goto; chicago

Serverless tales from the trenches

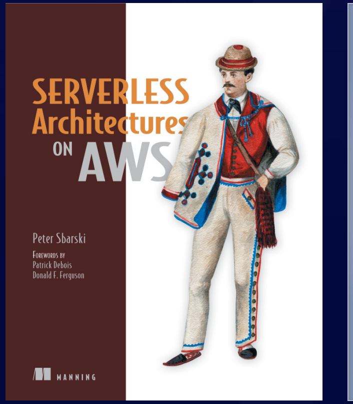
Peter Sbarski

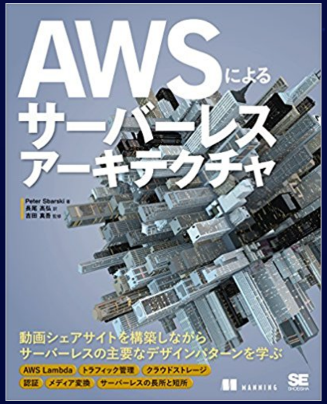
Peter Sbarski, PhD

AWS Community Hero

@sbarski

Author
Serverless Architectures on AWS
https://book.acloud.guru





VP Engineering | VP Content
A Cloud Guru
https://acloud.guru

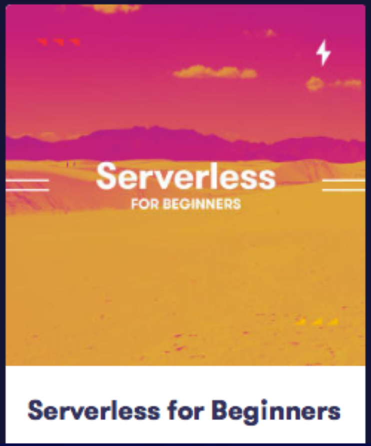
A CLOUD GURU

Organizer
Serverlessconf
https://serverlessconf.io

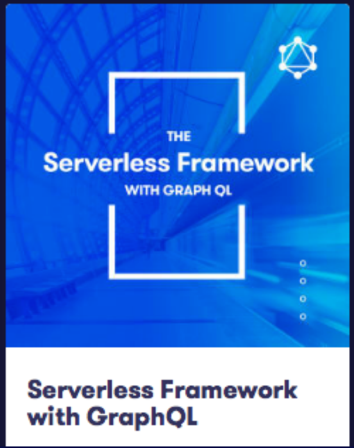
SERVERLESSCONF

TEACHING THE WORLD TO CLOUD

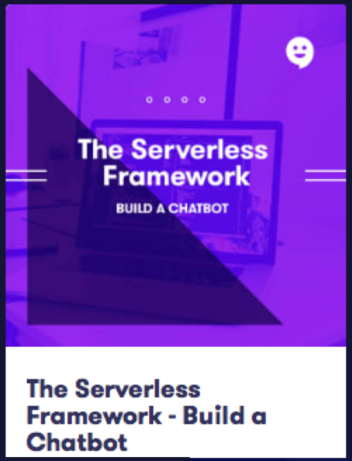
Join the 500,000+ engineers that have taken courses with A Cloud Guru.



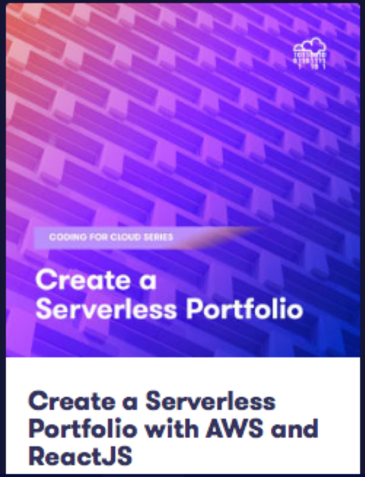
Serverless for Beginners



Serverless Framework with GraphQL



The Serverless Framework - Build a Chatbot



Create a Serverless Portfolio with AWS and ReactJS

AWS Lambda

Serverlessconf San Francisco

July 29 – August 1, 2018

https://sf.serverlessconf.io

Why Serverless?

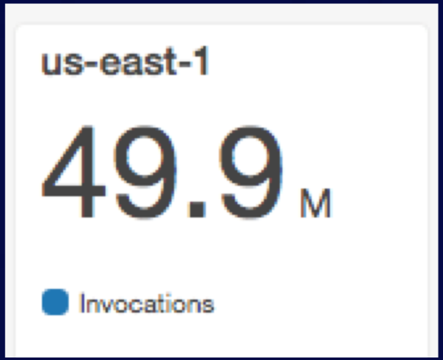
	IaaS	PaaS	Serverless
Unit of Scale	Virtual Machine or Container (Docker)	Application	Function
Fleet Operational Responsibility	Application developer	Shared between developer and vendor	Vendor only
Required Management & Maintenance	High – Operating System level	Medium – Application level	Low – function level
Billing	Per VM per minute or hour	Per VM per minute or hour	Per 100 milliseconds (continuous)
Impact of Idle Time	Economic loss when machines are idle or underused	Economic loss when machines are idle or underused	None – functions execute only when needed
Integration with other vendor services	Manual	Mixed	Automatic

Source: Serverless Design Patterns (T. Wagner, Y. Kiriaty, P. Sbarski)

