

# A quick intro to serverless for web developers

James Beswick, AWS Serverless

November 16, 2020

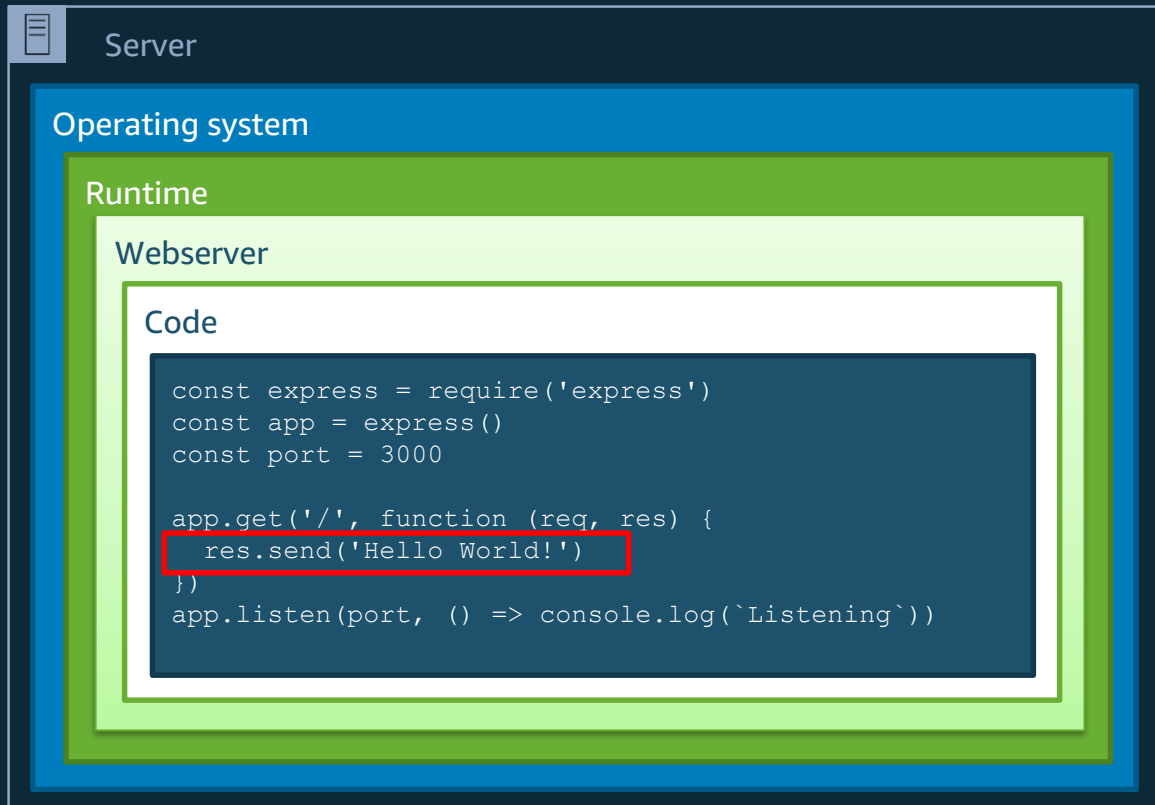
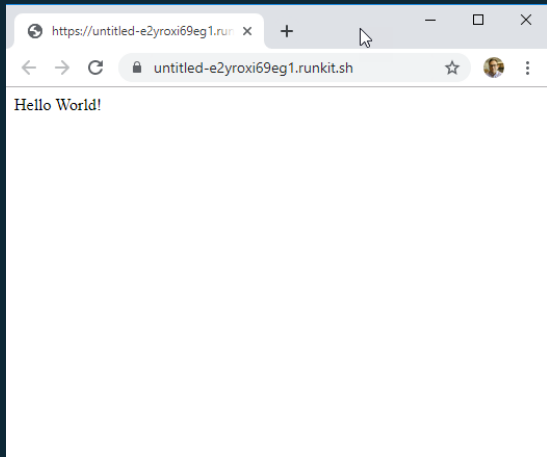
# About me



