



Building Full-Stack Serverless Applications

Serverless-Stack.com

Serverless-Stack.com is a free comprehensive resource on building full-stack serverless applications on AWS.

Frank

@fanjiewang



Jay

@jayair

“Serverless computing, is a cloud computing code execution model in which the cloud provider fully manages starting and stopping of a function’s container as necessary to serve requests, and requests are billed by an abstract measure of the resources required to satisfy the request”

Turns out serverless
computing involves...



Servers!

But fully managed to
auto-scale, and billed by
the millisecond!

Why Serverless?

Cost

Low
Maintenance

Scale

What is a Full-Stack
Serverless App?

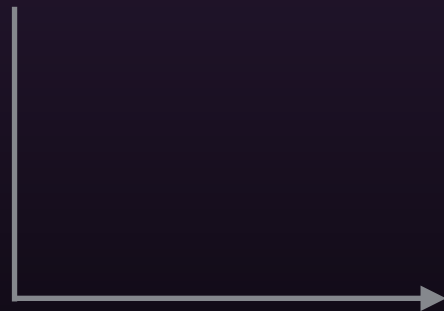
Static Site



S3



CloudFront



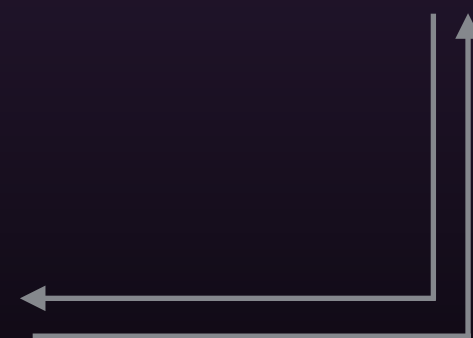
Functions



Lambdas



API Gateway

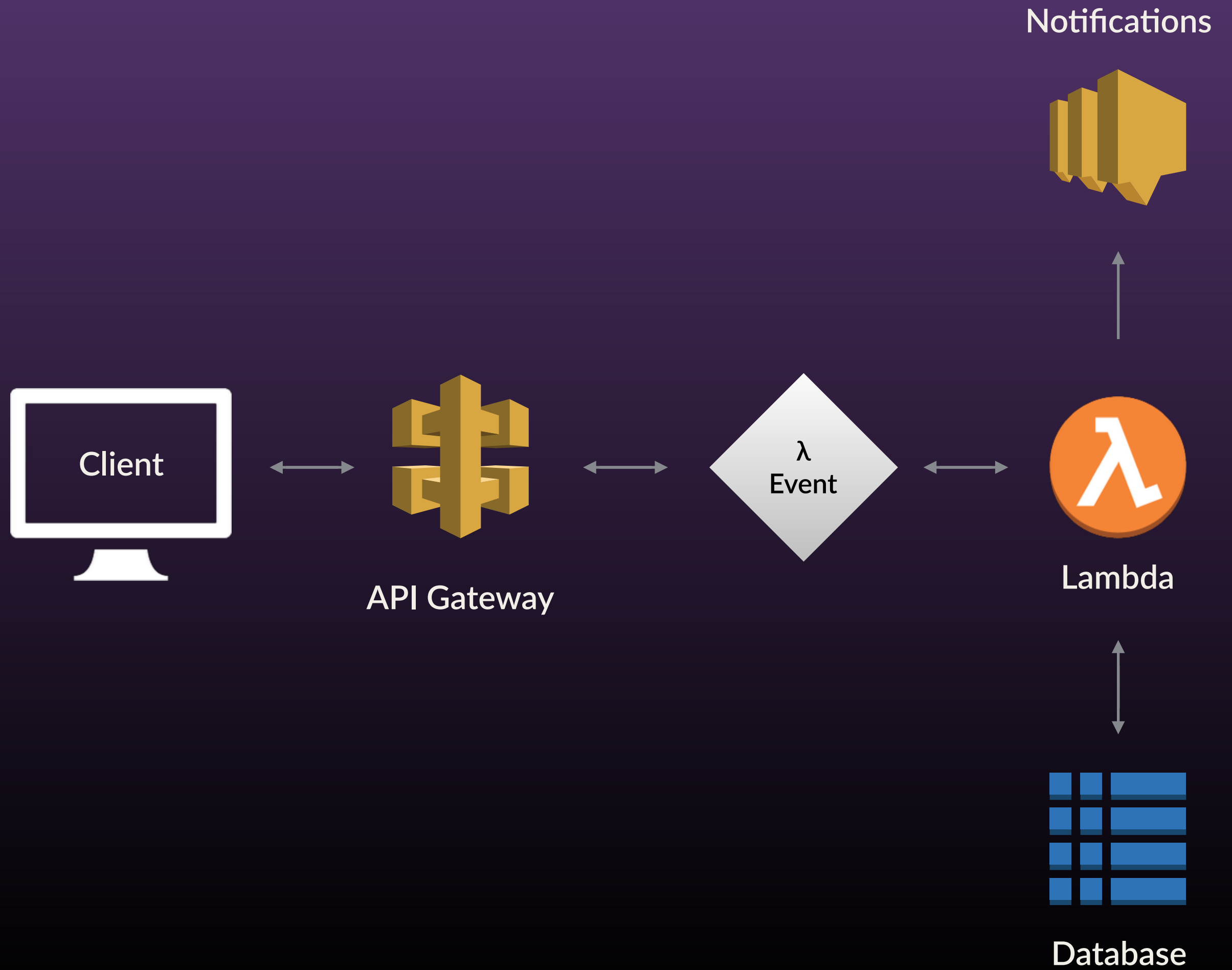


Client





Backend





Runs a function in a container

Supports JS, Python, Java, C#

Supports Node.js 4.3.2 & 6.10

