Go Serverless

Erin Commarato

Learn how to develop API architectures without a server using AWS and Serverless Framework.

InspiringApps

Located on Pearl Street in
Downtown Boulder

Over 10 years old

Well over 100 apps for Android,
iOS, Echo and the web

Go Serverless

Intro

Serverless

Does it really mean you don’t need servers anymore?

NORDICAPIS.COM

Go Serverless

Not exactly serverless

- A cloud provider manages allocation of resources and servers

SHUTTERSTOCK / AMAZON

Not exactly serverless

- A cloud provider manages allocation of resources and servers
- You just pay for the amount of resources consumed by your application

Not exactly serverless

- A cloud provider manages allocation of resources and servers
- You just pay for the amount of resources consumed by your application
- Called FaaS or **Functions as a Service**

SHUTTERSTOCK / AMAZON

Not exactly serverless

- A cloud provider manages allocation of resources and servers
- You just pay for the amount of resources consumed by your application
- Called FaaS or Functions as a Service
- Not to be confused with PaaS or SaaS

SaaS vs. PaaS vs. FaaS

SaaS

Software as a Service

- A service that is typically always running
- Applications managed by a third party
- Examples: Gmail, Salesforce, Dropbox

Go Serverless with AWS

SaaS

Software as a Service

- A service that is typically always running
- Applications managed by a third party
- Examples: Gmail, Salesforce, Dropbox

PaaS

Platform as a Service

- A service that is typically is running at all times
- Provides a platform for hosting
- Scaling unit is a server
- Examples: AWS Elastic Beanstalk, Heroku, Windows Azure

SaaS

Software as a Service

- A service that is typically always running
- Applications managed by a third party
- Examples: Gmail, Salesforce, Dropbox

PaaS

Platform as a Service

- A service that is typically is running at all times
- Provides a platform for hosting
- Scaling unit is a server
- Examples: AWS Elastic Beanstalk, Heroku, Windows Azure

FaaS

Functions as a Service

- Does not run unless it is needed
- Abstracts the idea of a server away from the developer
- Scaling unit is a single execution
- Examples: AWS Lambda, IBM Bluemix OpenWhisk

Advantages of Serverless Architecture

Advantages of Serverless Architecture

Event-driven
Fixed costs are transformed into operation only costs

Advantages of Serverless Architecture

Event-driven
Fixed costs are transformed into operation only costs

Micromanaged
You don’t have to think about internal system administration processes

Advantages of Serverless Architecture

Event-driven
Fixed costs are transformed into operation only costs

Micromanaged
You don’t have to think about internal system administration processes

Speed
Reduces the development and deployment costs (faster time to market)

Advantages of Serverless Architecture

Event-driven
Fixed costs are transformed into operation only costs

Micromanaged
You don’t have to think about internal system administration processes

Speed
Reduces the development and deployment costs (faster time to market)

Auto-scaling
It is scalable and fault tolerant by design

Disdvantages of Serverless Architecture

Disadvantages of Serverless Architecture

Problems due to third-party API system
Vendor control and vendor lock-in

Disadvantages of Serverless Architecture

Problems due to third-party API system
Vendor control and vendor lock-in

Lack of operational tools
Dependent on vendors for debugging and monitoring tools

Debugging distributed systems is challenging

Disadvantages of Serverless Architecture

Problems due to third-party API system
Vendor control and vendor lock-in

Lack of operational tools
Dependent on vendors for debugging and monitoring tools