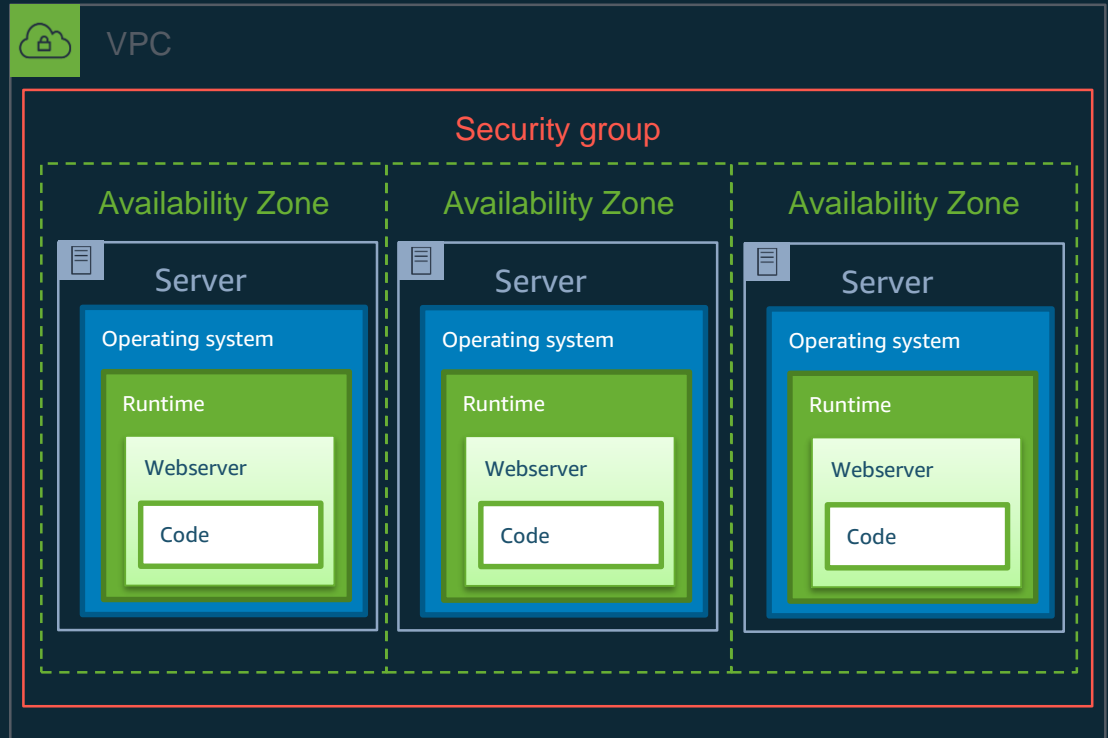
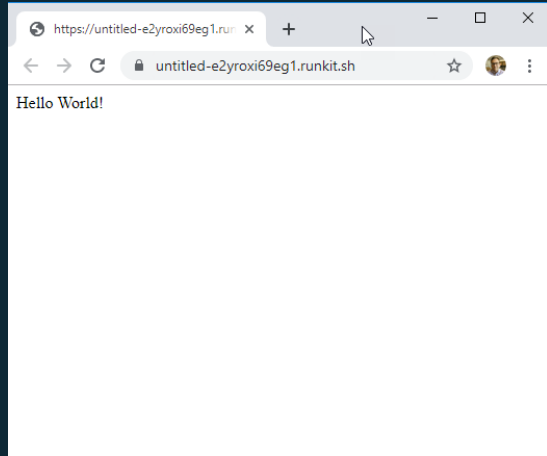
- James Beswick
  - Email: [jbeswick@amazon.com](mailto:jbeswick@amazon.com)
  - Twitter: [@jbesw](https://twitter.com/jbesw)
- Developer Advocate – AWS Serverless
- Self-confessed serverless geek
- Previously:
  - Software Developer and Product Manager
  - Multiple start-up tech guy
  - Rackspace, USAA, Morgan Stanley, J P Morgan
  - Enjoys comedy, travel, coffee and theme parks...

