IT/Dev CONNECTIONS

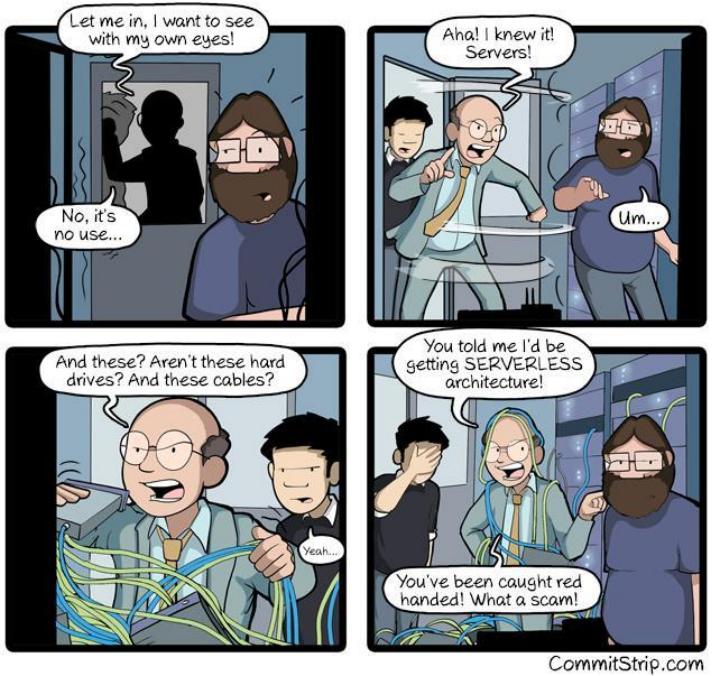
Defining Serverless APIs (AWS)

Turner Houghton
Junior Cloud Engineer
Home Technology Services

Delivered by KNect365 TMT an Informa business

What is Serverless?

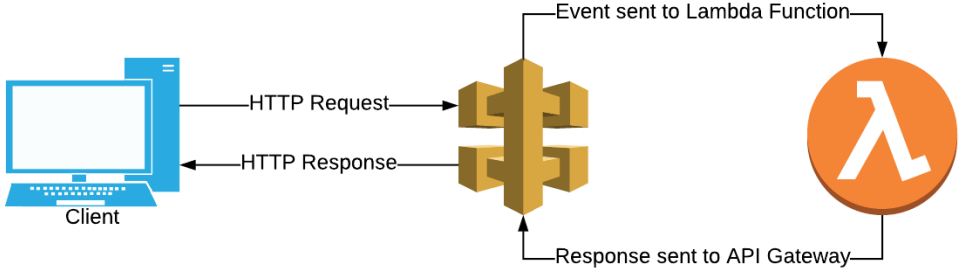
- Servers are still there, you just don't have to worry about them
- Primarily built around AWS Lambda
  - Run code on demand
  - Access other AWS Services using the SDK
- Built with API Gateway and scalable data stores (DynamoDB, RDS Aurora, S3)



Pros and Cons

- Pros
  - Great for high-variance applications
  - Billed by use
  - Don't have to worry about servers, can focus on business logic
  - Easy to deploy and monitor
- Cons
  - Maximum Lambda execution time is 15 minutes, API Gateway is 30 seconds
  - Cold starts are inevitable
  - Not ideal for constantly running processes

How it works



- Users use HTTP requests to interact with API Gateway endpoint
- API Gateway forwards event to Lambda function
- Lambda function does what you define it to do with event
  - Interact with other APIs
  - Access other AWS Services/data stores
  - Perform business logic
  - Define HTTP response to send back to API
- Lambda function finishes executing, returns response to API Gateway
- API Gateway sends HTTP response back to user

Data Stores

- DynamoDB
  - NoSQL Key-value store
  - Pay for read/write access + total storage
  - Recommended for simple applications
- Aurora (RDS)
  - MySQL and PostgreSQL compatible
  - Requires use of a Virtual Private Cloud (VPC)
  - Lambdas in a VPC have much worse cold starts
  - Now have Serverless Aurora
- S3
  - Document/file storage
  - Other services can interact with data in S3

Authentication

- API Gateway has 3 authentication types built in:
  - IAM Credentials
  - Lambda Function
  - Cognito User Pool Authorizer
- Cognito User Pools
  - User info is stored in Cognito
  - Supports groups and custom attributes
  - Easy to set up and integrate into mobile and web apps
  - Use API Gateway authorizer to restrict access to Cognito users using tokens

API Structure

- How many Lambda functions should I define?
  - Monolithic vs single-purposed
  - Single purposed is better performance and easier to understand
  - Monolithic is easier to deploy and "pre-warm"
- What API Gateway integration type should I use?
  - Lambda vs Lambda Proxy - **use Lambda Proxy**
  - Lambda Proxy integration is less customizable but easier to define
  - Lambda Integration uses more CloudFormation resources - max API size is 40-60 endpoints without splitting

Show Simple API

Common Problems and Solutions

- How can I allow my users to upload and view files?
  - You can do file uploads through API Gateway, but only up to 6mb
  - Give your users temporary S3 access by using pre-signed urls
- How can I give users roles and privileges in my application?
  - Create a user pool group for each role, check the user's roles in your lambda function
  - Groups can override each other using precedence levels
- How can I scalably host a web application to consume my API?
  - Use S3 static sites in combination with Amazon Route53
  - If you need https, you can create a CloudFront distribution with a free SSL certificate from Amazon Certificate Manager

AppSync

- New service released in February 2018
- Based on GraphQL
  - Single endpoint
  - Use special syntax to create and request data
- Users can subscribe to changes to get updates in real time
- Has a built in offline data store
  - Users can interact with and update data offline
  - When user reconnects, data is automatically updated and their changes are pushed to the application
- Great for user-focused PWA's and mobile applications
- Requires GraphQL knowledge to be an effective option

Frameworks

- CloudFormation
  - AWS infrastructure-as-code
  - YAML or JSON documents
- Serverless Framework
  - Easy to get started
  - Good documentation
  - Not great for complex architectures
- AWS Serverless Application Model (SAM)
  - Transformation layer in CloudFormation
  - CloudFormation with easier to define Lambda functions
  - Can be deployed just like any CloudFormation document
  - Ideal for teams in combination with CI/CD systems

Links

- CloudFormation Templates: https://github.com/tqhoughton/cloudformation-templates
- Serverless Demo: https://test.turnerhoughton.com
- LinkedIn: https://linkedin.com/in/turnerhoughton
- Questions? https://reddit.com/r/aws
- Commitstrip: http://www.commitstrip.com/en/2017/04/26/servers-there-are-no-servers-here/