Debugging distributed systems is challenging

Architectural complexity
Mini-monoliths are easy to build

Disadvantages of Serverless Architecture

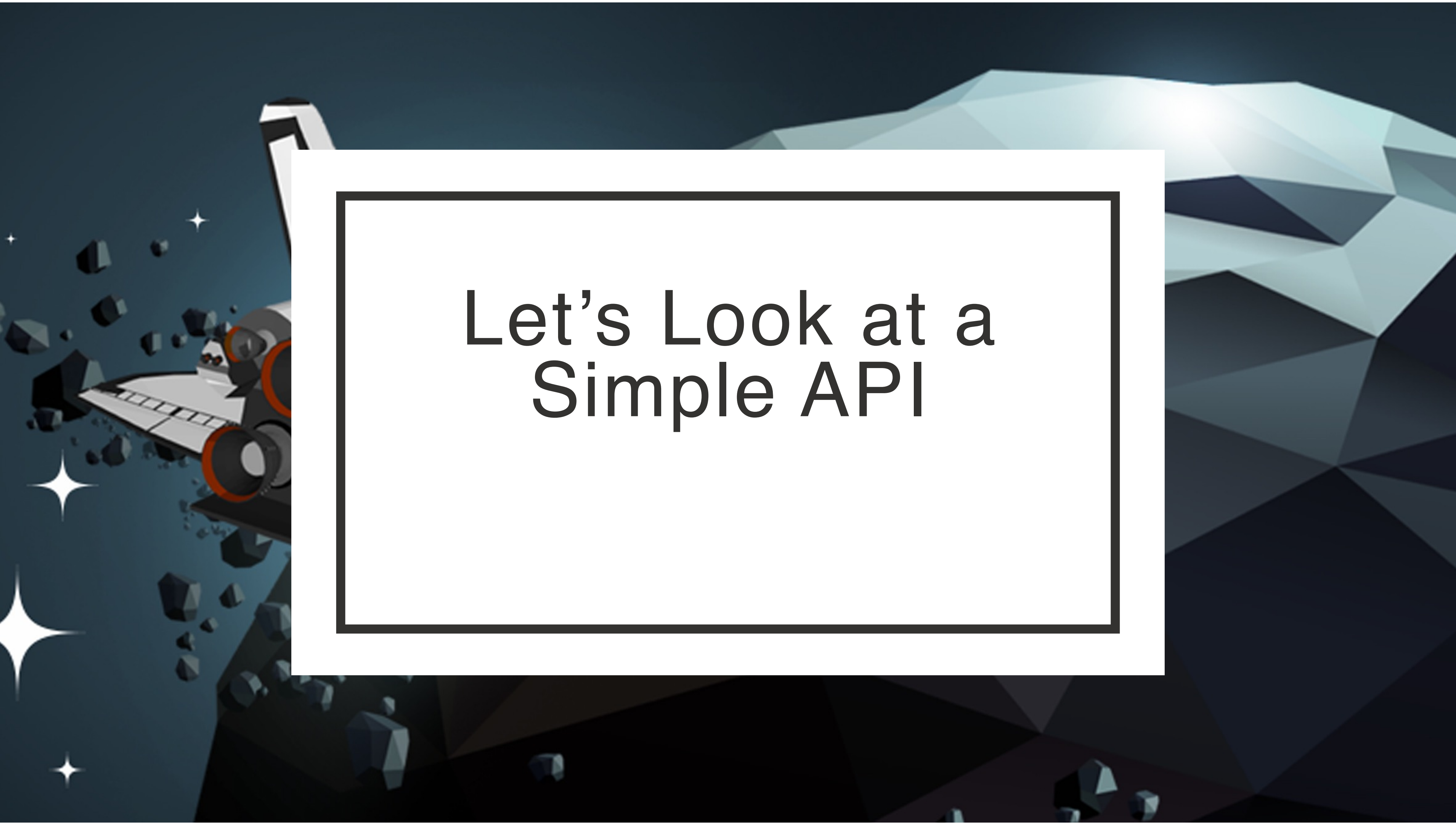
Problems due to third-party API system
Vendor control and vendor lock-in

Lack of operational tools
Dependent on vendors for debugging and monitoring tools

Debugging distributed systems is challenging

Architectural complexity
Mini-monoliths are easy to build

Implementation drawbacks
Deployment may mean deploying a single FaaS artifact for each function