# Hello World

# A "Hello World!" application



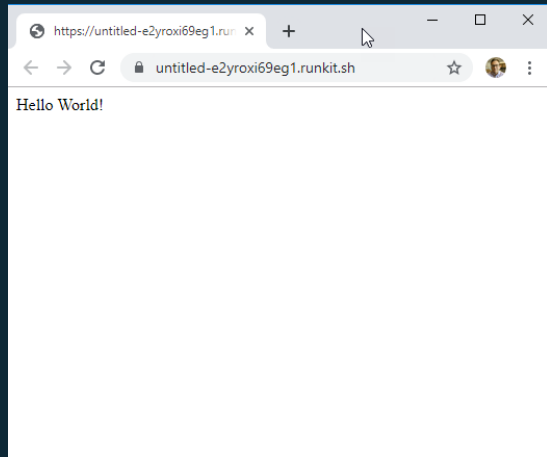
# A “Hello World!” application - at scale...



But I just want to run...

```
res.send('Hello World!')
```

# "Hello World!" – The serverless way



## Code

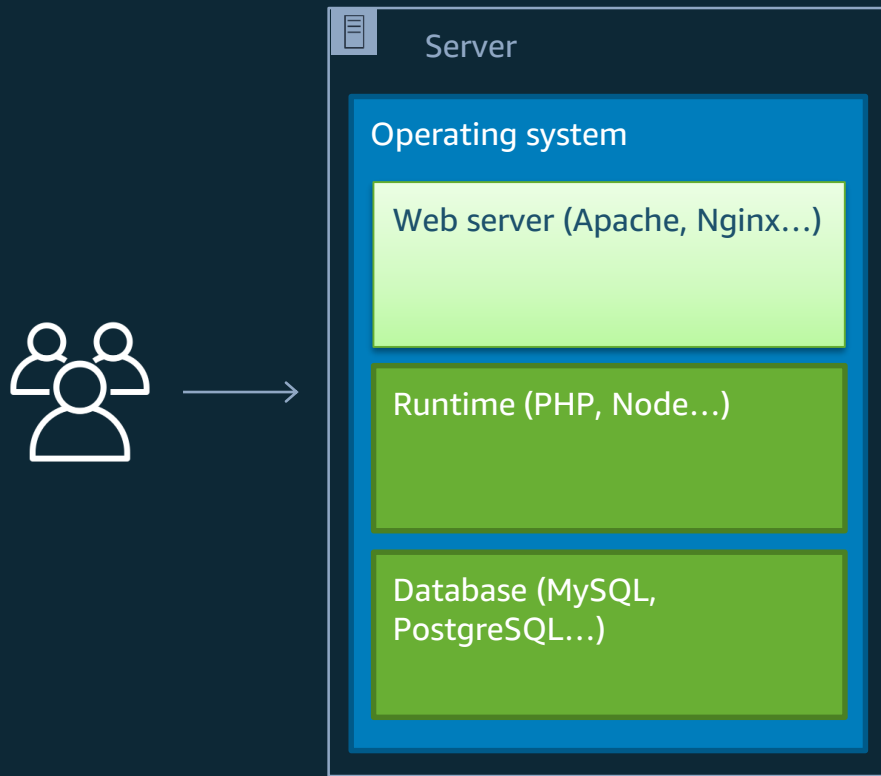
```
exports.handler = async (event) => {  
  return "Hello World!"  
}
```

