Stop - AWS CloudFormation

- CloudFormation is that it is a tool from AWS that allows you to spin up resources effortlessly.
- You define all the resources you want AWS to spin up in a blueprint document, click a button, and then AWS magically creates it all.
- This blueprint is called a template in CloudFormation speak.
- CloudFormation makes sure that dependent resources in your template are all created in the proper order.

How CloudFormation Works



Code your infrastructure from scratch with the CloudFormation template language, in either YAML or JSON format, or start from many available sample templates

Check out your template code locally, or upload it into an S3 bucket

Use AWS CloudFormation via the browser console, command line tools or APIs to create a stack based on your template code

AWS CloudFormation provisions and configures the stacks and resources you specified on your template

Serverless Framework Walk-Thru