Let’s Look at a Simple API

FaasCats API Overview

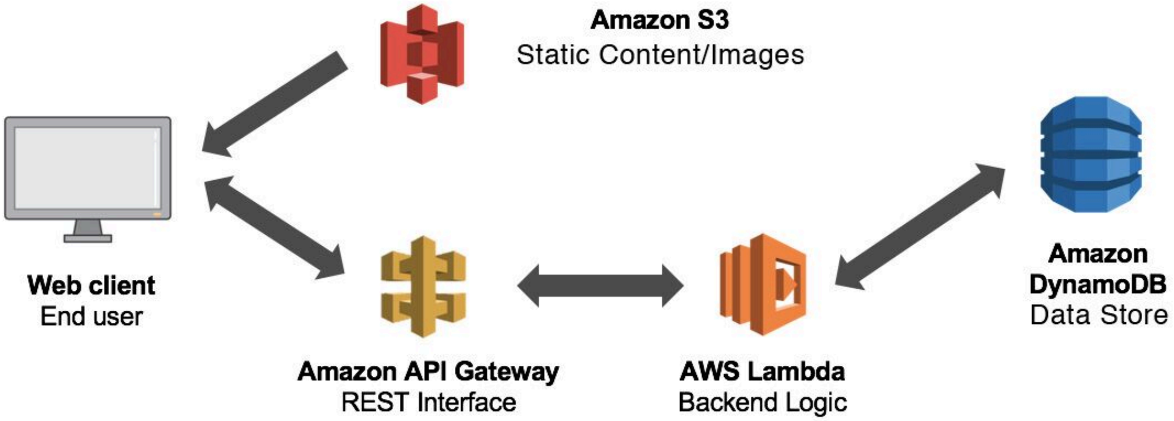
Objective:
Build a serverless API on AWS

Features:
Using the API, a user can Create, Read, Update and Delete famous cats

Using a web-based interface, a user can search for cats in the API

FaasCats API Overview

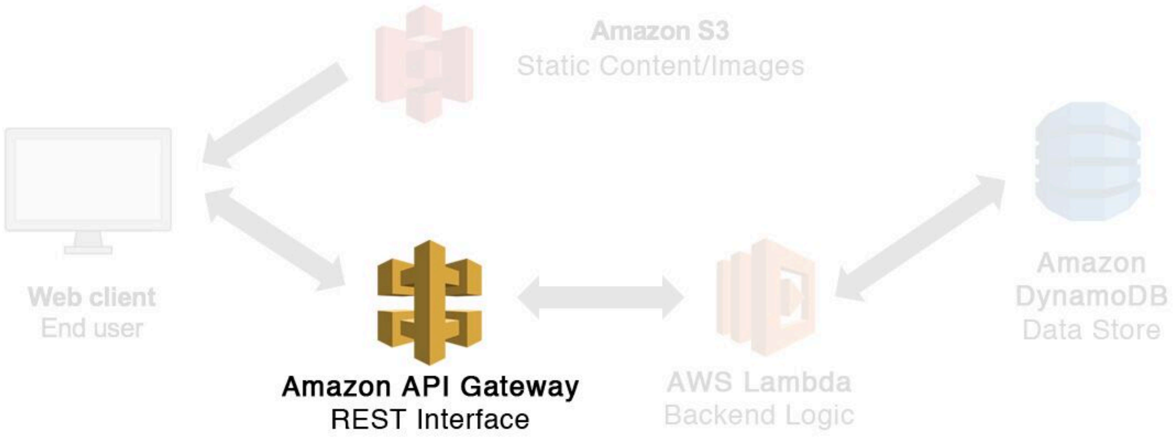
Build a serverless API



Serverless Framework can help us tie this all together

FaasCats API Overview

API Gateway



Directs traffic to different Lambda functions depending on endpoint

API Gateway

- Create, maintain and monitor APIs at scale

API Gateway

- Create, maintain and monitor APIs at scale
- Acts as a “front door” for your applications

Go Serverless with AWS

API Gateway

- Create, maintain and monitor APIs at scale
- Acts as a “front door” for your applications
- Handles thousands of concurrent calls for you automatically

Go Serverless with AWS

API Gateway

- Create, maintain and monitor APIs at scale
- Acts as a “front door” for your applications
- Handles thousands of concurrent calls for you automatically

Use Cases

Route to Lambda functions

Create RESTful endpoints to existing services

API Gateway

Cost:

Free Tier (first 12 months)

1 million API calls/mo.