**Focus on your business logic, not  
the infrastructure.**



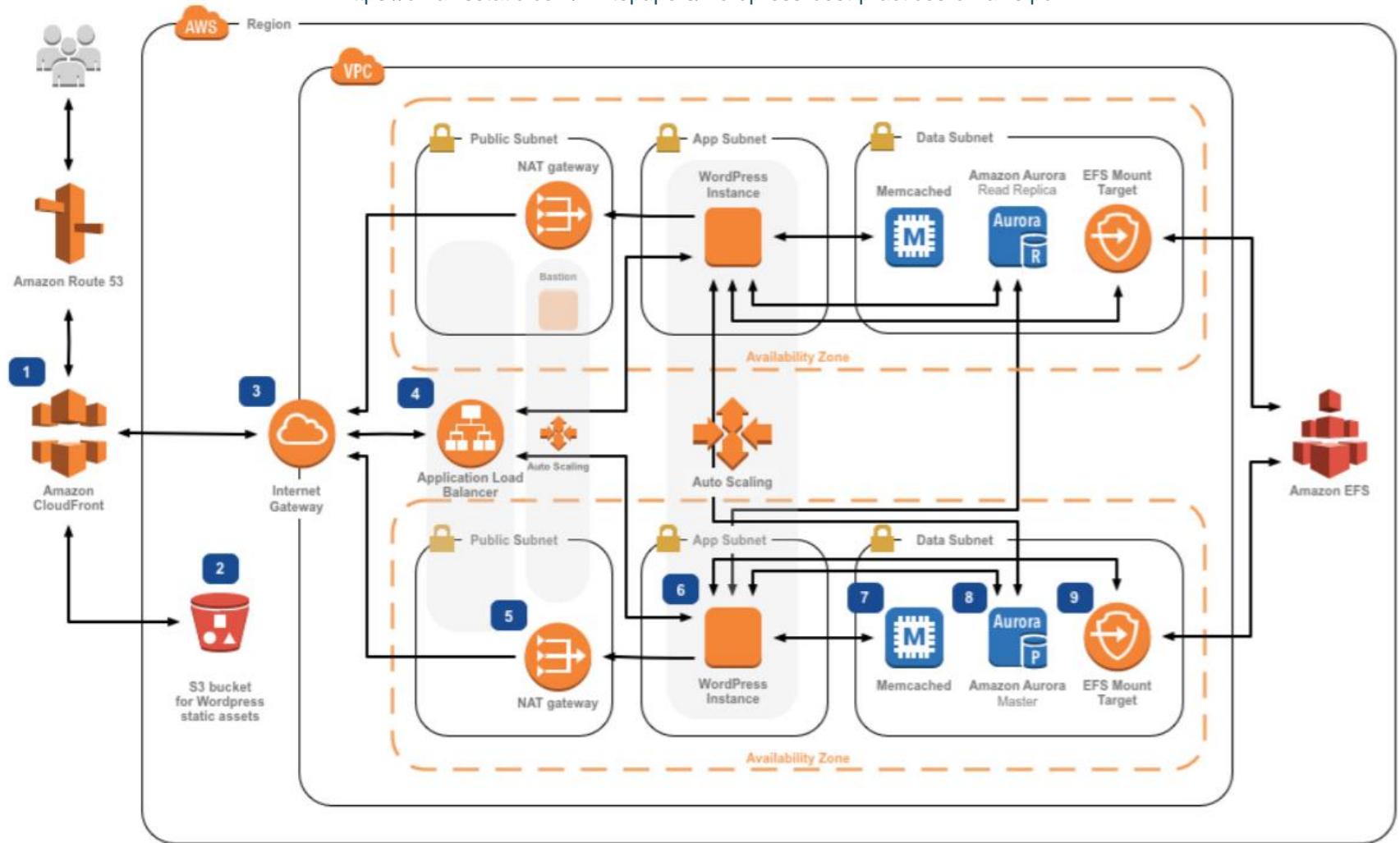
# It's about simplicity.