Linux kernel 4.4.51

Container memory 128 - 1536MB

Ephemeral disk space 512MB

\$0.000000208 for 128MB - 100ms

```
// Initialization code
```

```
// Runs once per container creation
```

The diagram illustrates the mapping of code variables to their respective objects in a container environment. It features a code snippet with several variables highlighted in boxes, and labels above and below these boxes connected by vertical lines to indicate their meaning.

Code Snippet:

```
exports.myHandler = function(event, context, callback) {  
    // Do stuff  
    callback(Error error, Object result);  
}
```

Labels and Mappings:

- Name of the Lambda:** Points to `myHandler`.
- Event Object:** Points to `event`.
- Context Object:** Points to `context`.
- Callback Function:** Points to `callback`.
- Error Object:** Points to `Error error`.
- Result Object:** Points to `Object result`.



Triggered by events or streams

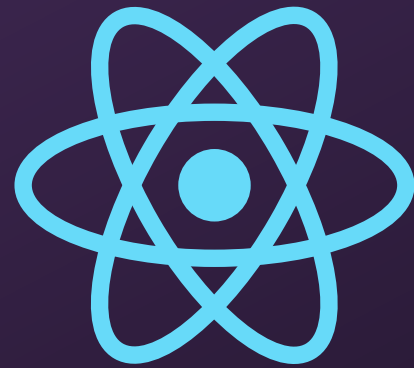
Function is “stateless”

