

Serverless Architecture

Mischa Spiegelmock
JetBridge, Inc.

About Me

- From Berkeley, California
- Co-founder of JetBridge in SF and Kyiv
- Do computer shit
- Been doing computer shit a long time
- I'm a **developer**

This Talk

- Ops from a developer's point of view
- Cloud-native application architecture
- Pros/cons of serverless applications

Ops

As a developer



What's the ?

Just somebody else's
computers?

- Use economies of scale of cloud providers
- Worry about higher-level issues more related to business
- Increase developer “leverage” 💪
- Don’t solve already solved problems (e.g. deploying web applications)







Serverless

Functions As A Service

“FaaS”

AWS Lambda

FaaS

- Write a function
- Deploy it “somewhere”
- Invoke it “somehow”

Write A Function

How it works

Run



"Hello 🌎!"

```
1 exports.handler = (event, context, callback) => {  
2     // Succeed with the string "Hello world!"  
3     callback(null, 'Hello 🌎!');  
4 };
```

Deploy It Somewhere

```
Last login: Mon Feb 25 17:12:46 on ttys015
```

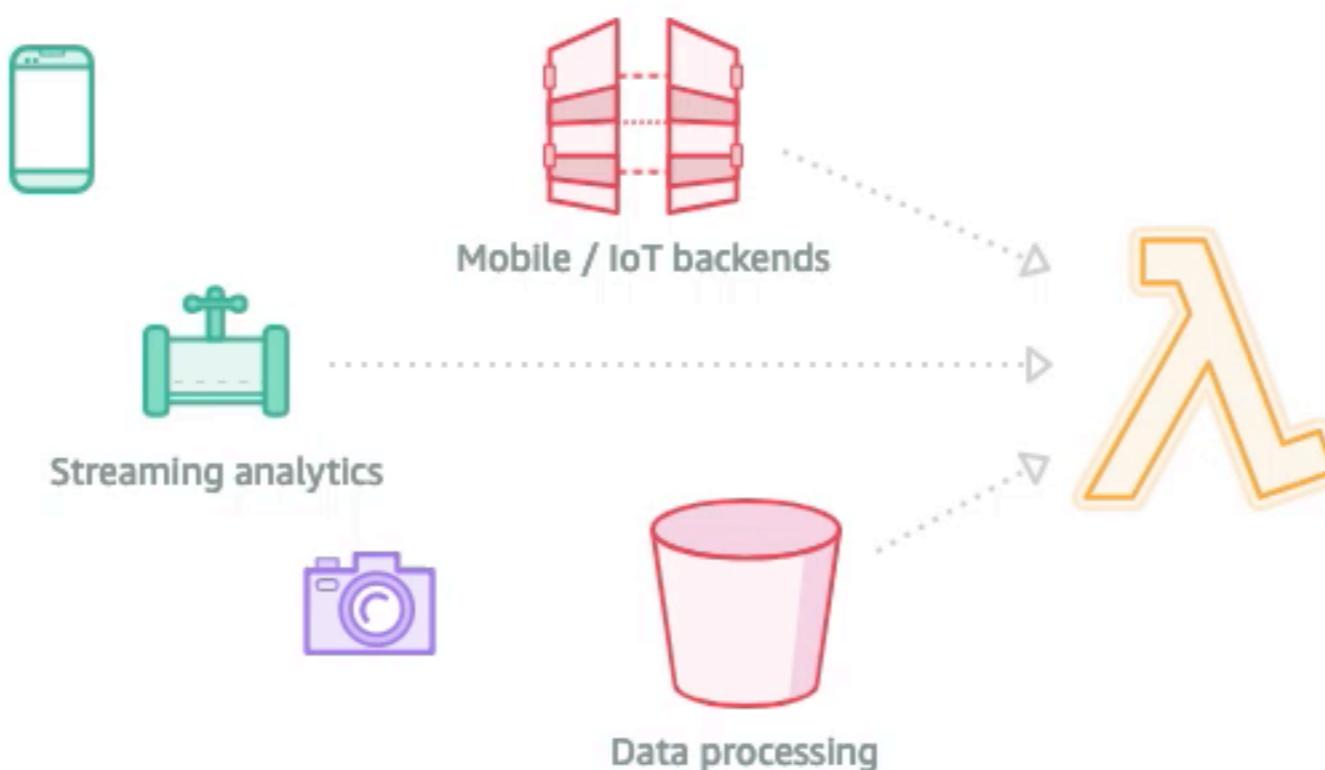
```
→ ~ slsp
```

```
Last login: Thu Feb 21 18:04:44 on console
```

```
→ crmservice git:(heroku-single-project) ✘
```

Invoke It Somehow

How it works

[Previous](#)[Next: Scale seamlessly](#)

6:49:04 PM: "Hello !"

Lambda responds to events

Once you create Lambda functions, you can configure them to respond to events from a variety of sources. Try sending a mobile notification, streaming data to Lambda, or placing a photo in an S3 bucket.

So what?