# A classic web app stack

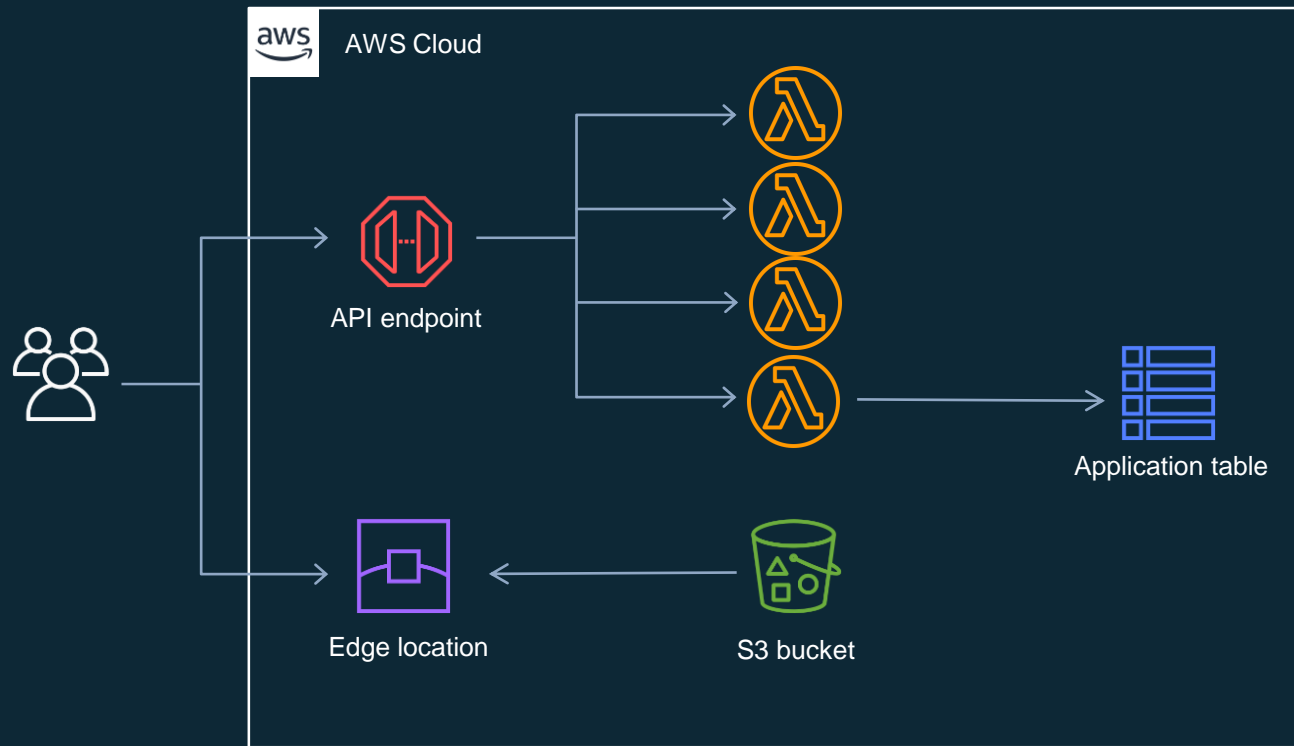


## Things to consider:

- Vertical scaling
- Horizontal scaling
- Load balancing
- Availability
- Security
- Maintenance
- Monitoring
- And... actual features



# Serverless web app architecture



# The voting website



Sign in

Get started



ITNEXT

by LINKIT

ABOUT ITNEXT

WRITE FOR ITNEXT

MEETUPS

SUMMIT

FEATURED

VACANCIES @ LINKIT

# Build a voting website that doesn't crash under load (in under an hour)



James Beswick

Follow

Mar 21 · 4 min read

*Spiky traffic? Unpredictable load? Sounds like a job for serverless.*

Petition websites can experience the extremes of demand — when a popular motion is set before the public, hundreds of thousands of people can appear from nowhere. Worse yet, many stay after voting, F5-ing their browser to see the new tallies, adding even more load onto the already-smoking servers:



**Petitions Committee** @HoCpetitions · Mar 21, 2019



We're very sorry that the site is still experiencing problems. We are working urgently to get it back up and running as soon as possible.



**Petitions Committee** @HoCpetitions

As many of you have guessed, the number of people using the site has caused problems this morning. It's a mix of people reloading the front page to watch the signature count go up and people trying to sign petitions.

1,171 6:53 AM - Mar 21, 2019



692 people are talking about this



AskJames - The Poll Counter X +

vote.jbes.dev

## Welcome to the Serverless Voting App

Click to vote on this very important issue.  
You can vote as many times as you like. Click away!