algolia

Firebase

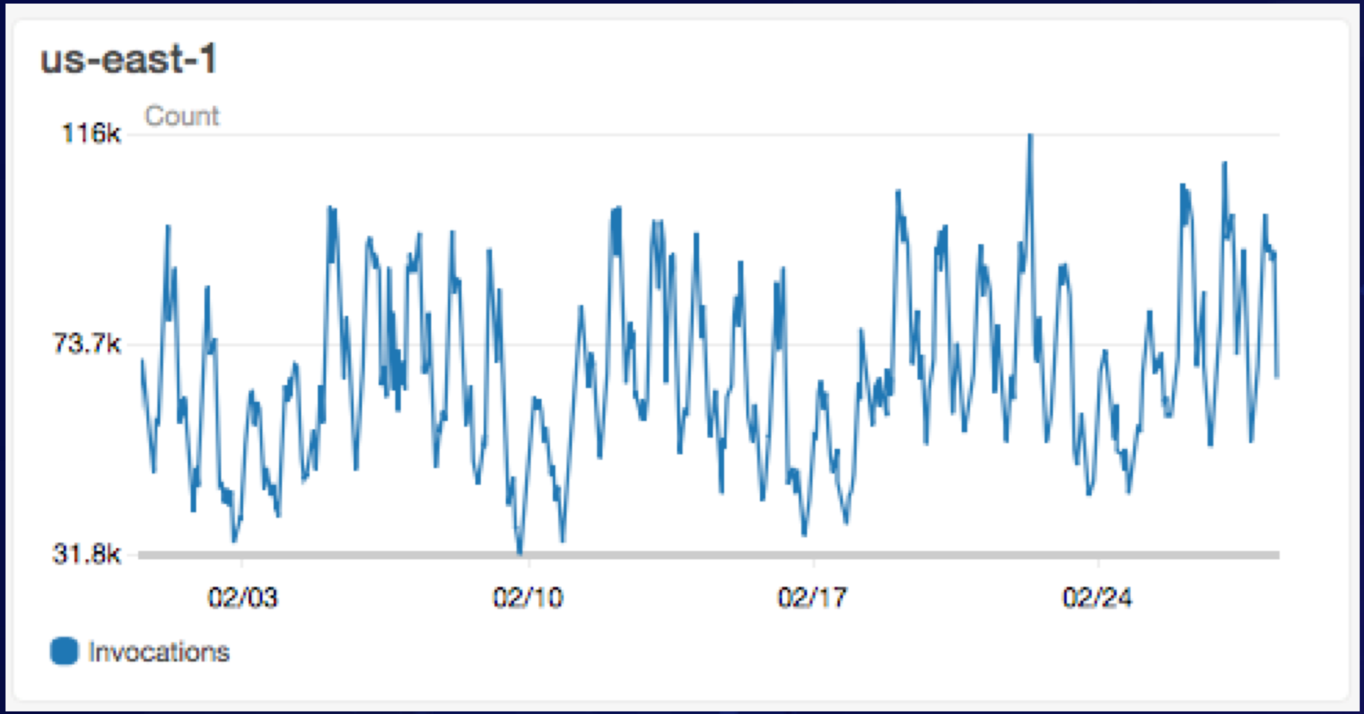
Auth0



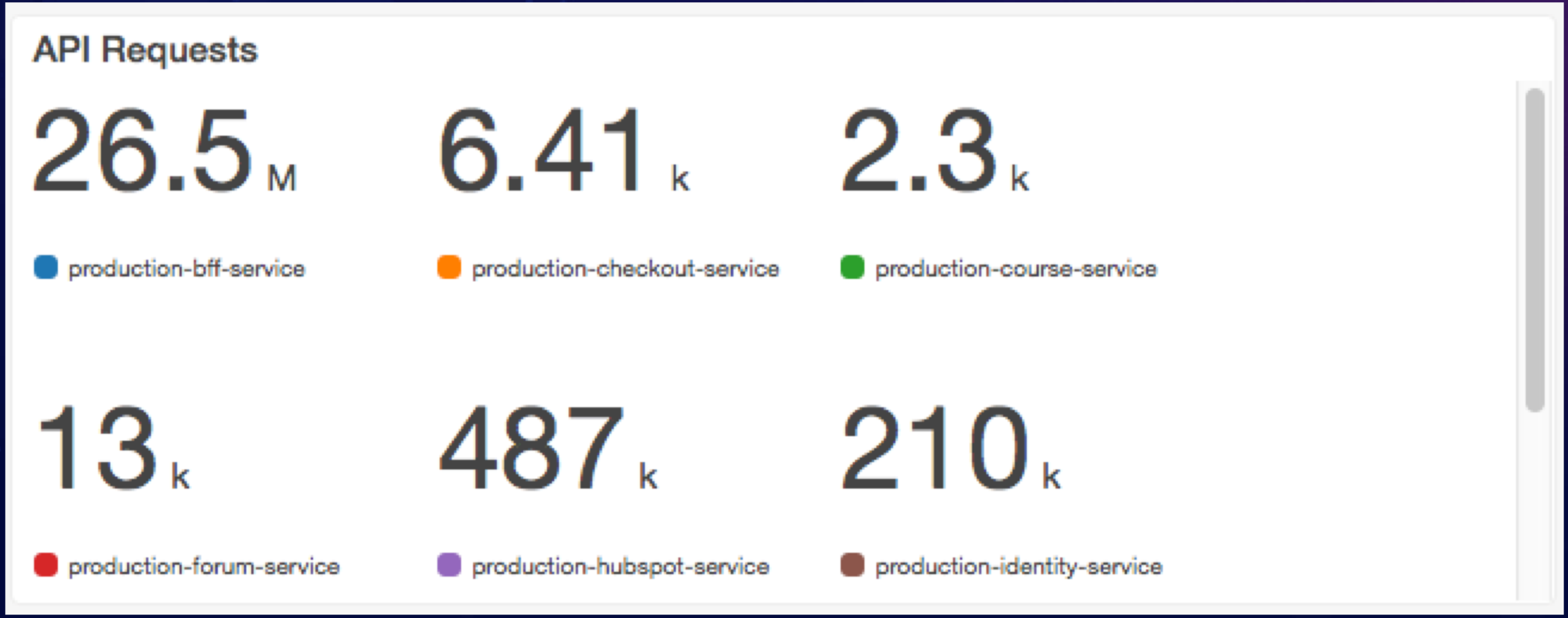
us-east-1

49.9 M

● Invocations



us-east-1



API Requests

26.5 M

● production-bff-service

6.41 k

● production-checkout-service

2.3 k

● production-course-service

13 k

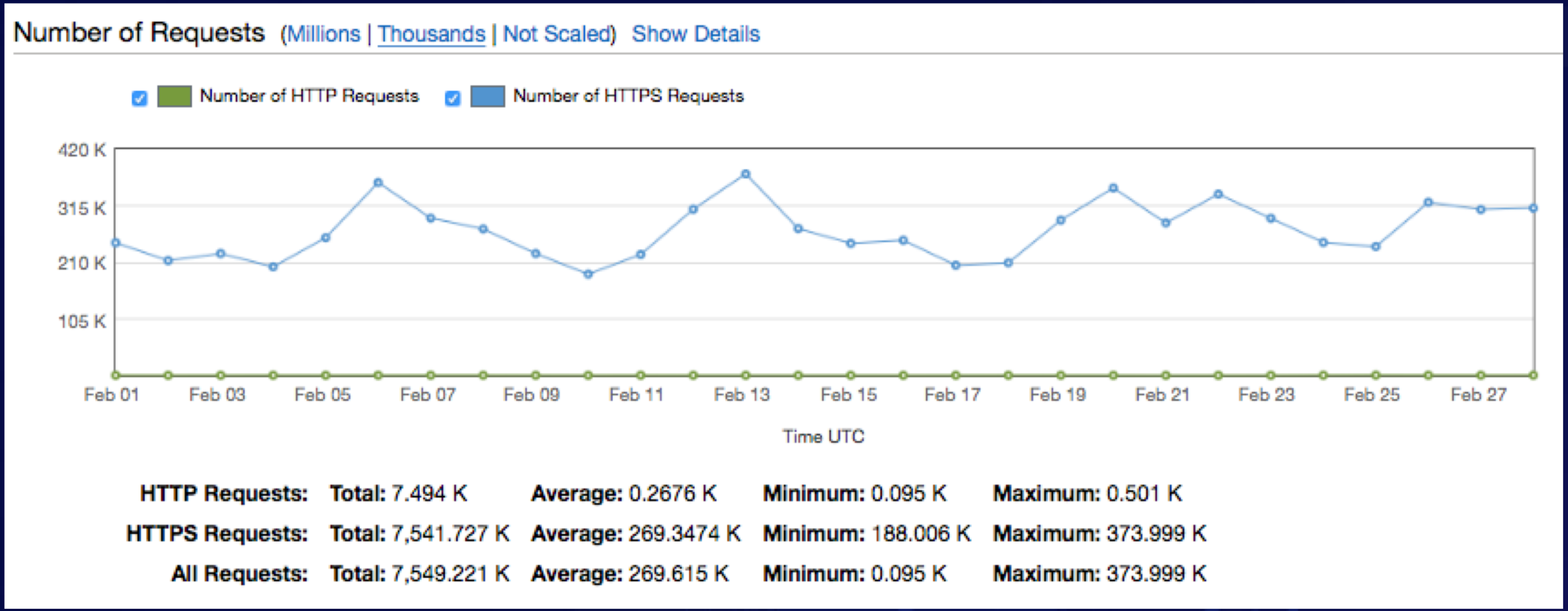
● production-forum-service

487 k

● production-hubspot-service

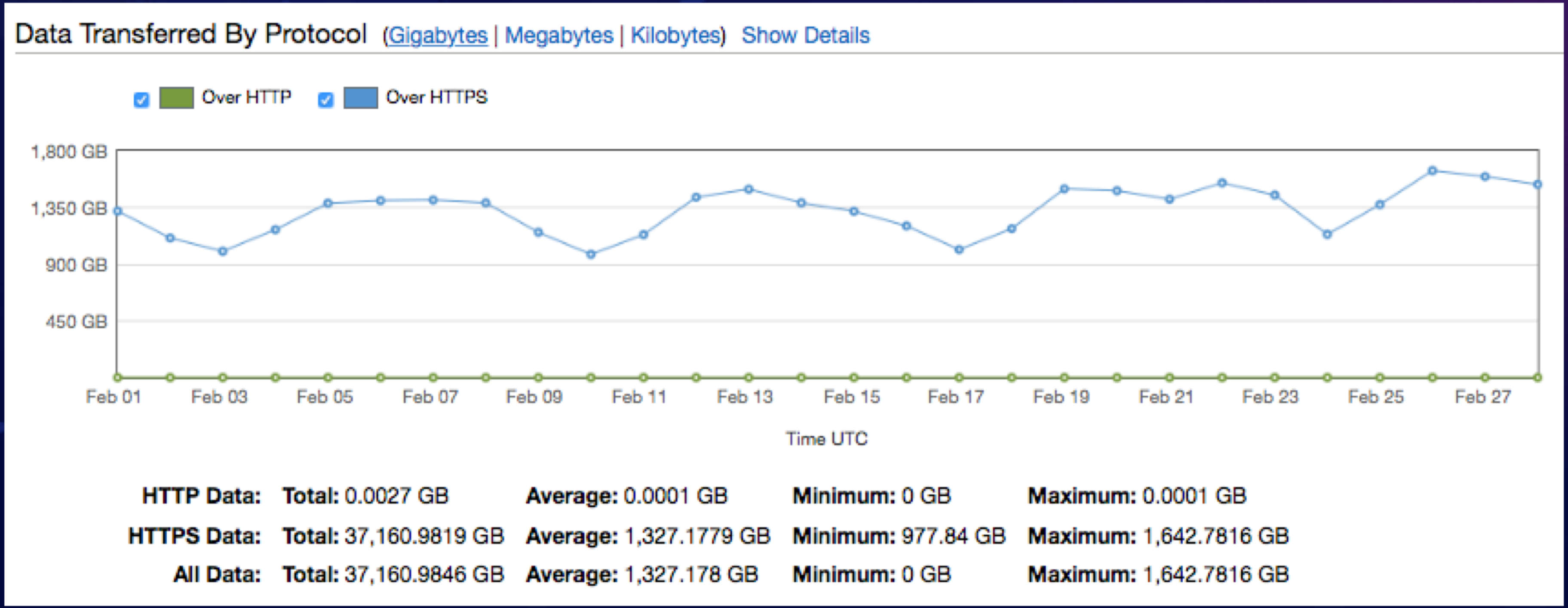
210 k

● production-identity-service



Number of Requests (Millions | Thousands | Not Scaled) Show Details

HTTP Requests:	**Total:** 7.494 K	**Average:** 0.2676 K	**Minimum:** 0.095 K	**Maximum:** 0.501 K
HTTPS Requests:	**Total:** 7,541.727 K	**Average:** 269.3474 K	**Minimum:** 188.006 K	**Maximum:** 373.999 K
All Requests:	**Total:** 7,549.221 K	**Average:** 269.615 K	**Minimum:** 0.095 K	**Maximum:** 373.999 K



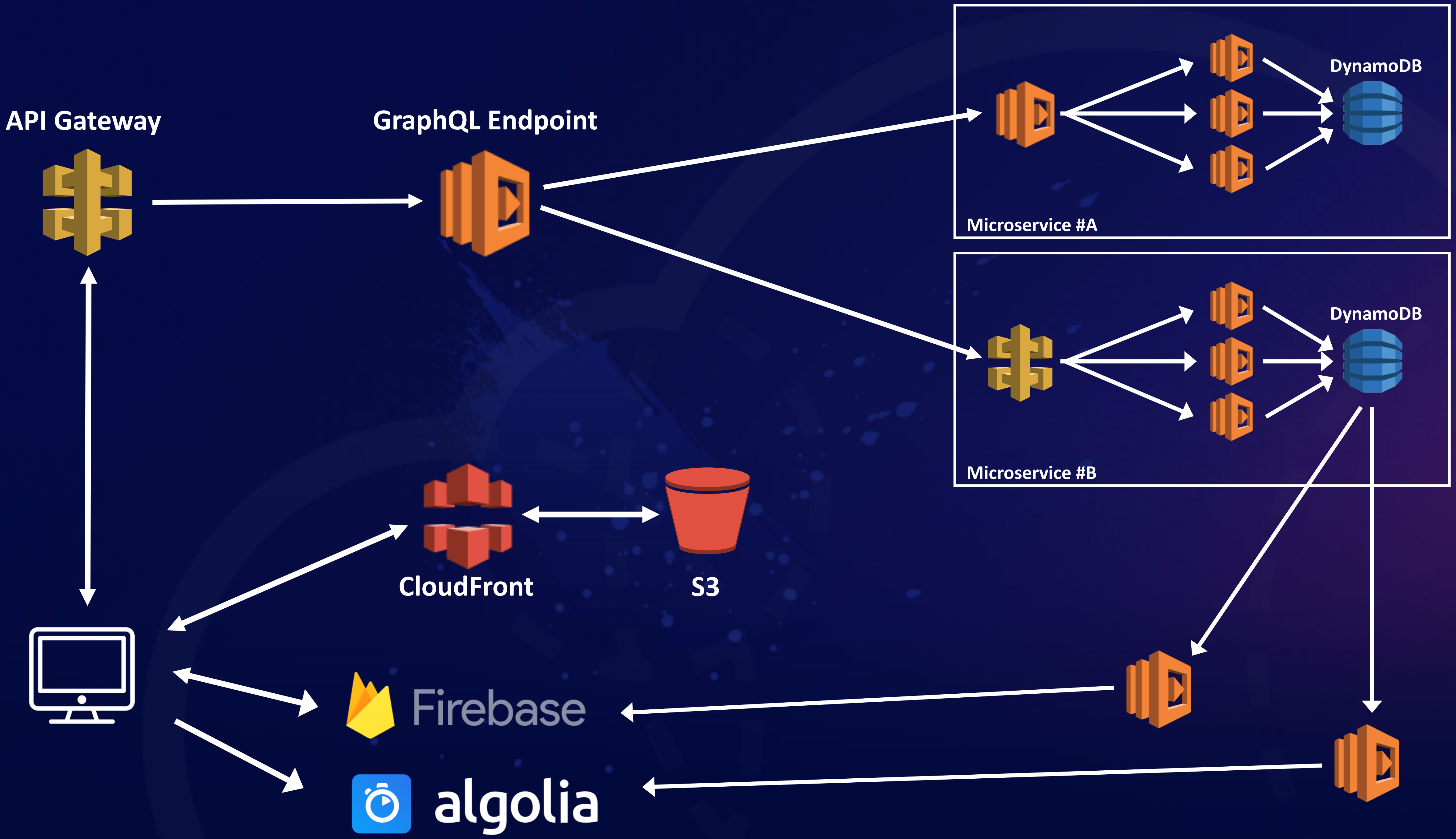
Data Transferred By Protocol (Gigabytes | Megabytes | Kilobytes) Show Details

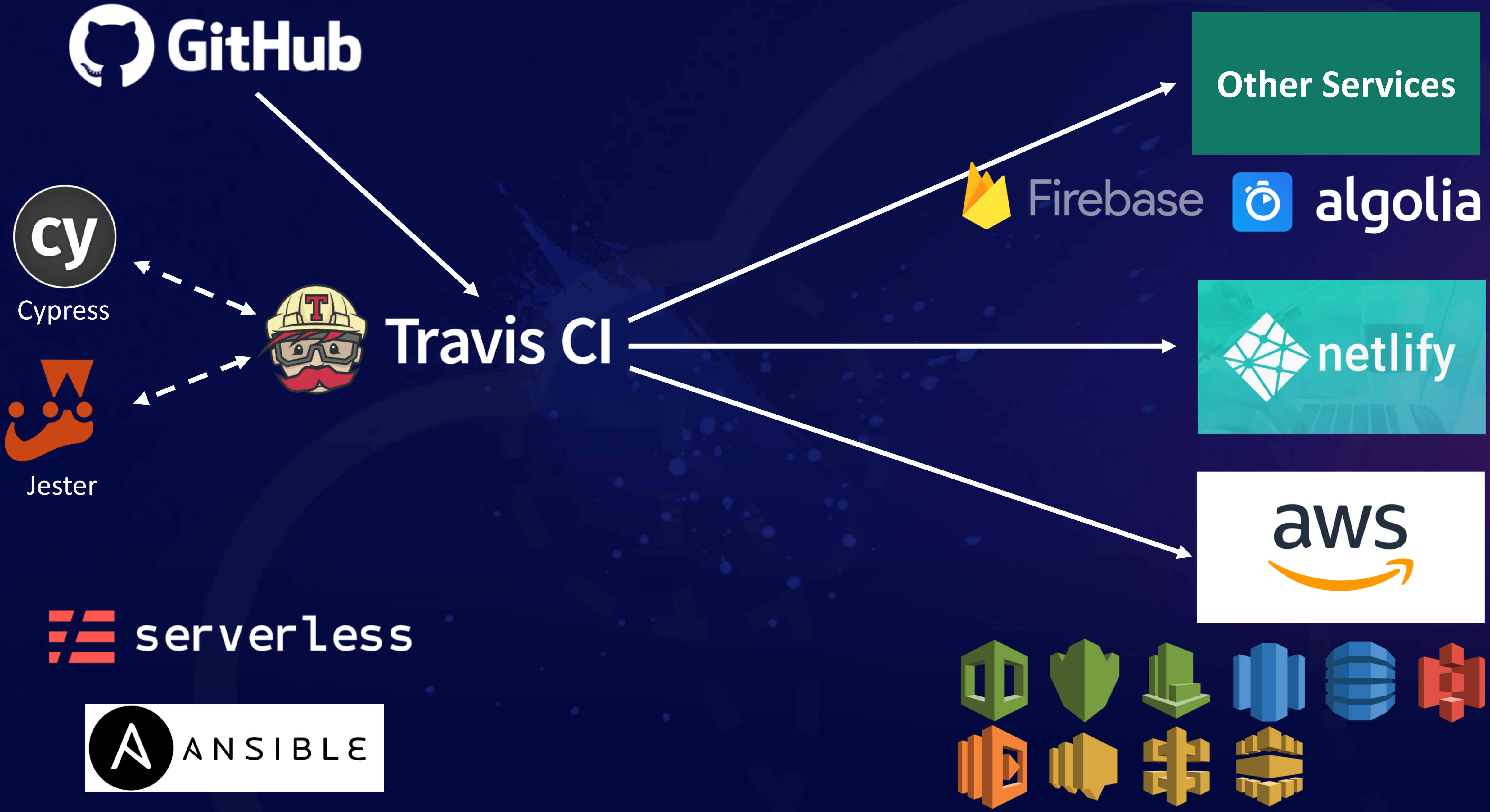
HTTP Data:	**Total:** 0.0027 GB	**Average:** 0.0001 GB	**Minimum:** 0 GB	**Maximum:** 0.0001 GB
HTTPS Data:	**Total:** 37,160.9819 GB	**Average:** 1,327.1779 GB	**Minimum:** 977.84 GB	**Maximum:** 1,642.7816 GB
All Data:	**Total:** 37,160.9846 GB	**Average:** 1,327.178 GB	**Minimum:** 0 GB	**Maximum:** 1,642.7816 GB

February 2018 – AWS Bill

Service	Cost
Key Management Service	$2.13
Kinesis	$20.16
Simple Storage Service	$58.36
API Gateway	**$100.00**
Elastic Transcoder	$169.89
Developer Support	$178.74
Lambda	**$206.66**
DynamoDB	$424.27
Redshift	$503.50
CloudWatch	$586.24
CloudFront	$3,775.42
Total	**$6,025.37**

Agility


API Gateway
GraphQL Endpoint
DynamoDB
Microservice #A
DynamoDB
Microservice #B
CloudFront
S3
Firebase
algolia



GitHub

Cypress

Jester

Travis CI

Other Services

Firebase

algolia

netlify

aws

serverless

ANSIBLE

Serverless Stories

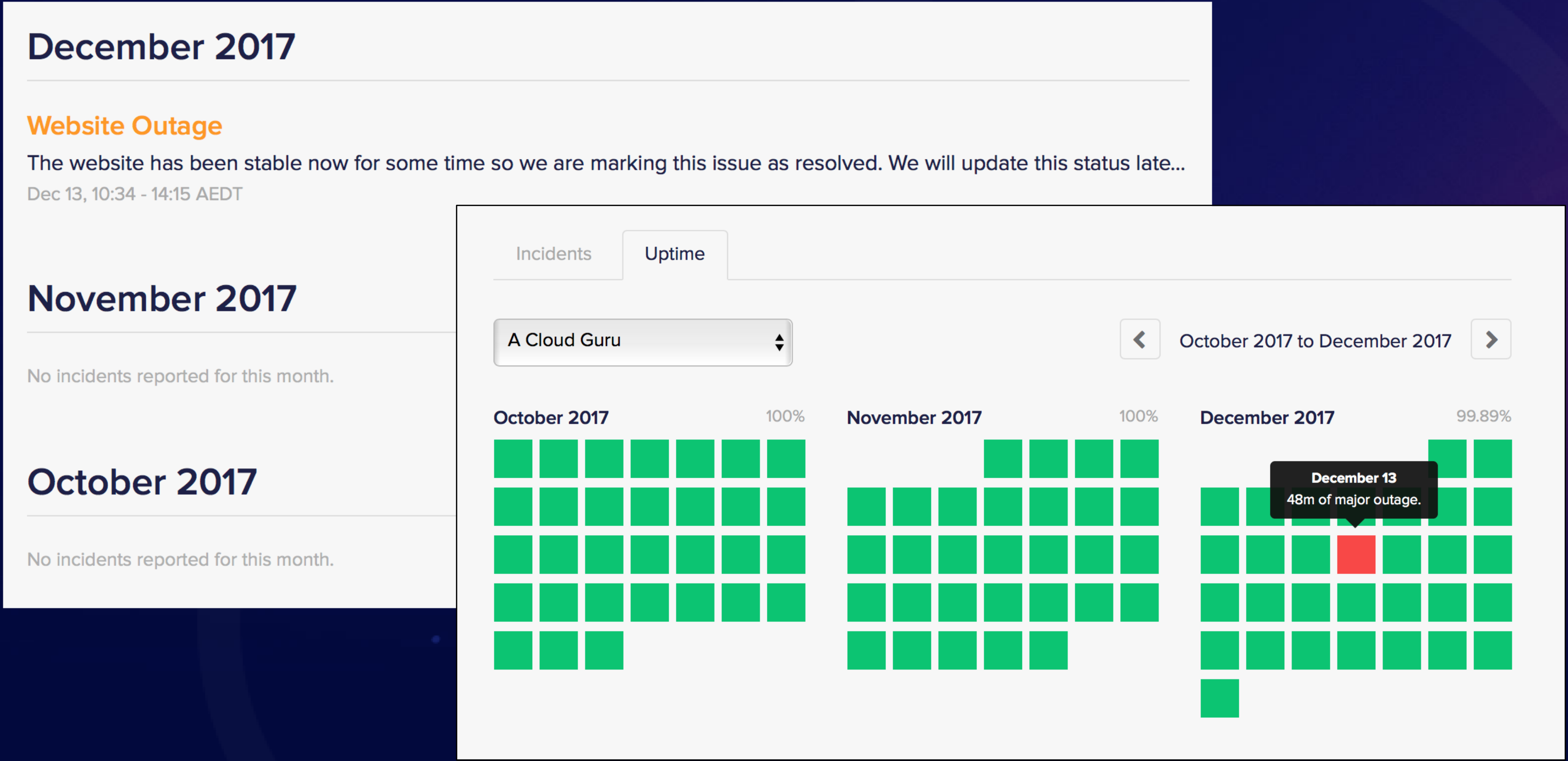
Frontend and API Failover

Making functions & services more resilient

Patterns and architectures

Frontend Failover

Frontend Failover



December 2017

Website Outage

The website has been stable now for some time so we are marking this issue as resolved. We will update this status late...