Scales to match requests

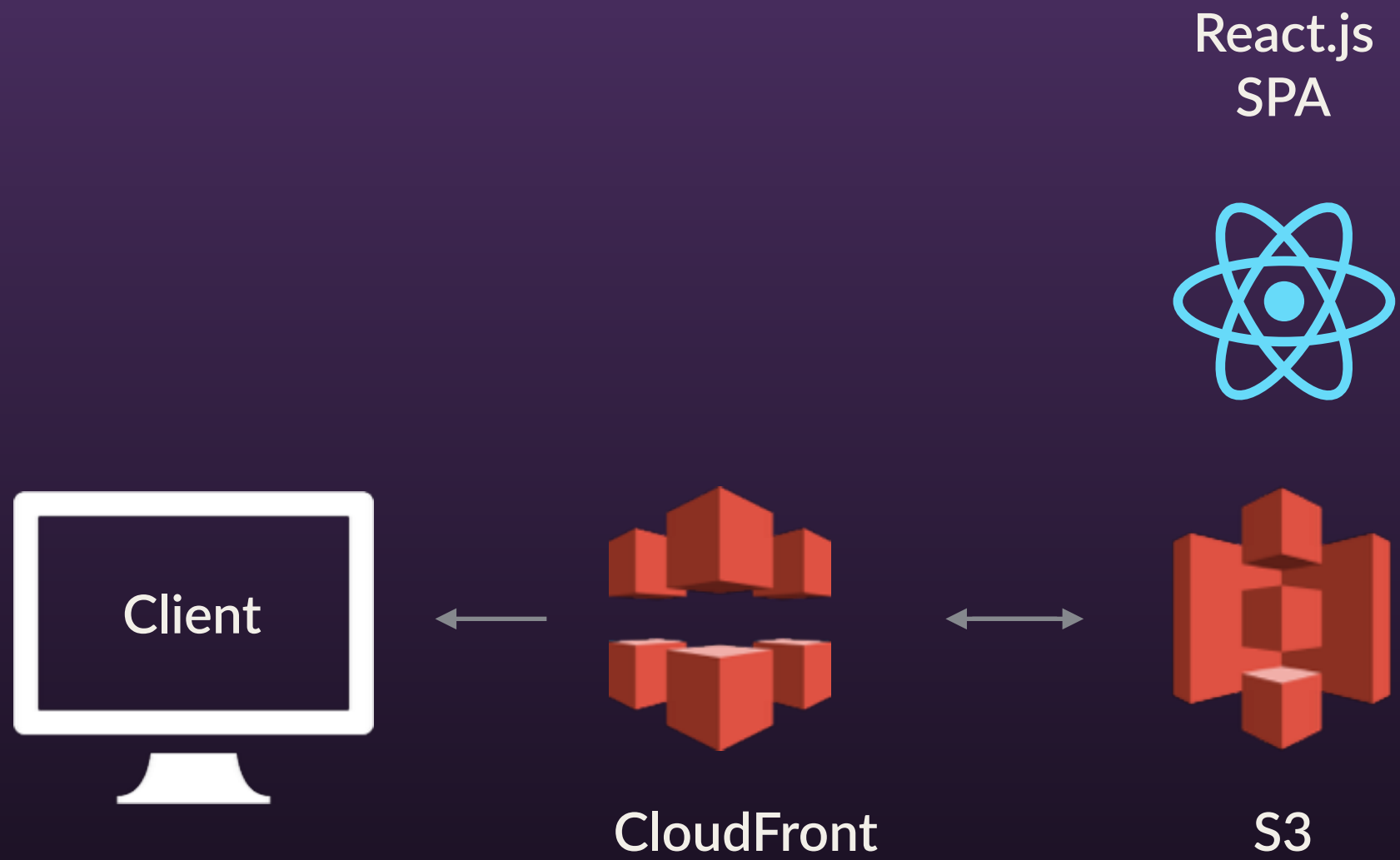
Concurrency limit of 1000

Timeout 300s, default 3s

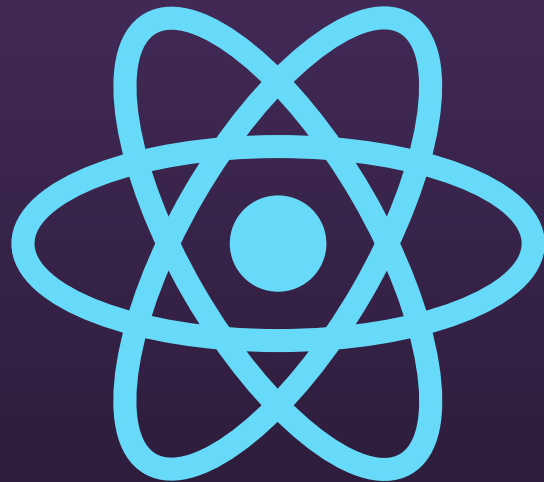
Executes till callback returns or call stack finishes or timeout