Vote Yes

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Button

75036 voted

Vote No

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Button

28742 voted

Questions? Ask James @jbesw.

https://vote.jbes.dev

```
! loadTest.yml • README.md
1 config:
2   target: 'https://j1xst3l8aa.execute-api.us-east-1.amazonaws.com/dev/voteyes'
3   phases:
4     - duration: 60
5       arrivalRate: 20
6   defaults:
7     headers:
8       x-my-service-auth: 'yourServiceName'
9   scenarios:
10    - flow:
11      - get:
12        url: "/"
13
```

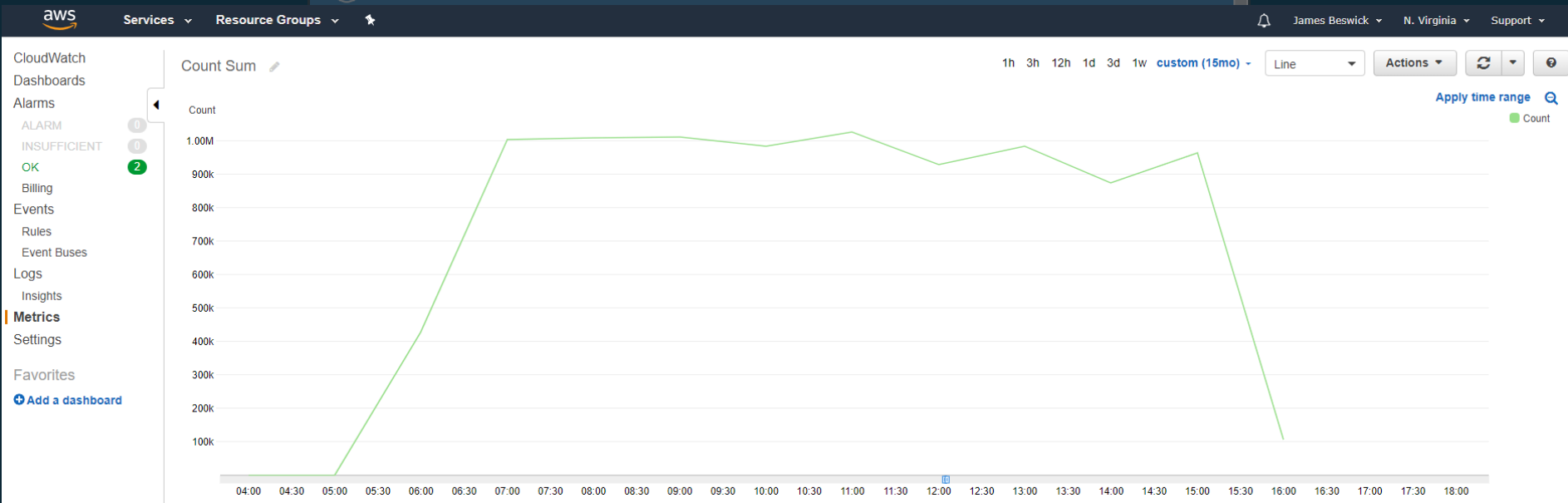
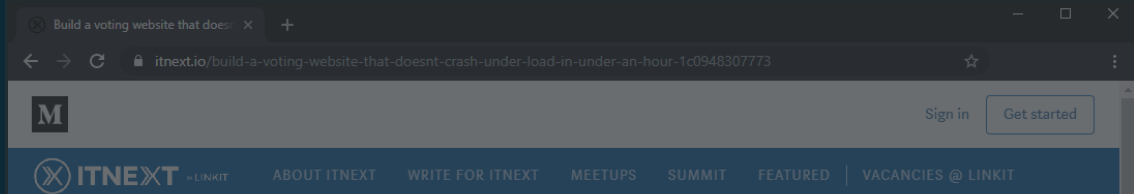
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
All virtual users finished
Summary report @ 10:31:02(-0400) 2019-03-21
Scenarios launched: 1200
Scenarios completed: 1200
Requests completed: 1200
RPS sent: 19.55
Request latency:
  min: 168.2
  max: 3314.7
  median: 242.6
  p95: 336.1
  p99: 544.8
Scenario counts:
  0: 1200 (100%)
Codes:
  200: 1200
```

site has caused problems this morning. It's a mix of people  
reloading the front page to watch the signature count go up and  
people trying to sign petitions  
6:53 AM - Mar 21  
692 people are talking about this

# https://vote.jbes.dev





working urgently to get it back up and running as soon as possible.