Dec 13, 10:34 - 14:15 AEDT

November 2017

No incidents reported for this month.

October 2017

No incidents reported for this month.

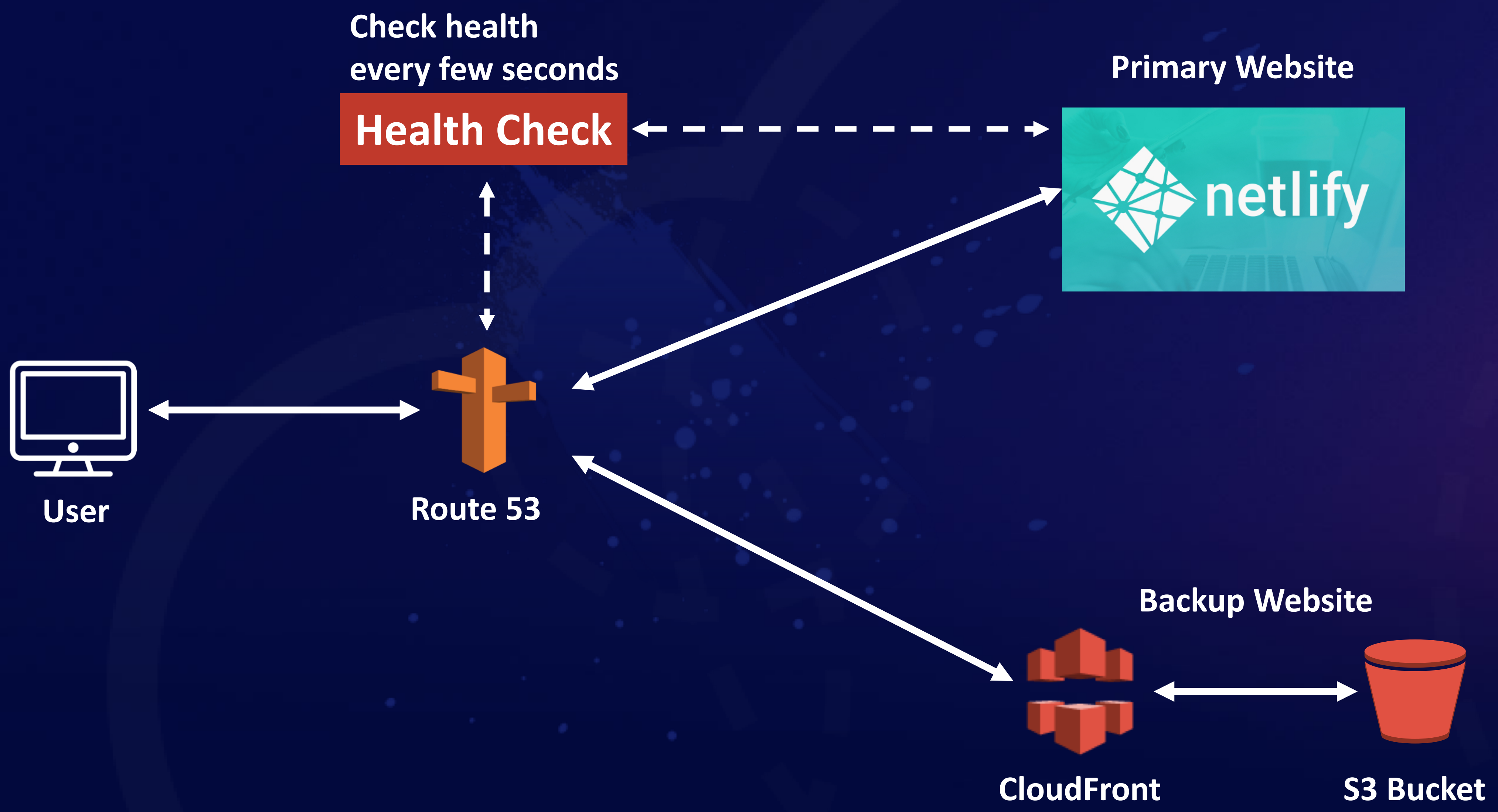
October 2017 100%

November 2017 100%

December 2017 99.89%

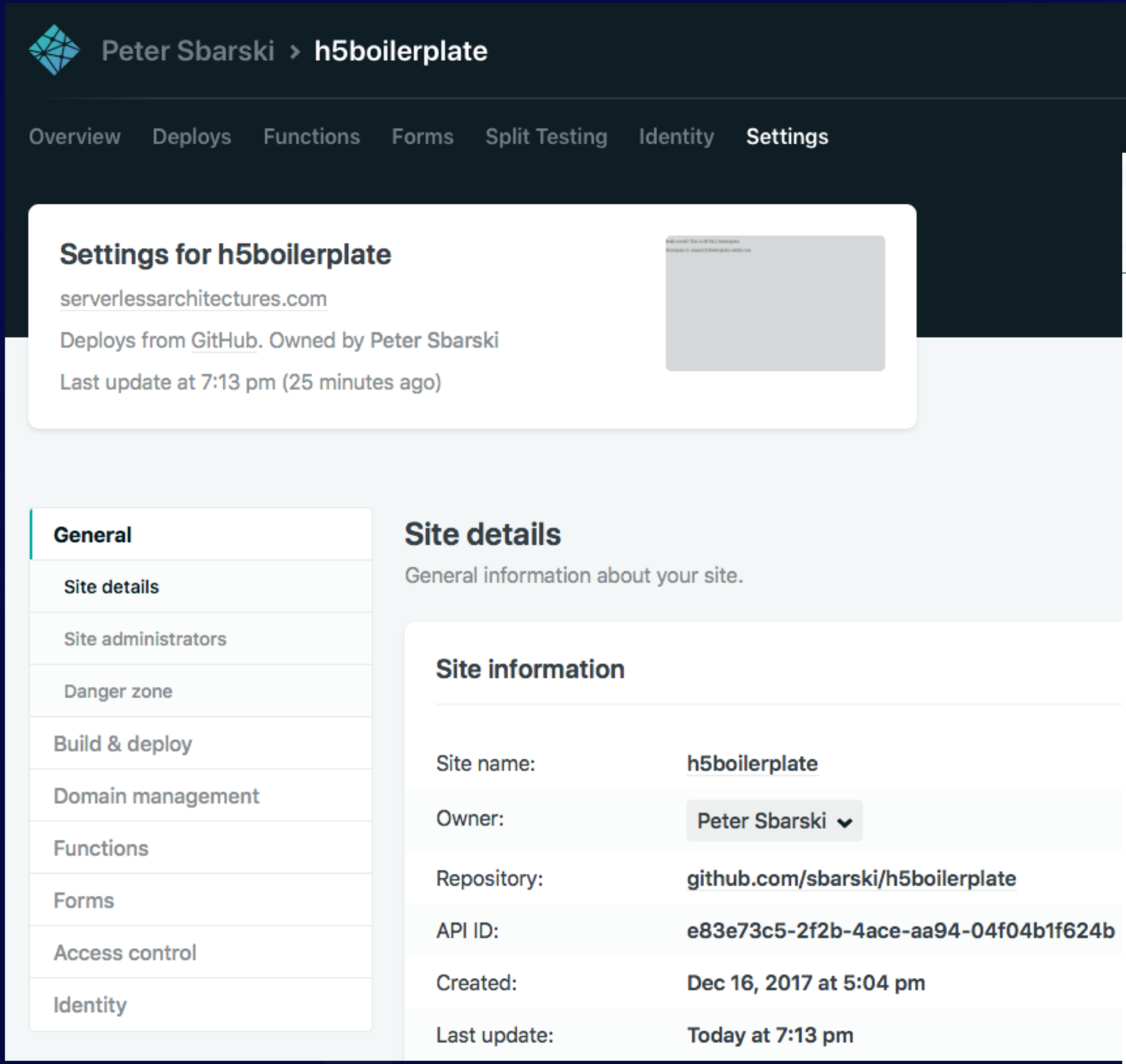
December 13
48m of major outage.

Frontend Failover



Check health every few seconds

Frontend Failover



Settings for h5boilerplate

serverlessarchitectures.com

Deploys from GitHub. Owned by Peter Sbarski

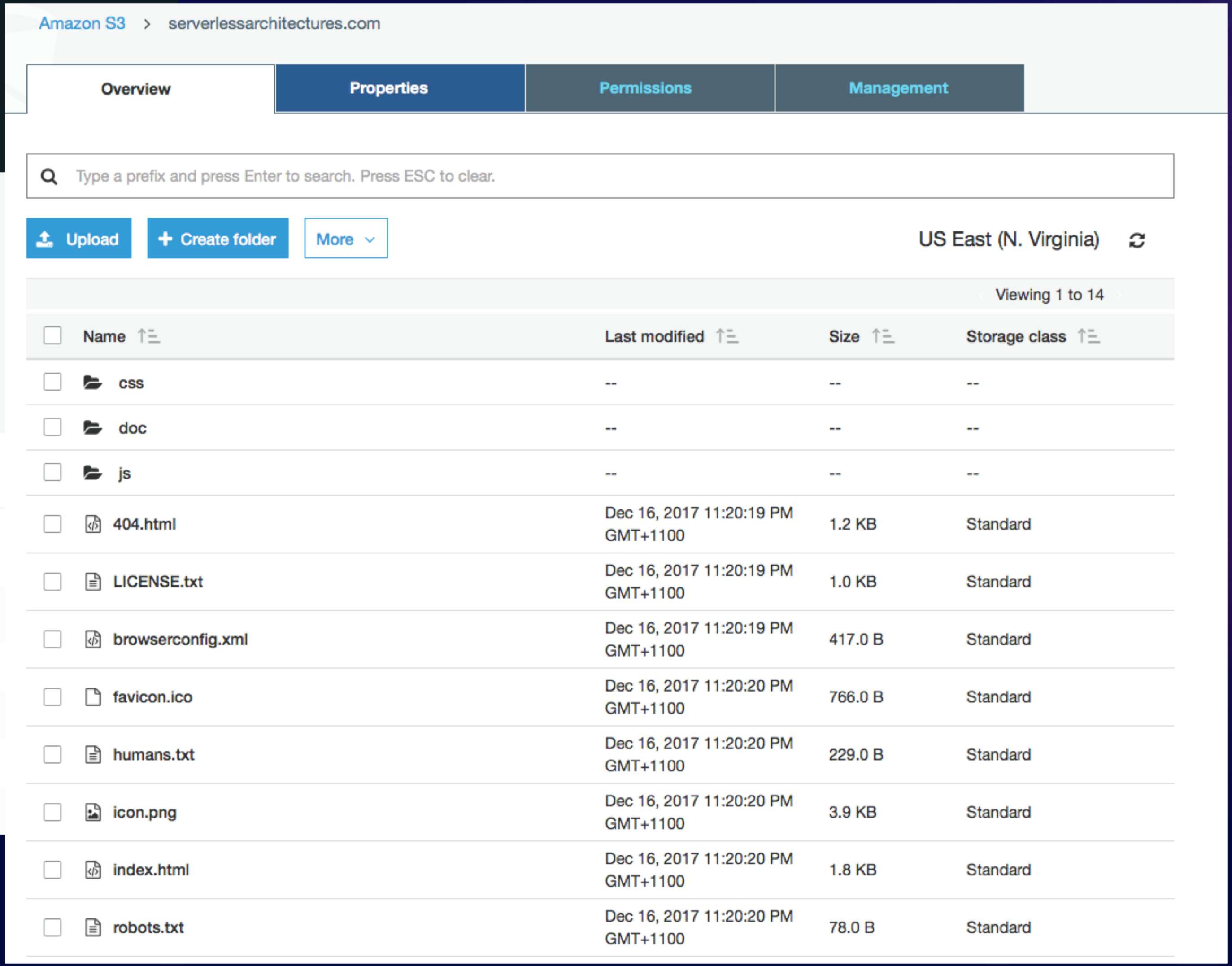
Last update at 7:13 pm (25 minutes ago)

Site details

General information about your site.