Regularly

$3.50 per 1 millon API calls

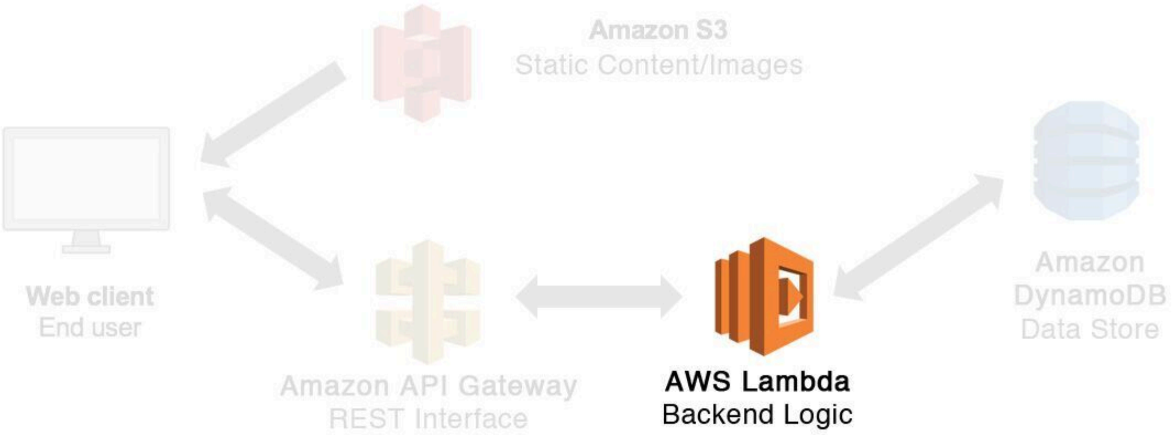
API Gateway

Dashboard

Go Serverless with AWS

FaasCats API Overview

AWS Lambda



The “brains” of our application
Code handles data input and output

AWS Lambda

- Run code without thinking about servers

Go Serverless with AWS

Go Serverless with AWS

AWS Lambda

- Run code without thinking about servers
- Code only runs when triggered

AWS Lambda

- Run code without thinking about servers
- Code only runs when triggered
- Scales with workload

AWS Lambda

- Run code without thinking about servers
- Code only runs when triggered
- Scales with workload

Use Cases

- Interact with a database
- Process uploaded S3 objects automatically

Go Serverless with AWS

AWS Lambda

Cost:

First 1 million requests / mo. free

Then

$0.20 per 1 million requests thereafter

Go Serverless with AWS

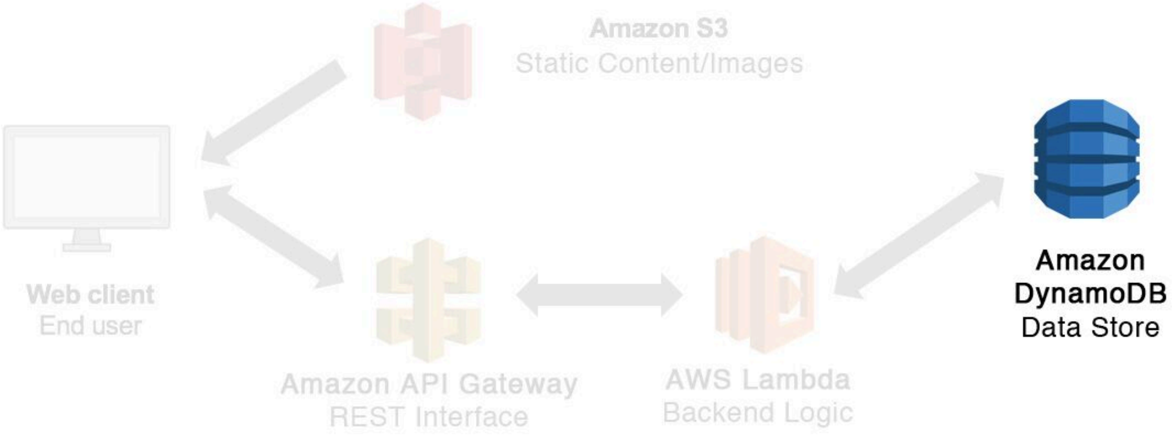
AWS Lambda

Dashboard

Go Serverless with AWS

FaasCats API Overview

DynamoDB



Database storage
Stores cats and their images

DynamoDB

- Fully managed noSQL cloud database

DynamoDB

- Fully managed noSQL cloud database
- Highly scaleable up or down as volume increases or decreases

DynamoDB

- Fully managed noSQL cloud database
- Highly scaleable up or down as volume increases or decreases
- Integrates with AWS Lambda for event-driven programming

Go Serverless with AWS

DynamoDB

- Fully managed noSQL cloud database
- Highly scaleable up or down as volume increases or decreases
- Integrates with AWS Lambda for event-driven programming

Use Cases:

- Big data storage/retrieval
- Structured or unstructured data

DynamoDB

Cost:

25 GB Storage plus 200 million requests / mo.

free

Then

Varies based on usage and provisioning

Go Serverless with AWS

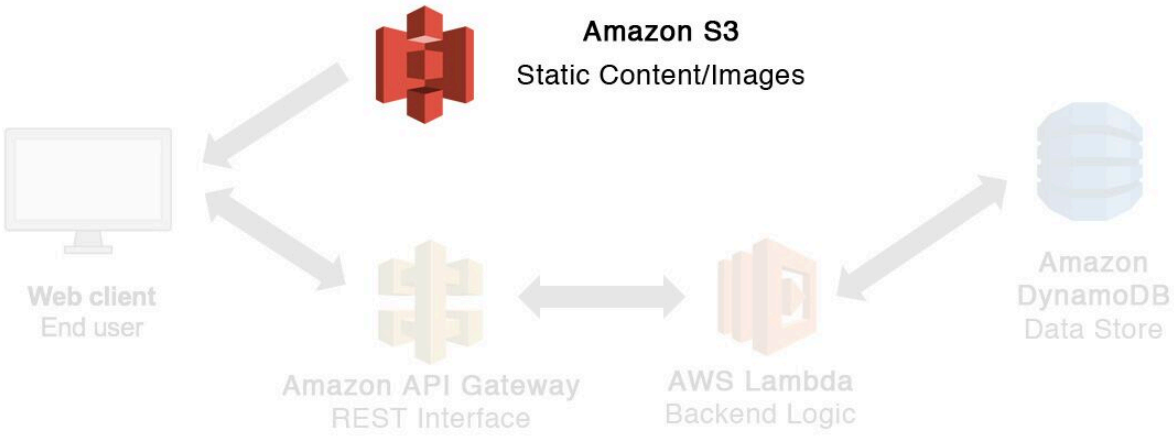
DynamoDB

Dashboard

Go Serverless with AWS

FaasCats API Overview

Amazon S3



Physical file storage
Also used to host a simple webpage used to interact with the app

Amazon S3

- Highly available object storage

Go Serverless with AWS

Amazon S3

- Highly available object storage

Use Cases

- Backup and recovery
- Data archiving
- Static website hosting

Amazon S3

Cost:

5 GB Storage free

Then

Varies based on usage starting at $0.023 / GB

for the first 50 TB storage

Amazon S3

Dashboard

Go Serverless with AWS

Let’s write some code!

Next Steps

Next Steps

- Protected Endpoints

Next Steps

- Protected Endpoints
- Throttling

Go Serverless with AWS

Go Serverless with AWS

Next Steps

- Protected Endpoints
- Throttling
- Scaling

Next Steps

- Protected Endpoints
- Throttling
- Scaling
- Validation

Q&A

Additional Resources

Get the code
https://github.com/InspiringApps/HackCU18-AWS

Serverless Framework documentation
https://serverless.com/framework/docs/providers/aws/

AWS Getting Started
https://aws.amazon.com/getting-started/