As many of you have guessed, the number of people using the site has caused problems this morning. It's a mix of people reloading the front page to watch the signature count go up and people trying to sign petitions.

6:53 AM - 14 Feb 2020

692 people are talking about this

<https://vote.jbes.dev>

# Things I didn't have to do...

Managing load balancers. Managing SSH keys. Monitoring hardware errors. Migrating instances. Migrating data centers. Dynamic volume provisioning. Orchestrating containers. Rotating root admin keys. Configuring subnets and VPCs. Handling datacenter replication. Renewing certificates. Monitoring disk usage. Resizing clusters. Auto-scaling instances. Restarting servers. `sudo yum update`. Defining network policies. Scheduling operating system updates. Troubleshooting networking. Configuring IP tables. Volume snapshots. Running server healthchecks.

# Agility

# The time when you know what to build...



Beginning of  
project?



End of project?



Along the way?



Non-stop changes?

# Why does serverless help with agility?



Less code, focused  
on business logic



Microservices are  
more agile



Services for  
common tasks



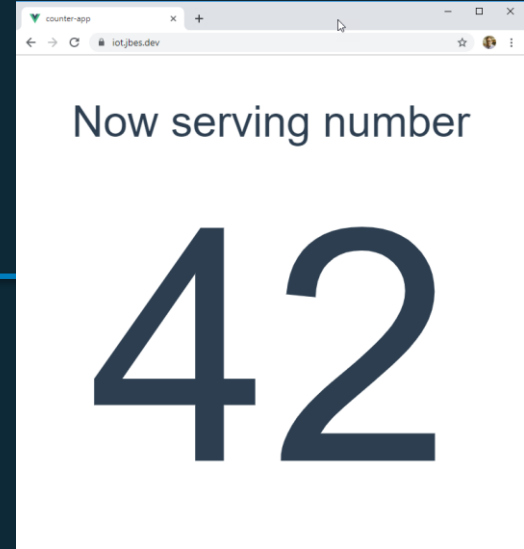
CI/CD, versioning  
and automation

# What a serverless application looks like



# What a serverless application looks like