Site information

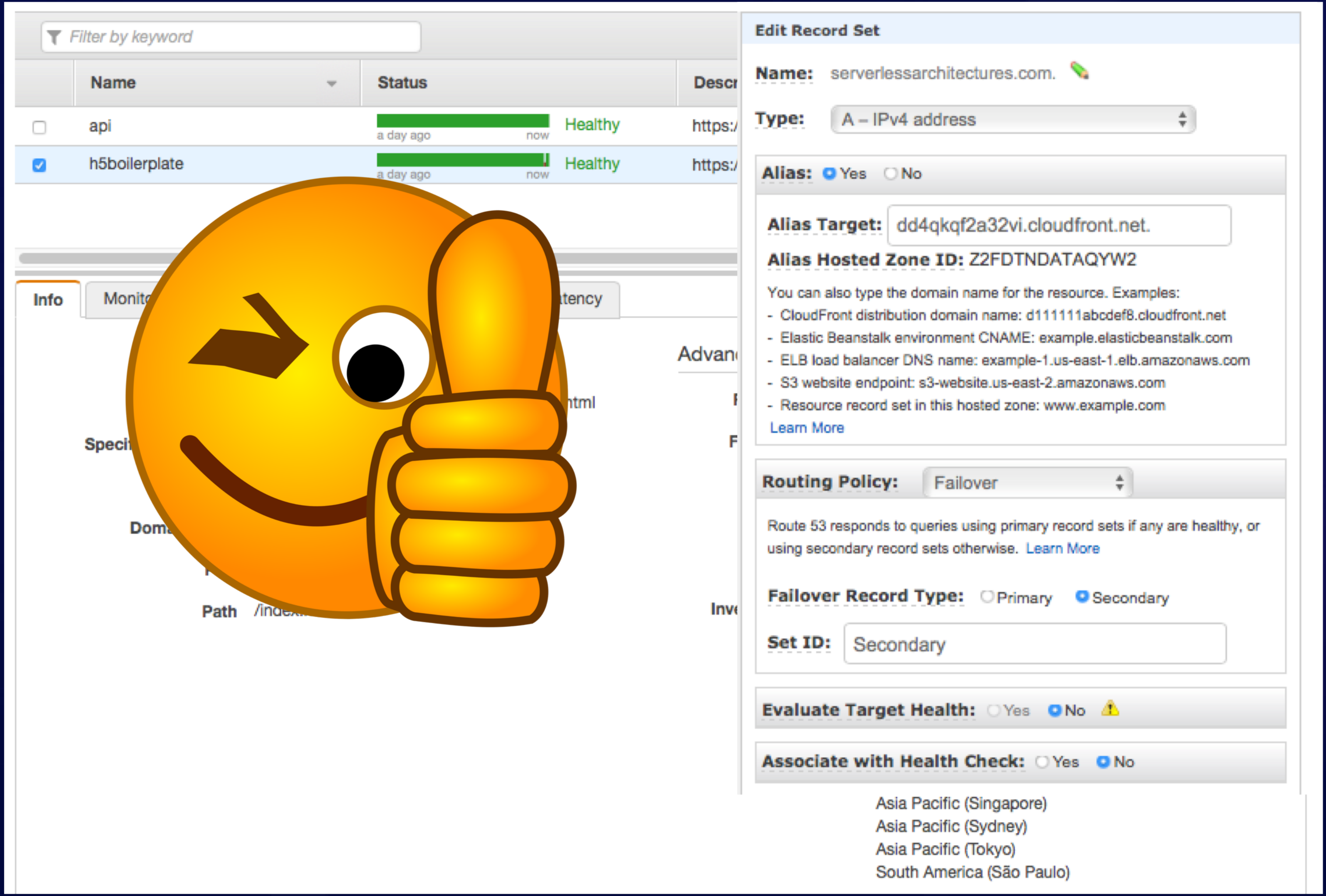
Site name:	h5boilerplate
Owner:	Peter Sbarski
Repository:	github.com/sbarski/h5boilerplate
API ID:	e83e73c5-2f2b-4ace-aa94-04f04b1f624b
Created:	Dec 16, 2017 at 5:04 pm
Last update:	Today at 7:13 pm



Amazon S3 > serverlessarchitectures.com
Overview
Properties
Permissions
Management
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
More
US East (N. Virginia)
Viewing 1 to 14

□	Name	Last modified	Size	Storage class
□	css	--	--	--
□	doc	--	--	--
□	js	--	--	--
□	404.html	Dec 16, 2017 11:20:19 PM GMT+1100	1.2 KB	Standard
□	LICENSE.txt	Dec 16, 2017 11:20:19 PM GMT+1100	1.0 KB	Standard
□	browserconfig.xml	Dec 16, 2017 11:20:19 PM GMT+1100	417.0 B	Standard
□	favicon.ico	Dec 16, 2017 11:20:20 PM GMT+1100	766.0 B	Standard
□	humans.txt	Dec 16, 2017 11:20:20 PM GMT+1100	229.0 B	Standard
□	icon.png	Dec 16, 2017 11:20:20 PM GMT+1100	3.9 KB	Standard
□	index.html	Dec 16, 2017 11:20:20 PM GMT+1100	1.8 KB	Standard
□	robots.txt	Dec 16, 2017 11:20:20 PM GMT+1100	78.0 B	Standard

Frontend Failover



Filter by keyword

	Name	Status		Descr
□	api	a day ago now	Healthy	https:/
☑	h5boilerplate	a day ago now	Healthy	https:/

Edit Record Set

Name: serverlessarchitectures.com.

Type: A – IPv4 address

Alias: ● Yes ○ No

Alias Target: dd4qkqf2a32vi.cloudfront.net.

Alias Hosted Zone ID: Z2FDTNDATAQYW2

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

Learn More

Routing Policy: Failover

Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. Learn More

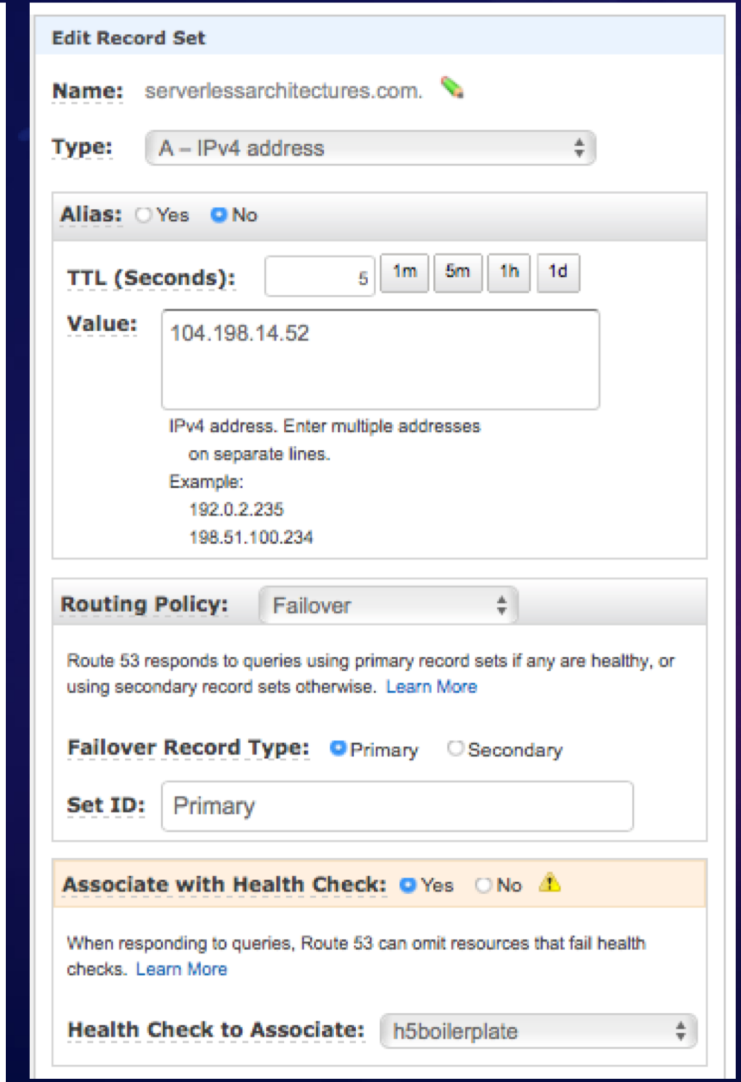
Failover Record Type: ○ Primary ● Secondary

Set ID: Secondary

Evaluate Target Health: ○ Yes ● No ⚠

Associate with Health Check: ○ Yes ● No

Asia Pacific (Singapore)
Asia Pacific (Sydney)
Asia Pacific (Tokyo)
South America (São Paulo)



Edit Record Set

Name: serverlessarchitectures.com.

Type: A – IPv4 address

Alias: ○ Yes ● No

TTL (Seconds): 5 1m 5m 1h 1d

Value: 104.198.14.52

IPv4 address. Enter multiple addresses on separate lines.
Example:
192.0.2.235
198.51.100.234

Routing Policy: Failover

Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. Learn More

Failover Record Type: ● Primary ○ Secondary

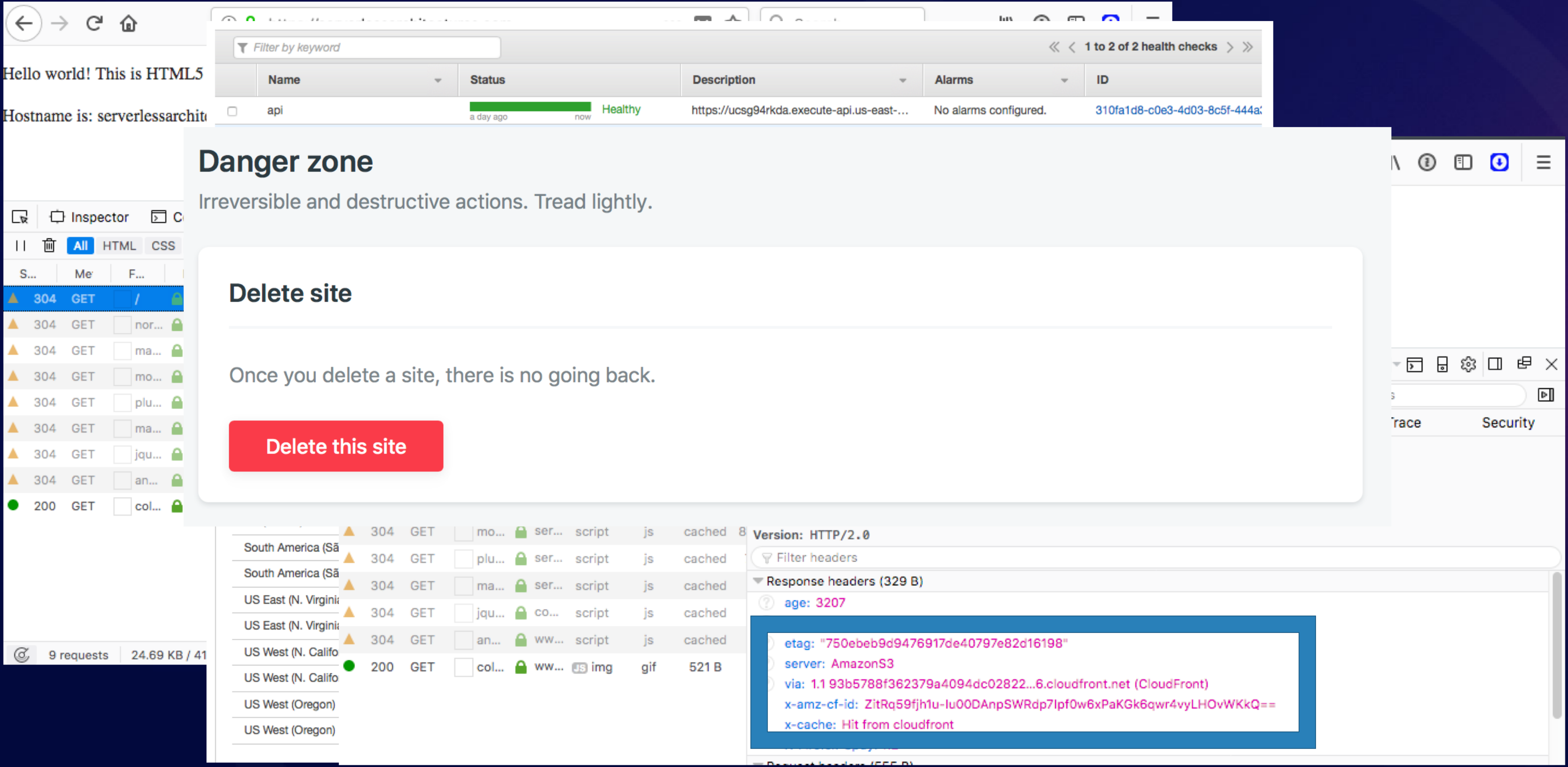
Set ID: Primary

Associate with Health Check: ● Yes ○ No ⚠

When responding to queries, Route 53 can omit resources that fail health checks. Learn More

Health Check to Associate: h5boilerplate

Frontend Failover



Hello world! This is HTML5

Hostname is: serverlessarchit

Filter by keyword

1 to 2 of 2 health checks

	Name	Status	Description	Alarms	ID
□	api	a day ago now Healthy	https://ucsg94rkda.execute-api.us-east-...	No alarms configured.	310fa1d8-c0e3-4d03-8c5f-444a

Danger zone

Irreversible and destructive actions. Tread lightly.

Delete site

Once you delete a site, there is no going back.

Delete this site

S...	Me	F...
304	GET	/
304	GET	nor...
304	GET	ma...
304	GET	mo...
304	GET	plu...
304	GET	ma...
304	GET	jqu...
304	GET	an...
200	GET	col...