For Free:

High Availability

Not My Problem

Versioning

Every function gets a version

Autoscaling

1,000 polling concurrent invocations

Free-ish

Free as in 

Function-Level Permissions

Least privilege resource access

Filter events

all 2019-02-24 (19:38:23)

Time (UTC +00:00)	Message
2019-02-25	
▶ 19:24:34	START RequestId: 67c09671-9369-432e-9762-74e47662cfb9 Version: \$LATEST
▶ 19:24:34	END RequestId: 67c09671-9369-432e-9762-74e47662cfb9
▶ 19:24:34	REPORT RequestId: 67c09671-9369-432e-9762-74e47662cfb9 Duration: 38.15 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 41 MB
▶ 19:24:34	START RequestId: bce0fba0-9390-4218-9a86-9557ba8f046c Version: \$LATEST
▶ 19:24:34	END RequestId: bce0fba0-9390-4218-9a86-9557ba8f046c
▶ 19:24:34	REPORT RequestId: bce0fba0-9390-4218-9a86-9557ba8f046c Duration: 12.40 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 41 MB
▶ 19:24:34	START RequestId: 942d2d7b-71fe-4f86-be43-bc449b0d0bb3 Version: \$LATEST
▶ 19:24:34	END RequestId: 942d2d7b-71fe-4f86-be43-bc449b0d0bb3
▶ 19:24:34	REPORT RequestId: 942d2d7b-71fe-4f86-be43-bc449b0d0bb3 Duration: 20.35 ms Billed Duration: 100 ms Memory Size: 128 MB Max Memory Used: 41 MB
▶ 19:24:42	START RequestId: d9d89820-fcd2-4178-bbc3-65dccde83dd5 Version: \$LATEST
▶ 19:24:42	END RequestId: d9d89820-fcd2-4178-bbc3-65dccde83dd5

ALARM 0 INSUFFICIENT DATA 0 OK 0

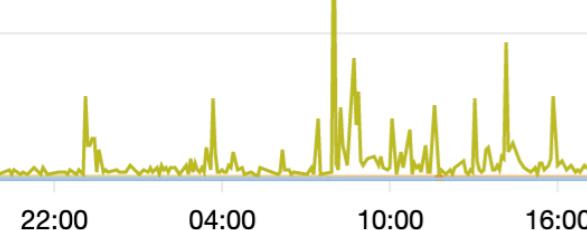
Invocations Sum

Various units

1.27k

635

1.00



- jb-contact-form-dev-unitFact...
- MakePyPIDep
- fingerprint-custom-domain-dev...
- salesPipe-prod-dispatch
- salesPipe-dev-handleSlackRe...
- costimator-prod-submit
- check_worker_queue_size
- monitor

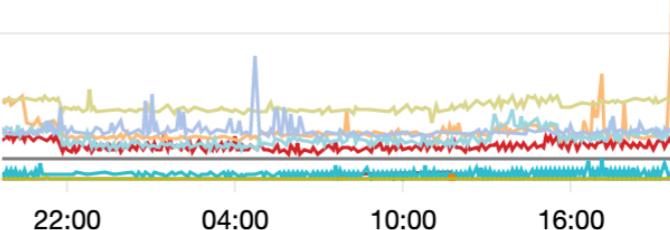
Duration Average

Various units

36.0k

18.0k

0.862



- jb-contact-form-dev-unitFact...
- MakePyPIDep
- fingerprint-custom-domain-dev...
- salesPipe-prod-dispatch
- salesPipe-dev-handleSlackRe...
- costimator-prod-submit
- check_worker_queue_size
- monitor

Errors Sum

Various units

3.00

1.50



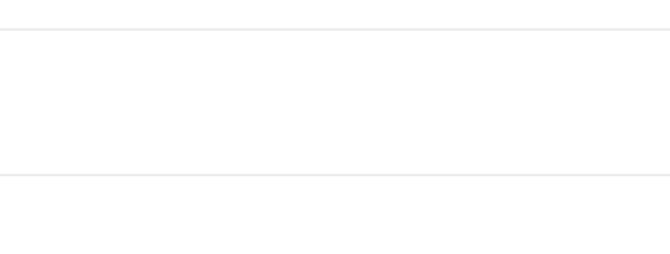
- jb-contact-form-dev-unitFact...
- MakePyPIDep
- fingerprint-custom-domain-dev...
- salesPipe-prod-dispatch
- salesPipe-dev-handleSlackRe...
- costimator-prod-submit
- check_worker_queue_size
- monitor

Throttles Sum

Various units

1.00

0.5



- jb-contact-form-dev-unitFact...
- MakePyPIDep
- fingerprint-custom-domain-dev...
- salesPipe-prod-dispatch
- salesPipe-dev-handleSlackRe...
- costimator-prod-submit
- check_worker_queue_size
- monitor

Building Blocks

- AWS Lambda
- Google Cloud Functions
- Kubeless
- Azure
- IBM



serverless framework

The easy and open way to build serverless applications



Serverless (AWS)

- Interface to CloudWatch Logs (logging)
- Interface to CloudFormation (IaC, deployments)
- Interface to API Gateway (HTTP)
- Interface to Parameter Store (secrets)
- Plugins

Additional Building Blocks

- S3 (Object store)
- CloudFront (CDN, TLS)
- RDS, Aurora Serverless, DynamoDB, AppSync
- Cognito (Authentication)
- IoT
- SNS/SES (SMS/Email)
- Amplify (Web & Mobile toolkits)
- Secrets Manager
- ...

Benefits To FaaS

- Deployments: solved
- Infrastructure packaged with application
- Autoscaling
- High availability
- Tight integration with other building blocks
- No DevOps, just Dev

Downsides To FaaS

- Vendor lock-in
- Platform limitations
- Raw power
- Shared hosting
- Local development environment
- Cold start time

FaaS Use Cases

- Microservices
- Triggers (IoT, Storage, Message Queue, Auth, Alarms..)
- Web Applications
- Webhook Processors
- Streams (WebSockets, DynamoDB, Kinesis)

FaaS Is Not For

- Raw compute power (CPU, RAM, GPU, Disk)
- Long-running tasks
- Streaming media
- Large (disk size) applications
- High startup time
- Non-linux/amd64
- Dedicated hosting requirements (compliance)
- Real-time
- Probably a lot of other scenarios

Something In The
Middle?

AWS Fargate

(managed containers)

Future Of DevOps



The End

Email me: mischa@jetbridge.com

mish.dev

spiegelmock.com

jetbridge.com