```
1 exports.handler = async (event) => {  
2   ...  
3   const payload = event.clickType  
4   const params = {  
5     topic: 'turnstyle-counter',  
6     payload,  
7     qos: 0  
8   }  
9   try {  
10    const result = await iot.publish  
11    | | | | | (params).promise()  
12    return { statusCode: 200 }  
13  } catch (err) {  
14    console.error('Error! ', err)  
15    return { statusCode: 501 }  
16  }  
17 }
```



# What a serverless application looks like

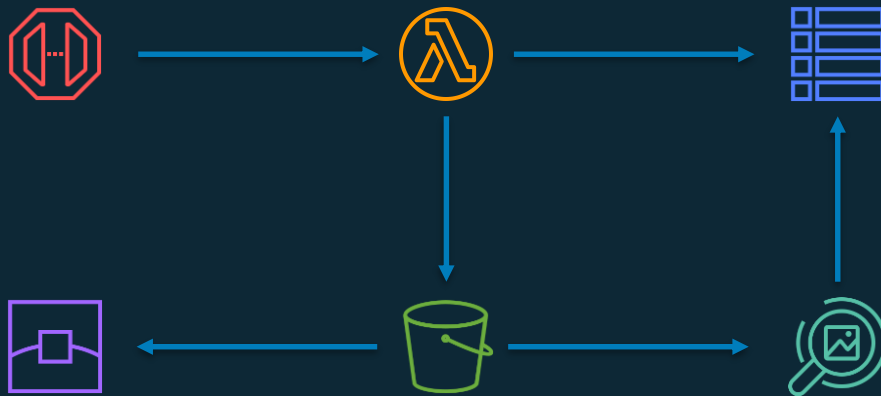




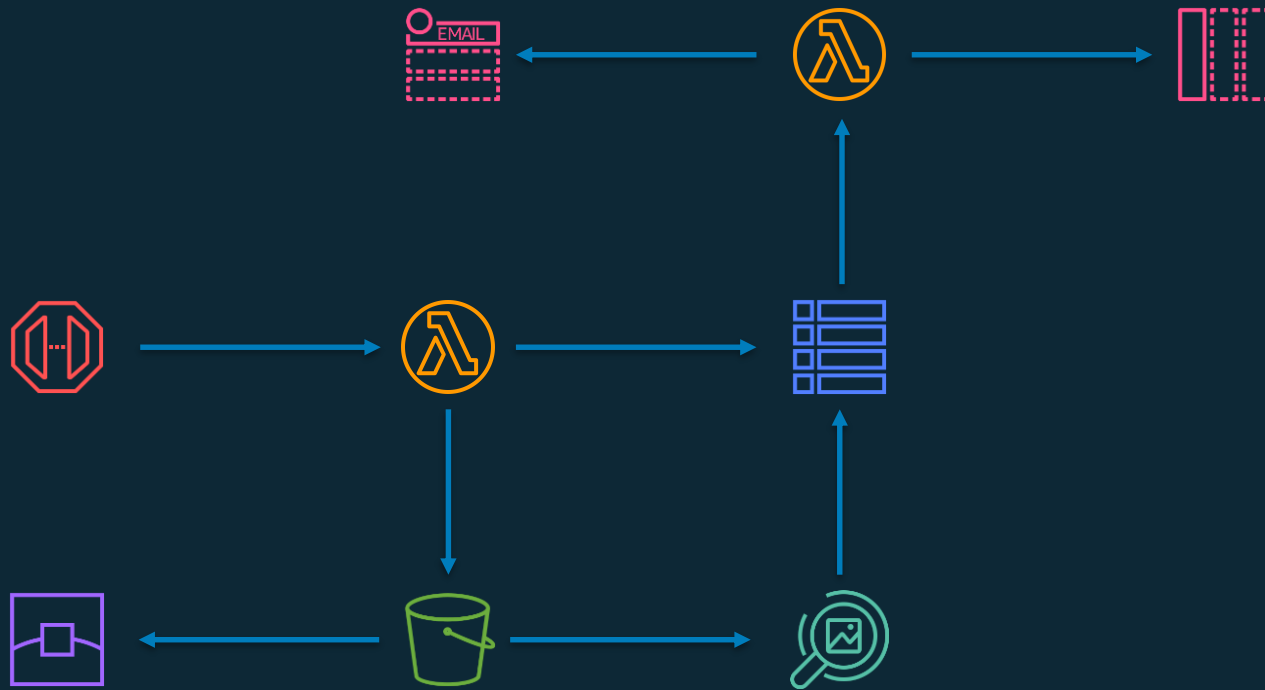
# Serverless applications



# Serverless applications



# Serverless applications



# Application services

## Machine Learning



Amazon SageMaker



Amazon Comprehend



Amazon Lex



Amazon Polly



Amazon Rekognition



Amazon Rekognition Image



Amazon Rekognition Video



Amazon Translate



Amazon Transcribe



AWS DeepLens



AWS Deep Learning AMIs

## Internet of Things



AWS IoT Core



Amazon FreeRTOS



AWS Greengrass



AWS IoT 1-Click



AWS IoT Analytics



AWS IoT Button



AWS IoT Device Defender



AWS IoT Device Management

## Analytics



Amazon Athena



Amazon Kinesis Data Analytics



Amazon Redshift



Amazon QuickSight



AWS Glue



AWS Batch

## Media



Amazon Elastic Transcoder



Amazon Kinesis Video Streams



AWS Elemental MediaConvert



AWS Elemental MediaLive



AWS Elemental MediaPackage



AWS Elemental MediaStore



AWS Elemental MediaTailor

## Web/Mobile/Digital



Amazon Connect



Amazon Pinpoint



Amazon Simple Email Service



AWS AppSync



AWS Device Farm



Amazon Pinpoint

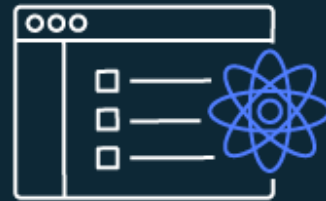
# Common serverless application types



Web applications



Backends



Data processing



Chatbots



Amazon Alexa



IT Automation

# Fundamental changes to development



Agility

# Fundamental changes to development



Agility



Scaling

# Fundamental changes to development



Agility



Scaling



Cost



# Fundamental changes to development



Agility



Scaling



Cost



Environment

# Thank you!

James Beswick, AWS Serverless  
@jbesw