9 requests 24.69 KB / 41

South America (Sã
South America (Sã
US East (N. Virgini
US East (N. Virgini
US West (N. Califo
US West (N. Califo
US West (Oregon)
US West (Oregon)

304	GET	mo...	ser...	script	js	cached	8
304	GET	plu...	ser...	script	js	cached	
304	GET	ma...	ser...	script	js	cached	
304	GET	jqu...	co...	script	js	cached	
304	GET	an...	ww...	script	js	cached	
200	GET	col...	ww...	img	gif	521 B	

Version: HTTP/2.0

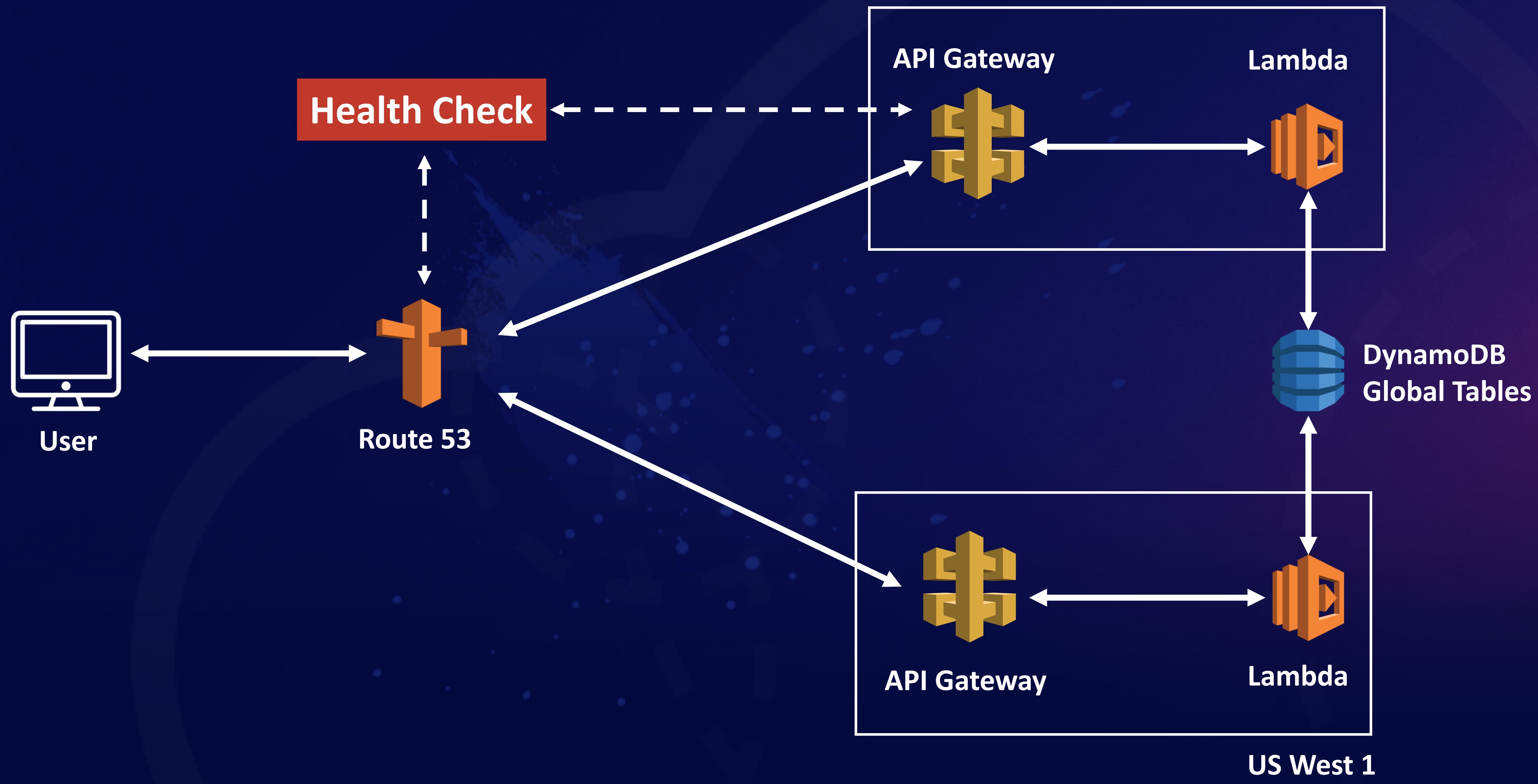
Filter headers

Response headers (329 B)

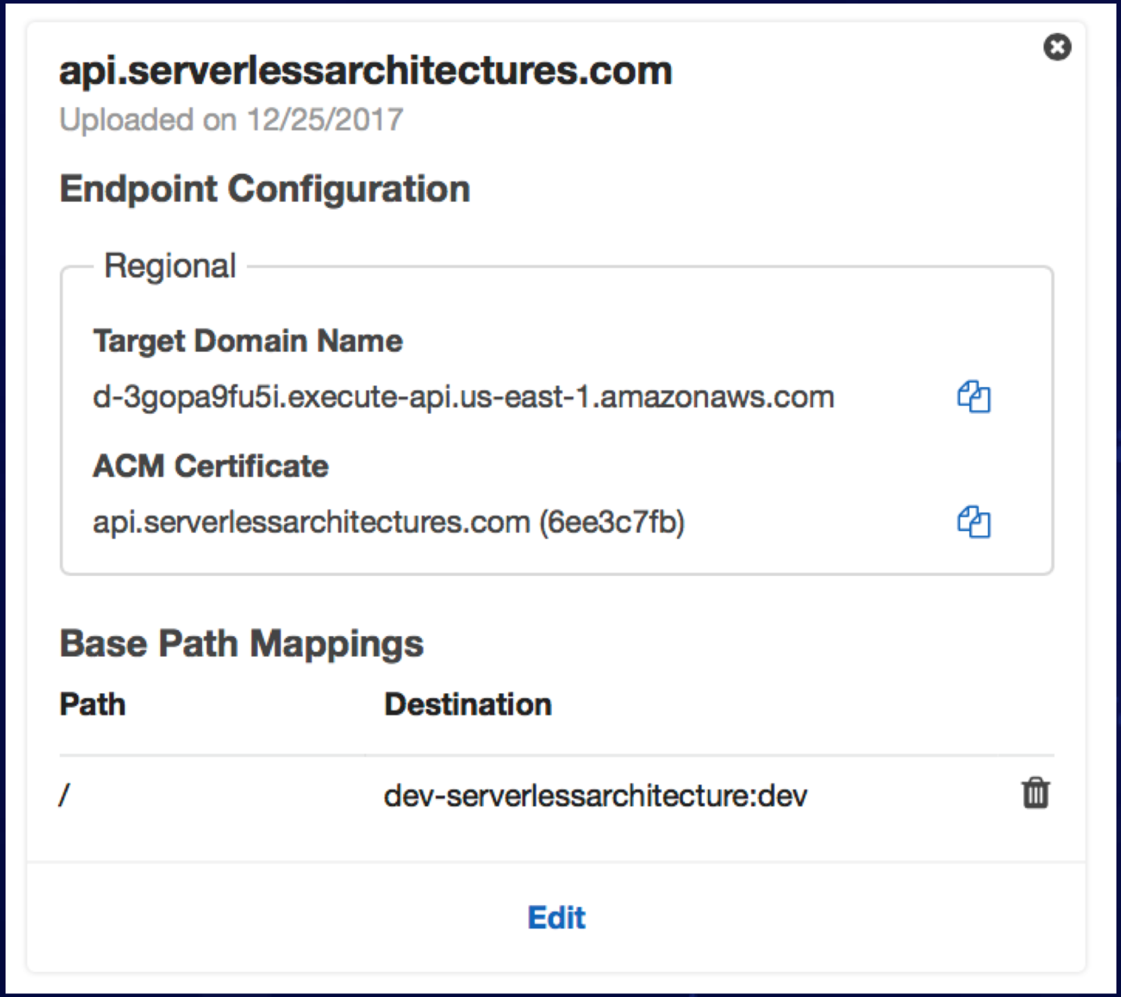
```
age: 3207
etag: "750ebeb9d9476917de40797e82d16198"
server: AmazonS3
via: 1.1 93b5788f362379a4094dc02822...6.cloudfront.net (CloudFront)
x-amz-cf-id: ZitRq59fjh1u-lu00DAnpSWRdp7lpf0w6xPaKGk6qwr4vyLHOvWKkQ==
x-cache: Hit from cloudfront
```

API Failover

API Failover



API Failover



api.serverlessarchitectures.com

Uploaded on 12/25/2017

Endpoint Configuration

Regional

Target Domain Name

d-3gopa9fu5i.execute-api.us-east-1.amazonaws.com

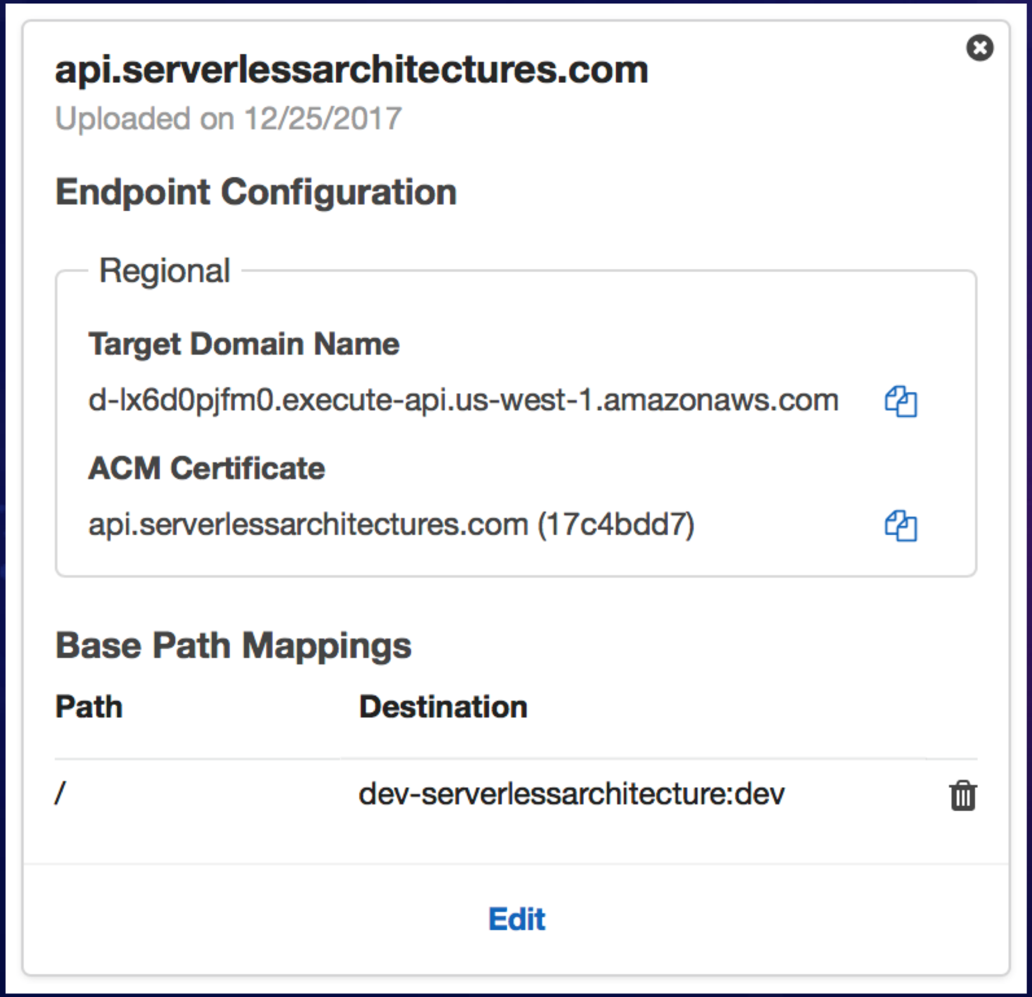
ACM Certificate

api.serverlessarchitectures.com (6ee3c7fb)

Base Path Mappings

Path	Destination
/	dev-serverlessarchitecture:dev

API Gateway – US East 1



api.serverlessarchitectures.com

Uploaded on 12/25/2017

Endpoint Configuration

Regional

Target Domain Name

d-lx6d0pjfm0.execute-api.us-west-1.amazonaws.com

ACM Certificate

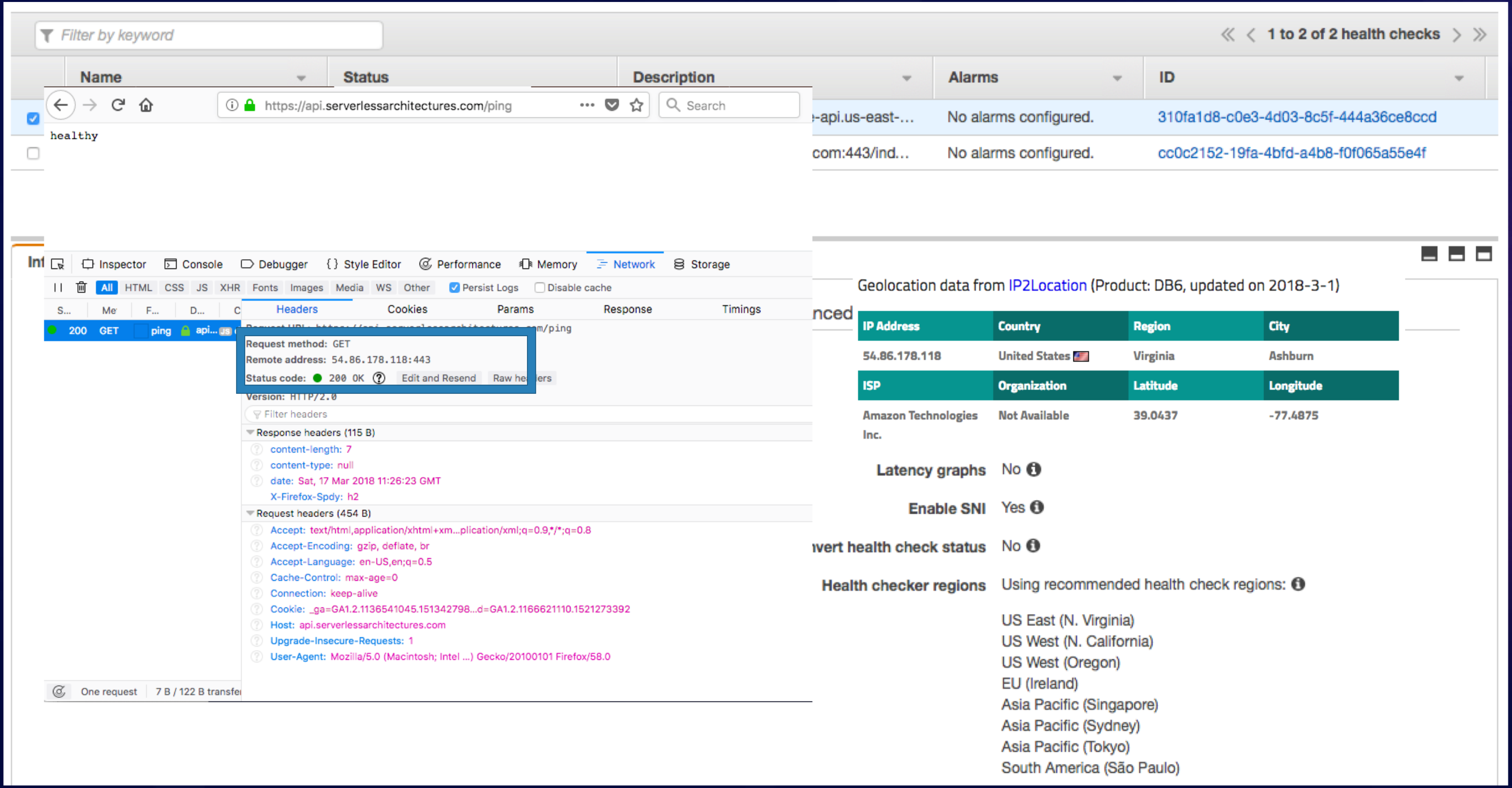
api.serverlessarchitectures.com (17c4bdd7)