Frontend



React.js
SPA



+



S3 & CloudFront

Infinite scale

Edge delivery

Extremely cost-effective

Client/API separation

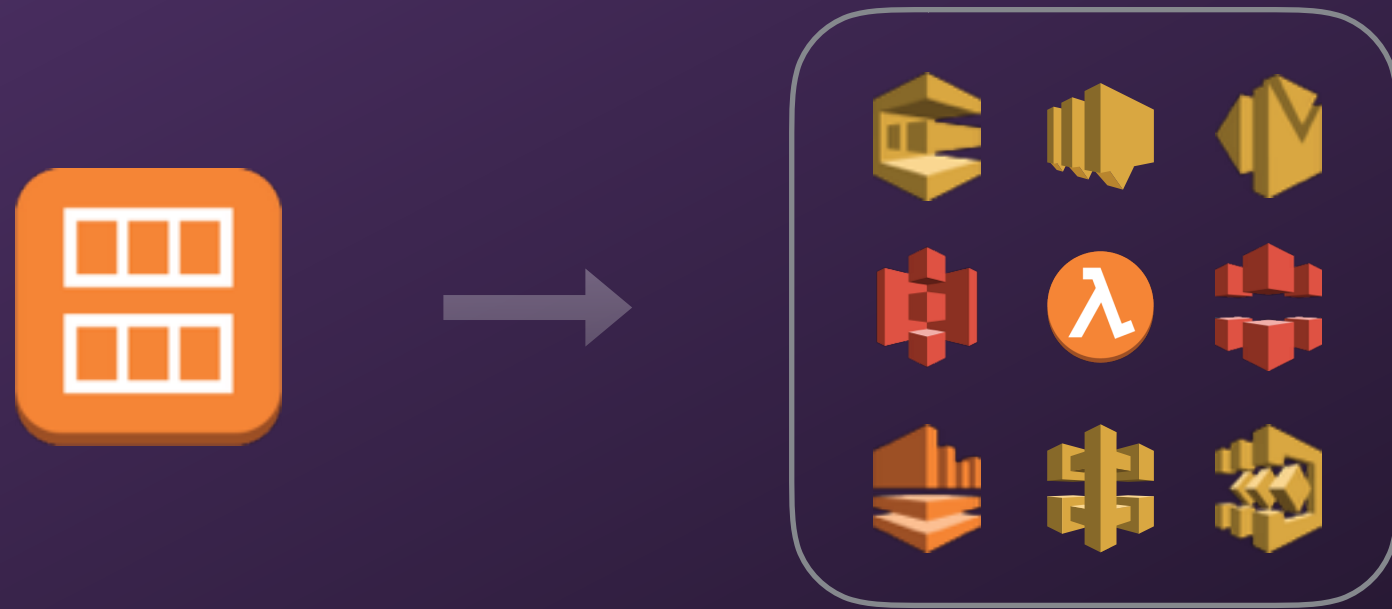


Things to look
out for



Lambda Cold Starts

✨ Warm every 5mins



Switch from Monolithic

✨ Offload a feature at a time



Tooling & Docs

✨ Very active community



SEO

✨ Pre-render or use marketing pages

Further Reading

Lambda Start Performance - <https://robertvojta.com/aws-journey-api-gateway-lambda-vpc-performance-452c6932093b>

Lambda Container Lifetime - <https://www.linkedin.com/pulse/aws-lambda-container-lifetime-config-refresh-frederik-willaert>

PHP, Ruby, and Go, on Lambda - <https://aws.amazon.com/blogs/compute/scripting-languages-for-aws-lambda-running-php-ruby-and-go/>

Questions?

Serverless-Stack.com

Frank

@fanjiewang



Jay

@jayair