Base Path Mappings

Path	Destination
/	dev-serverlessarchitecture:dev

API Gateway – US West 1

API Failover



	Name	Status	Description	Alarms	ID
☑			-api.us-east-...	No alarms configured.	310fa1d8-c0e3-4d03-8c5f-444a36ce8ccd
□			com:443/ind...	No alarms configured.	cc0c2152-19fa-4bfd-a4b8-f0f065a55e4f

https://api.serverlessarchitectures.com/ping

```
healthy
```

S...	Me	F...	D...
200	GET	ping	api...

```
Request method: GET
Remote address: 54.86.178.118:443
Status code: ● 200 OK
Version: HTTP/2.0

Response headers (115 B)
content-length: 7
content-type: null
date: Sat, 17 Mar 2018 11:26:23 GMT
X-Firefox-Spdy: h2

Request headers (454 B)
Accept: text/html,application/xhtml+xm...plication/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.5
Cache-Control: max-age=0
Connection: keep-alive
Cookie: _ga=GA1.2.1136541045.151342798...d=GA1.2.1166621110.1521273392
Host: api.serverlessarchitectures.com
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel ...) Gecko/20100101 Firefox/58.0
```

One request 7 B / 122 B transfer

Geolocation data from IP2Location (Product: DB6, updated on 2018-3-1)

IP Address	Country	Region	City
54.86.178.118	United States	Virginia	Ashburn

ISP	Organization	Latitude	Longitude
Amazon Technologies Inc.	Not Available	39.0437	-77.4875

Latency graphs: No

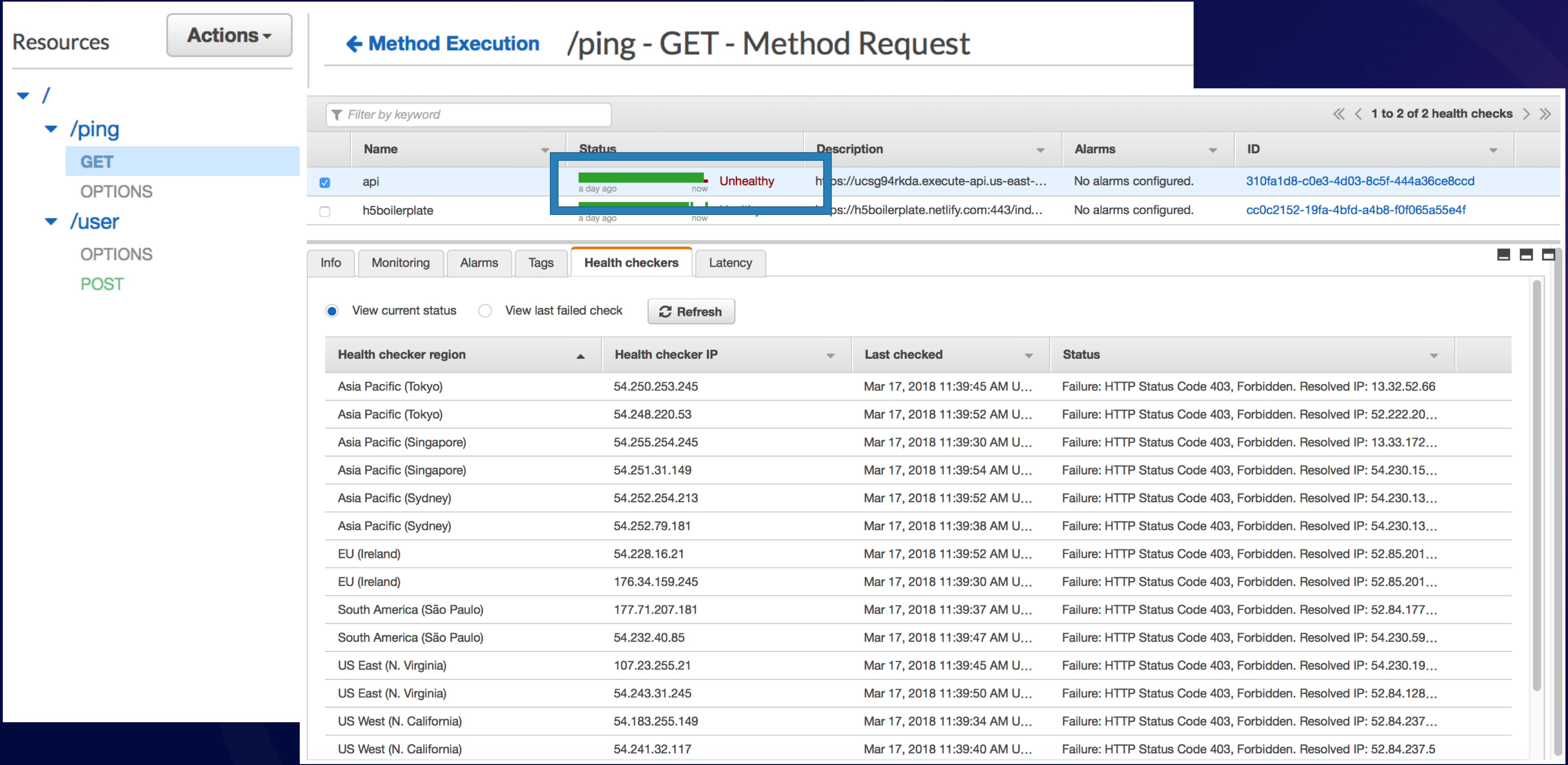
Enable SNI: Yes

vert health check status: No

Health checker regions: Using recommended health check regions:

US East (N. Virginia)
US West (N. California)
US West (Oregon)
EU (Ireland)
Asia Pacific (Singapore)
Asia Pacific (Sydney)
Asia Pacific (Tokyo)
South America (São Paulo)

API Failover



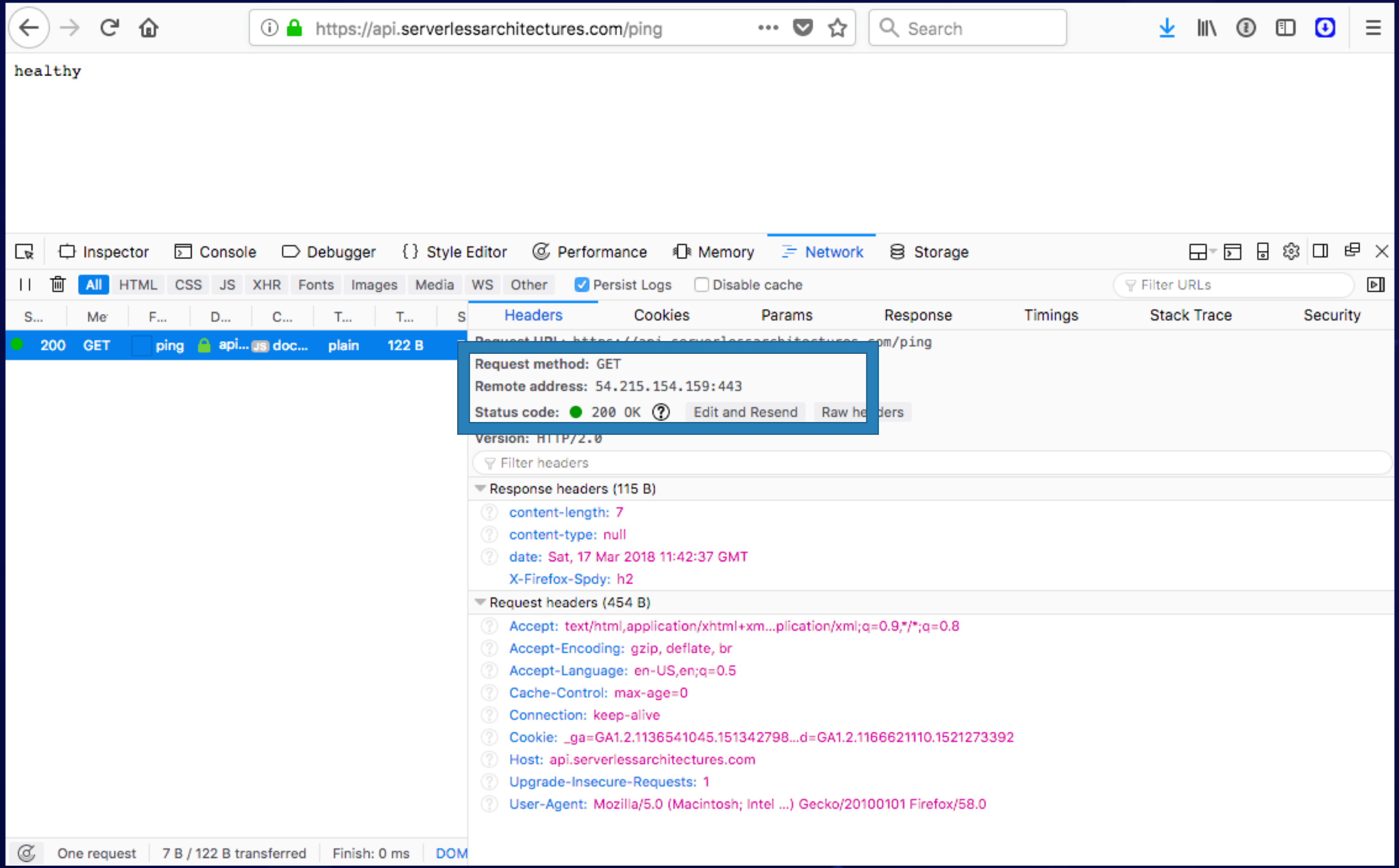
/ping - GET - Method Request

	Name	Status	Description	Alarms	ID
☑	api	a day ago now Unhealthy	h ps://ucsg94rkda.execute-api.us-east-…	No alarms configured.	310fa1d8-c0e3-4d03-8c5f-444a36ce8ccd
□	h5boilerplate	a day ago now	ps://h5boilerplate.netlify.com:443/ind…	No alarms configured.	cc0c2152-19fa-4bfd-a4b8-f0f065a55e4f

● View current status ○ View last failed check

Health checker region	Health checker IP	Last checked	Status
Asia Pacific (Tokyo)	54.250.253.245	Mar 17, 2018 11:39:45 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 13.32.52.66
Asia Pacific (Tokyo)	54.248.220.53	Mar 17, 2018 11:39:52 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 52.222.20…
Asia Pacific (Singapore)	54.255.254.245	Mar 17, 2018 11:39:30 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 13.33.172…
Asia Pacific (Singapore)	54.251.31.149	Mar 17, 2018 11:39:54 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 54.230.15…
Asia Pacific (Sydney)	54.252.254.213	Mar 17, 2018 11:39:52 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 54.230.13…
Asia Pacific (Sydney)	54.252.79.181	Mar 17, 2018 11:39:38 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 54.230.13…
EU (Ireland)	54.228.16.21	Mar 17, 2018 11:39:52 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 52.85.201…
EU (Ireland)	176.34.159.245	Mar 17, 2018 11:39:30 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 52.85.201…
South America (São Paulo)	177.71.207.181	Mar 17, 2018 11:39:37 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 52.84.177…
South America (São Paulo)	54.232.40.85	Mar 17, 2018 11:39:47 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 54.230.59…
US East (N. Virginia)	107.23.255.21	Mar 17, 2018 11:39:45 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 54.230.19…
US East (N. Virginia)	54.243.31.245	Mar 17, 2018 11:39:50 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 52.84.128…
US West (N. California)	54.183.255.149	Mar 17, 2018 11:39:34 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 52.84.237…
US West (N. California)	54.241.32.117	Mar 17, 2018 11:39:40 AM U…	Failure: HTTP Status Code 403, Forbidden. Resolved IP: 52.84.237.5

API Failover



https://api.serverlessarchitectures.com/ping

healthy

```
● 200  GET  ping  api...  doc...  plain  122 B
```

Request URL: https://api.serverlessarchitectures.com/ping

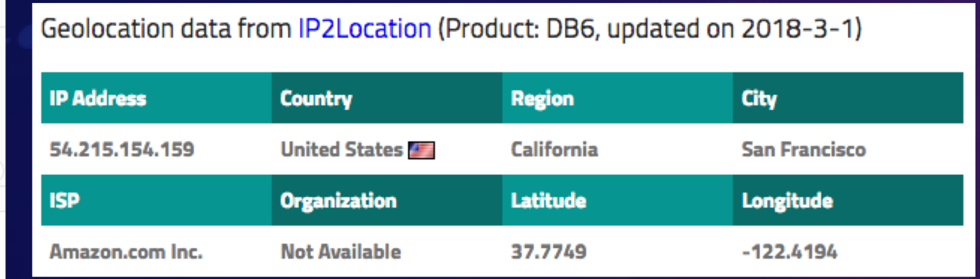
Request method: GET
Remote address: 54.215.154.159:443
Status code: ● 200 OK
Version: HTTP/2.0

Response headers (115 B)

- content-length: 7
- content-type: null
- date: Sat, 17 Mar 2018 11:42:37 GMT
- X-Firefox-Spdy: h2

Request headers (454 B)

- Accept: text/html,application/xhtml+xm...plication/xml;q=0.9,*/*;q=0.8
- Accept-Encoding: gzip, deflate, br
- Accept-Language: en-US,en;q=0.5
- Cache-Control: max-age=0
- Connection: keep-alive
- Cookie: _ga=GA1.2.1136541045.151342798...d=GA1.2.1166621110.1521273392
- Host: api.serverlessarchitectures.com
- Upgrade-Insecure-Requests: 1
- User-Agent: Mozilla/5.0 (Macintosh; Intel ...) Gecko/20100101 Firefox/58.0

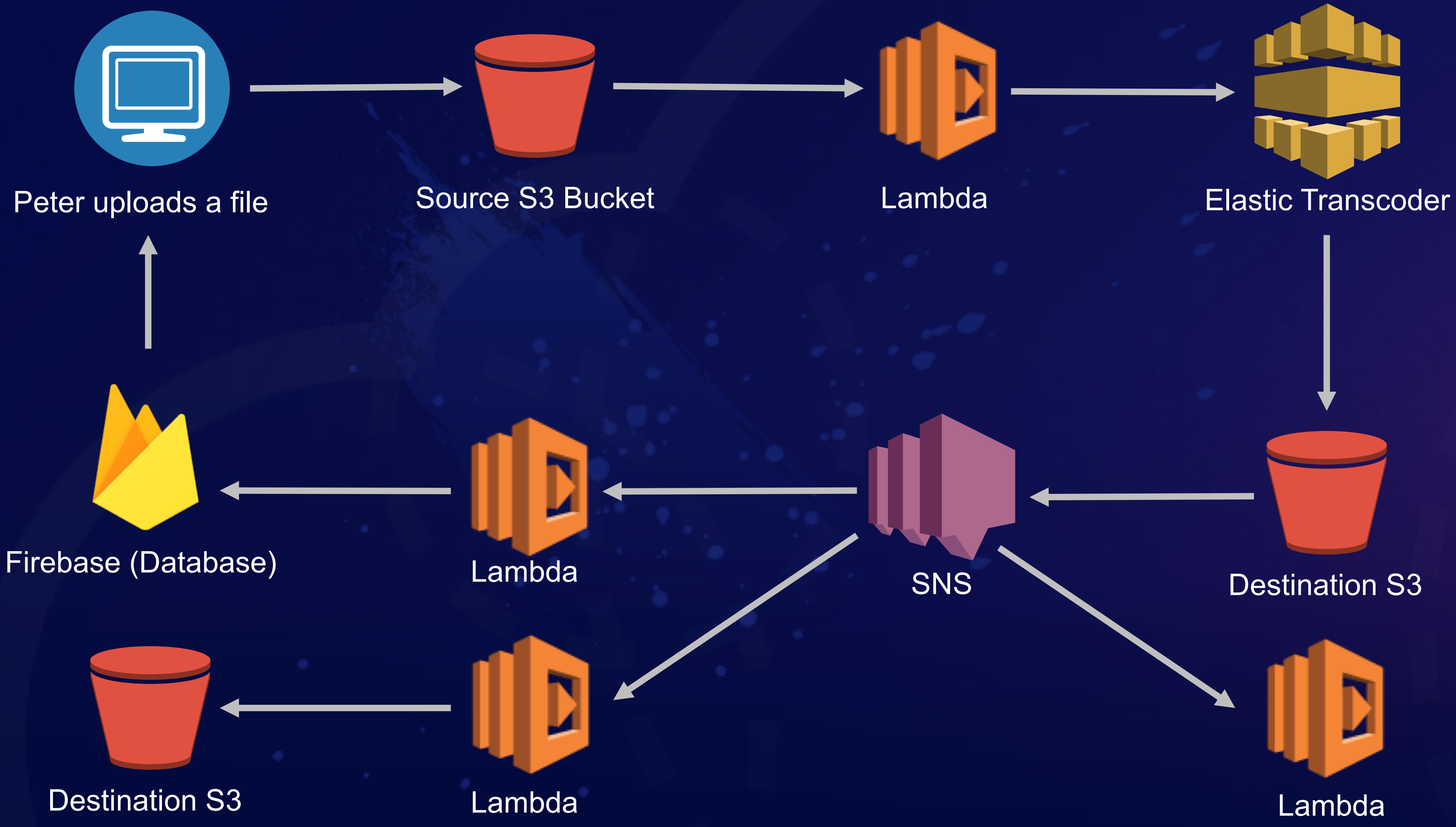


Geolocation data from IP2Location (Product: DB6, updated on 2018-3-1)

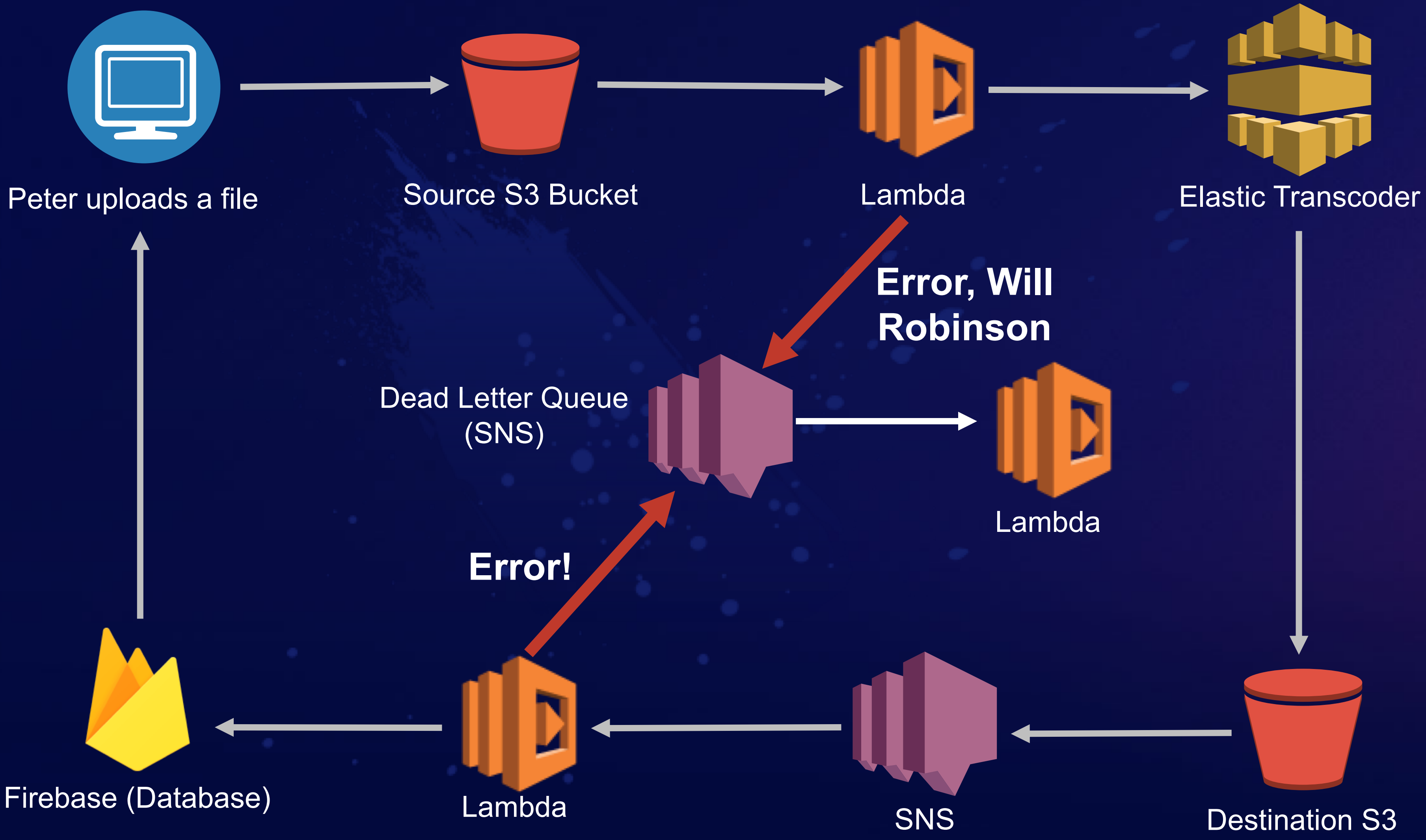
IP Address	Country	Region	City
54.215.154.159	United States	California	San Francisco
ISP	**Organization**	**Latitude**	**Longitude**
Amazon.com Inc.	Not Available	37.7749	-122.4194

Making functions & services more resilient

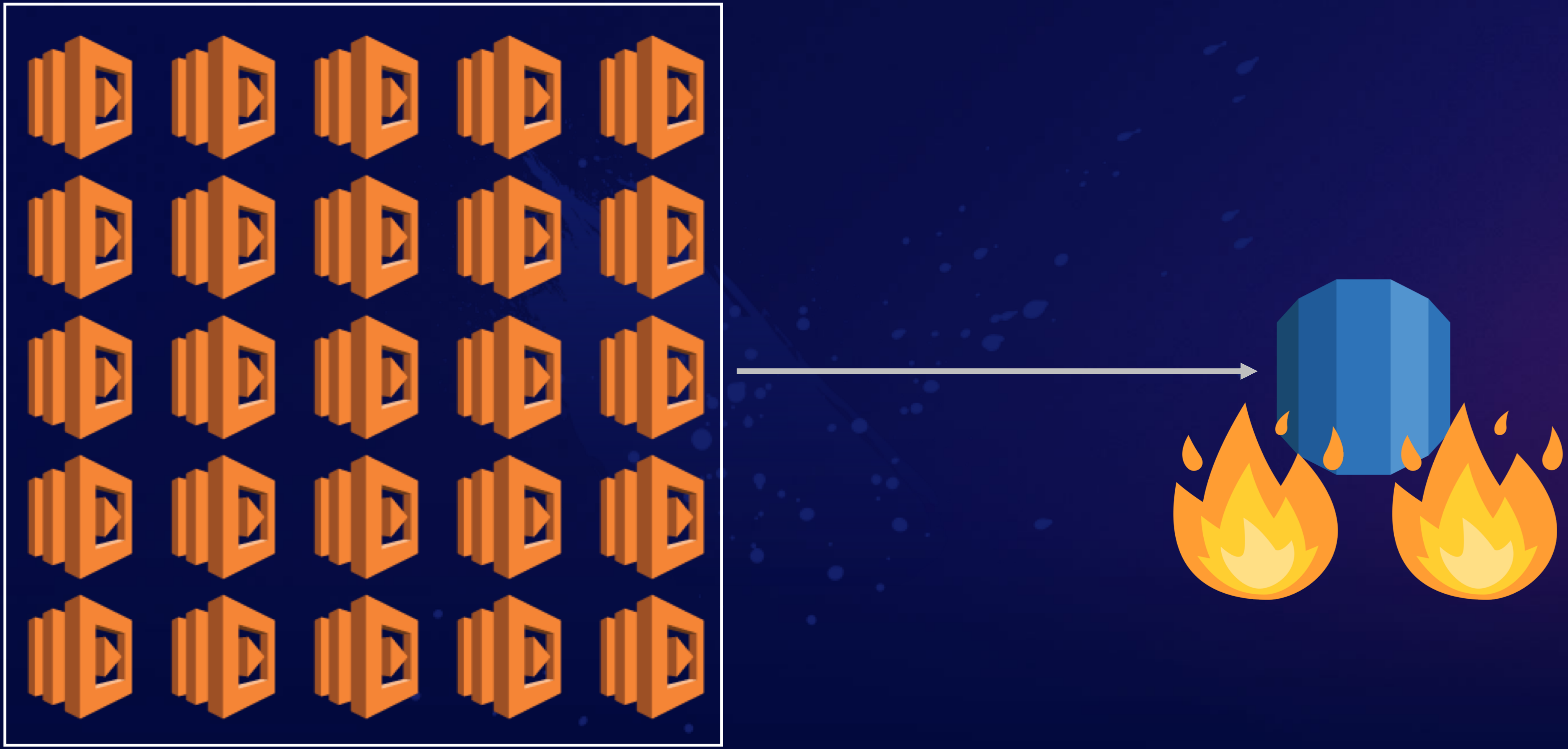
Handling Errors



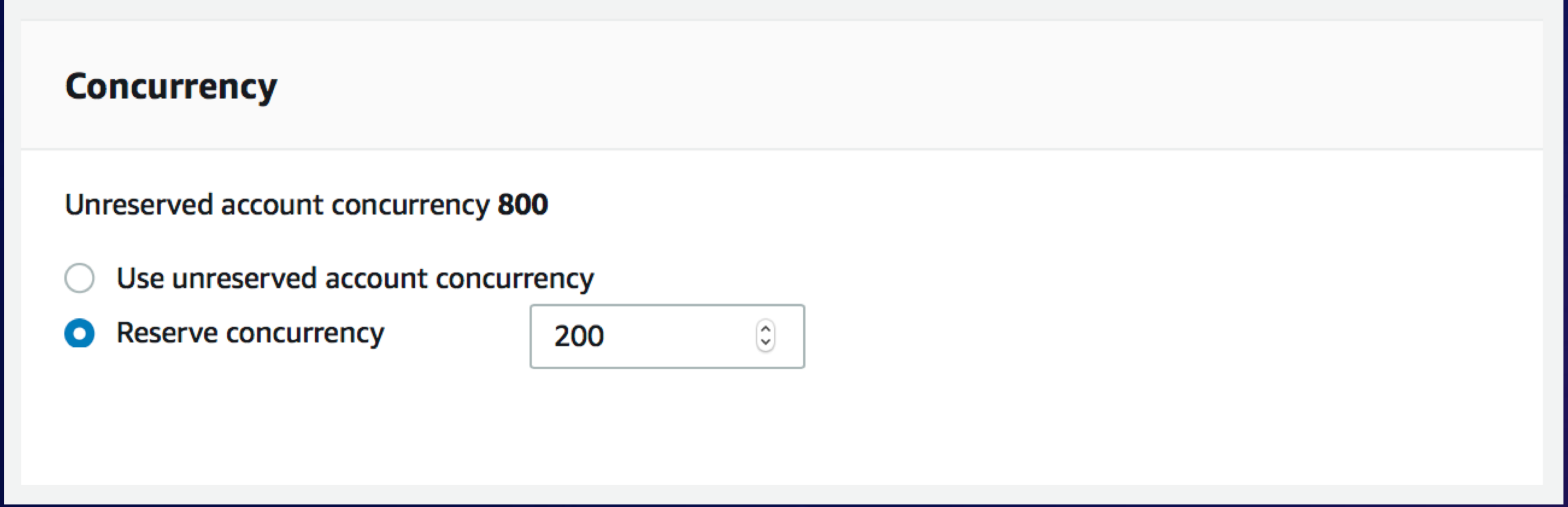
Handling Errors



Burning down the house



Burning down the house



Concurrency

Unreserved account concurrency **800**

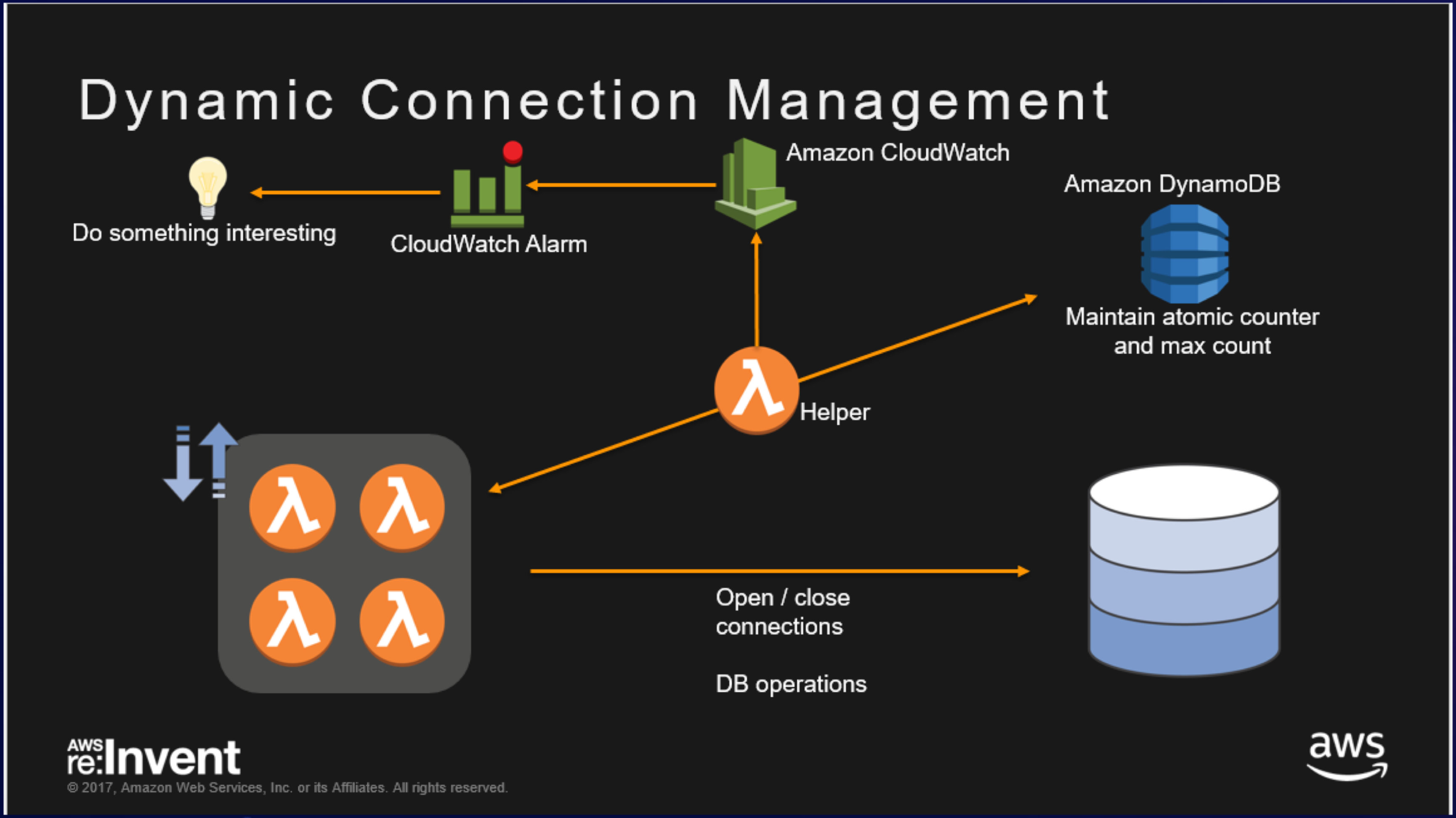
○ Use unreserved account concurrency

● Reserve concurrency 200

You might still over provision or under provision DB connections

Multiple functions may need DB access with different usage profiles at different times.

Burning down the house



Dynamic Connection Management

https://github.com/aws-samples/aws-lambda-manage-rds-connections

Tips

- 1 function = 1 task (avoid fat monoliths)
- No state (be idempotent)
- Design for failure
- More memory = more CPU and IO
- Set function concurrency to 0 as a kill switch
- Keep permissions and roles tight
- Incremental architecture is not dirty

Patterns and Architectures

Patterns

Primitive

- Periodical (Cron Jobs)

API

- Proxy
- Facade

Orchestration

- One way chain
- Two way chain
- Fan in
- Fan out

Workflows

- Long Running tasks
- Pipes and Filters
- Inline Stream Transform

Traditional

- Command
- Singleton

Compound

- Backends
- CQRS
- Data processing

Source: Serverless Design Patterns (T. Wagner, Y. Kiriaty, P. Sbarski)

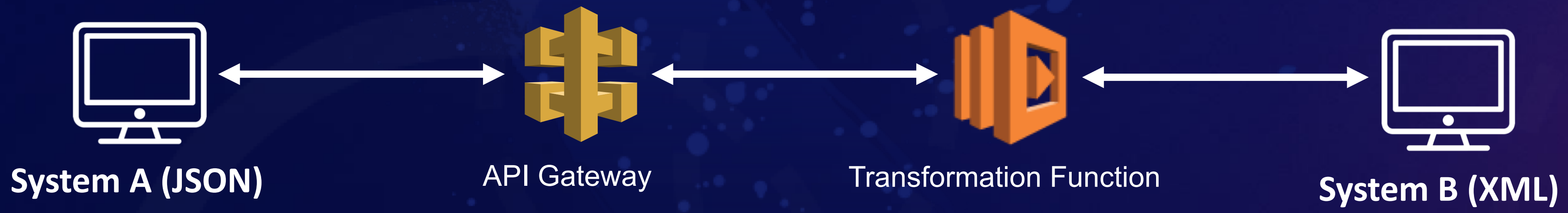
Patterns

Name
API Proxy (also known as wrapper)

Description
Acts as a mediator between two systems that cannot communicate directly. Transforms request and response payloads to facilitate exchange of information.

Motivation
Useful when incompatible systems need to talk. Reduces coupling by removing the need to build direct dependencies between incompatible systems.

API Proxy



Transformer function transforms JSON to XML and back again.

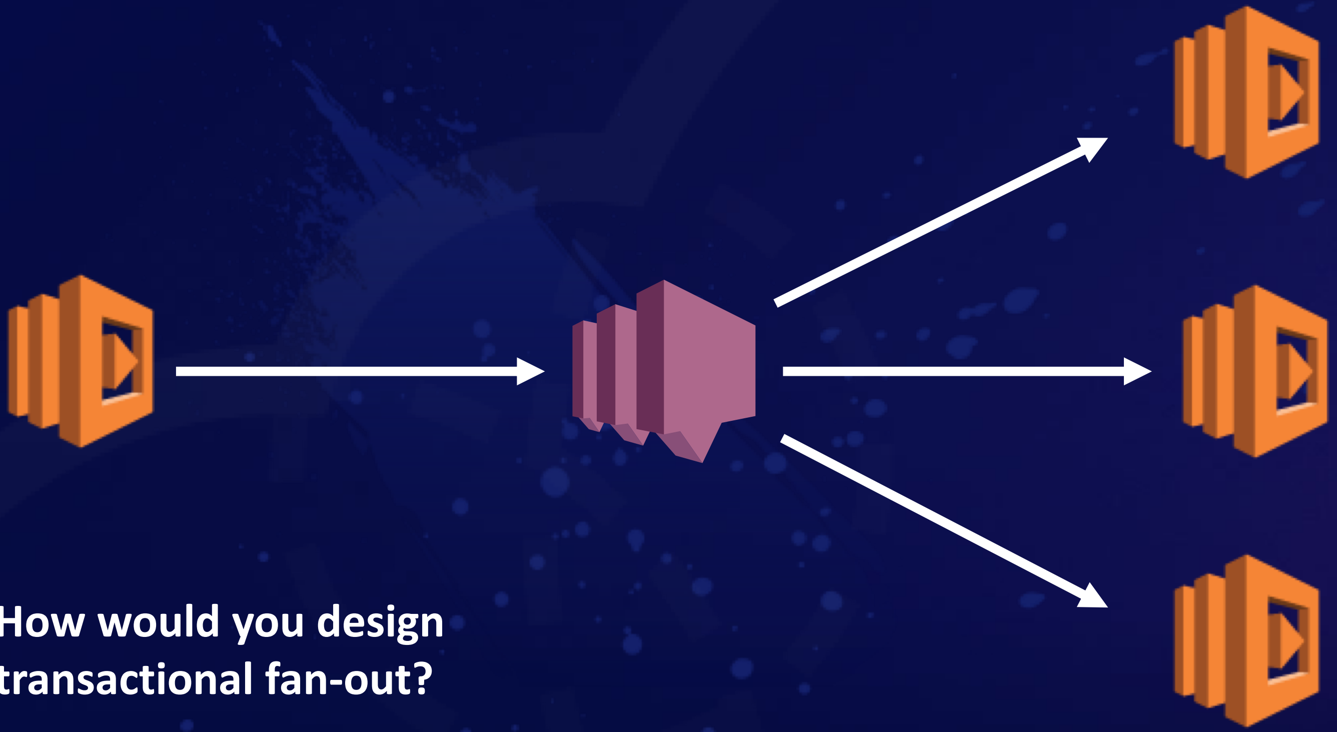
Patterns

Name
Simple fan-out

Description
Allows multiple endpoints to receive a copy of an input event. Turns any single-receiver delivery system into a multiple-receiver system.

Motivation
Event-based systems are often designed to have a single receiver for events, and API calls are by definition single receiver. The simple fan-out pattern asynchronously delivers its triggering event to one or more workers.

Simple fan-out



How would you design transactional fan-out?

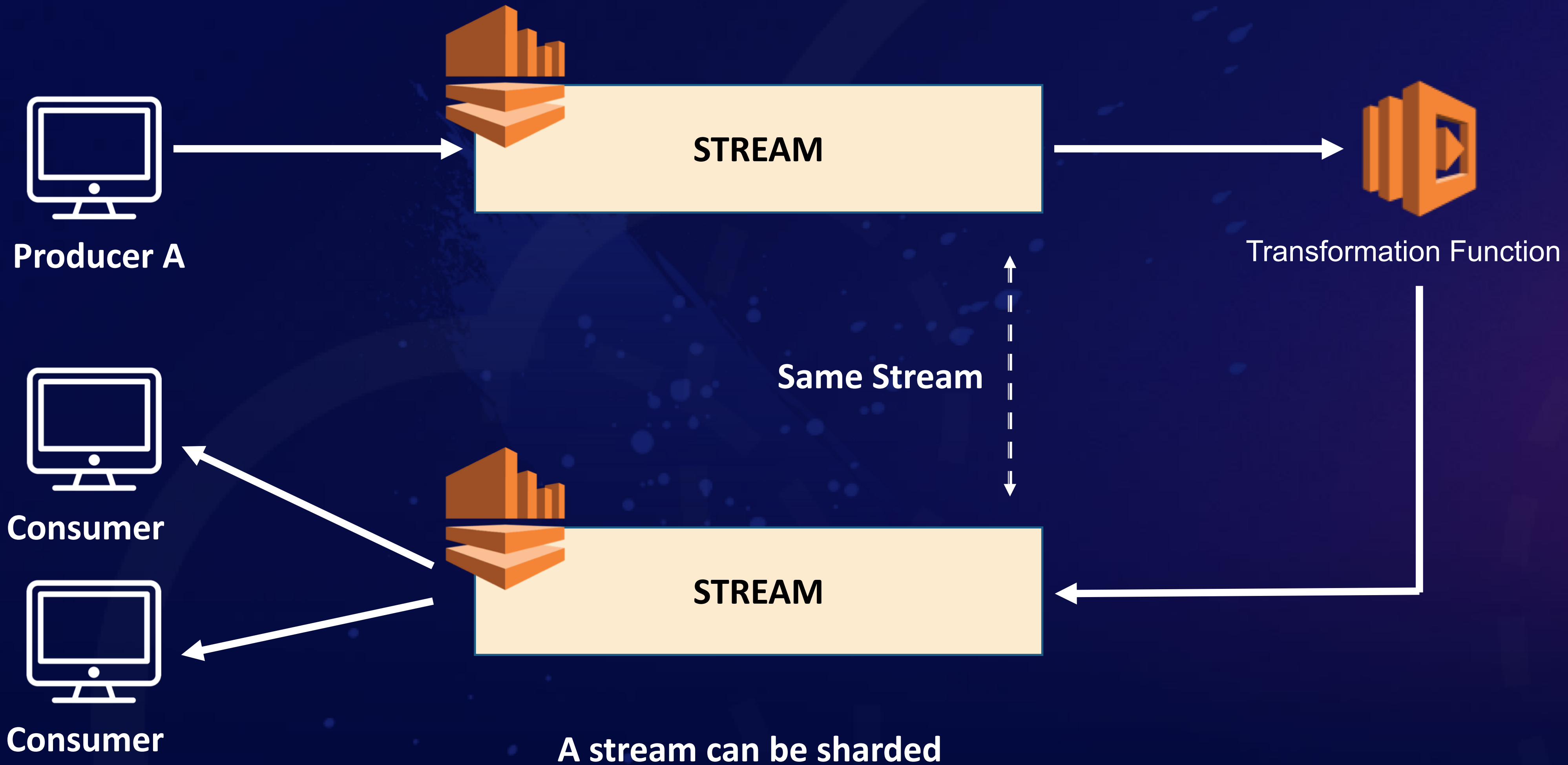
Patterns

Name
Inline stream transform

Description
Transmits data between systems. Can be chained, can multiple and demultiplex at the source or destination. A transformation function can transform a record with the result progressing through the stream.

Motivation
A way to decouple systems that share data. Can offer temporal decoupling by allowing producers and consumers to operate at different rates. A transform function is used to clean, modify, group, analyze data before it gets to the consumer.

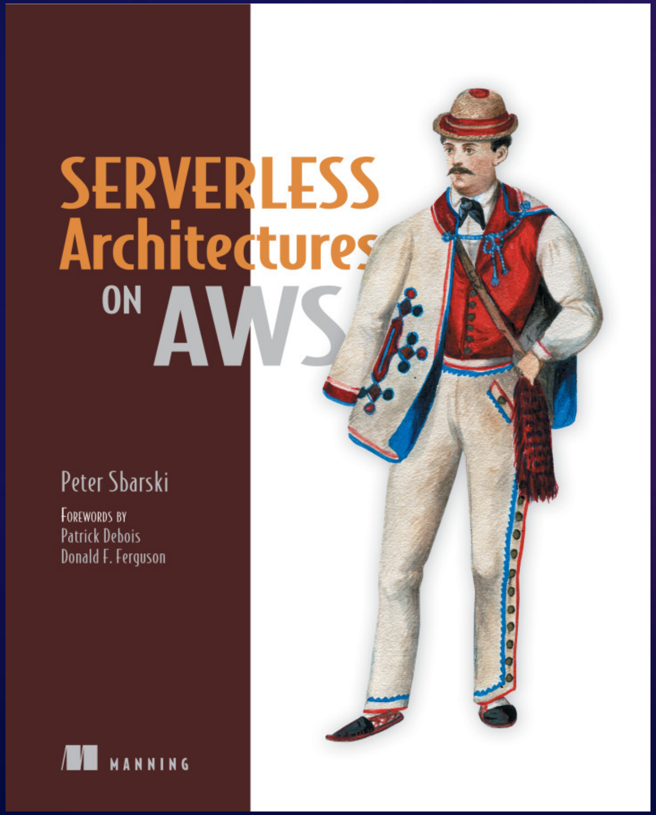
Streams



A stream can be sharded
if order is not important

How can you get started?

- **serverlessconf.io** & video.serverlessconf.io
- Follow **@serverlessconf** for serverlessconf info
- A Cloud Guru **acloud.guru/serverless**
- Book: “Serverless Architectures on AWS”
  https://book.acloud.guru
- Follow **@acloudguru** and **@sbarski**



SERVERLESS Architectures ON AWS

Peter Sbarski

Forewords by
Patrick Debois
Donald F. Ferguson

MANNING

A CLOUD GURU

Thanks :-)

https://acloud.guru
https://serverlessconf.io
@sbarski

goto; chicago

Please

Remember to rate this session

Thank you!

follow us @gotochgo