Serverless Tour of Heroes

Jon Gear
@GearedUpTech
Github.com/JonGear

Agenda

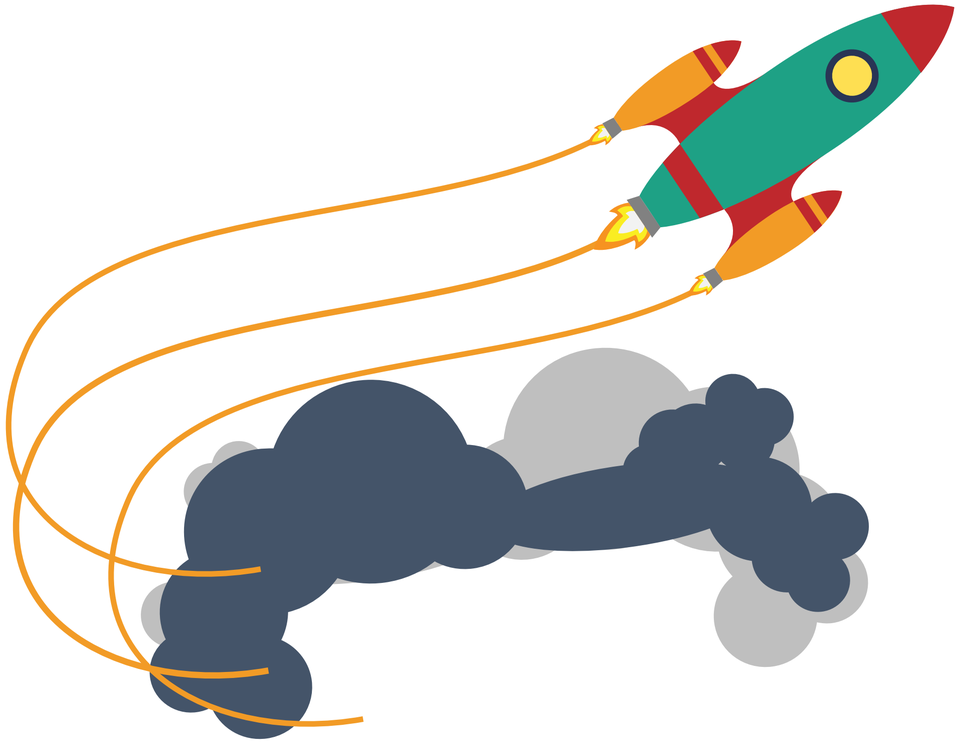
Building serverless enterprises of the future

1 What is serverless?

2 Why serverless?

3 Serverless ToH

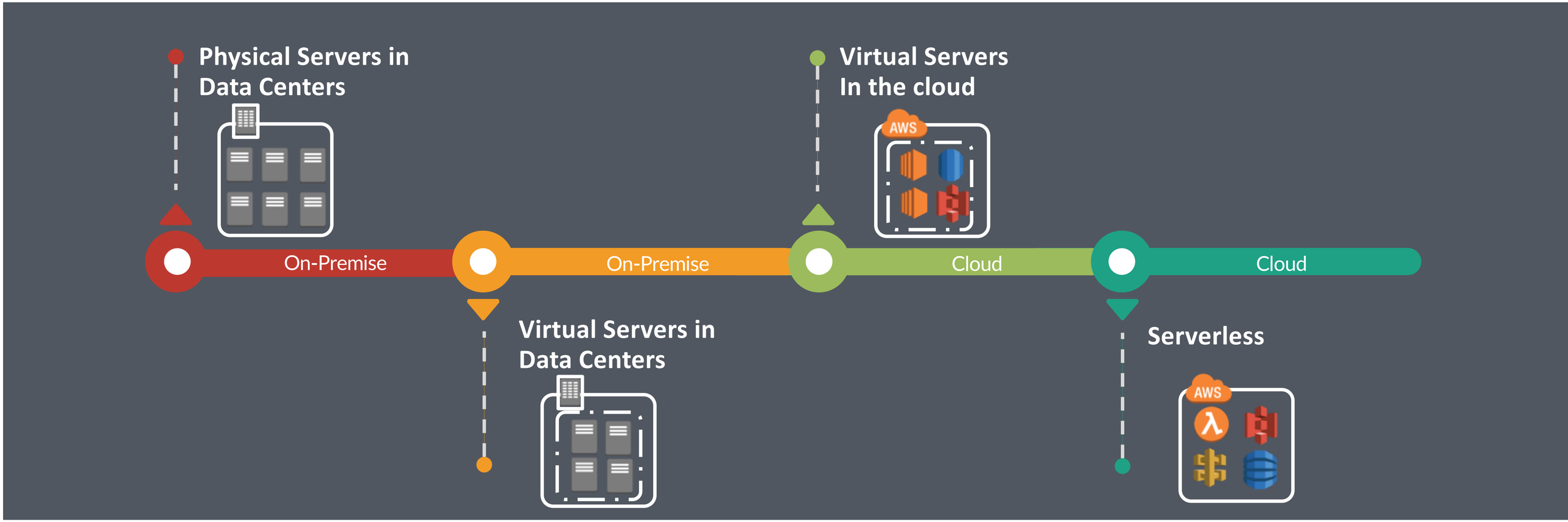
4 Next steps



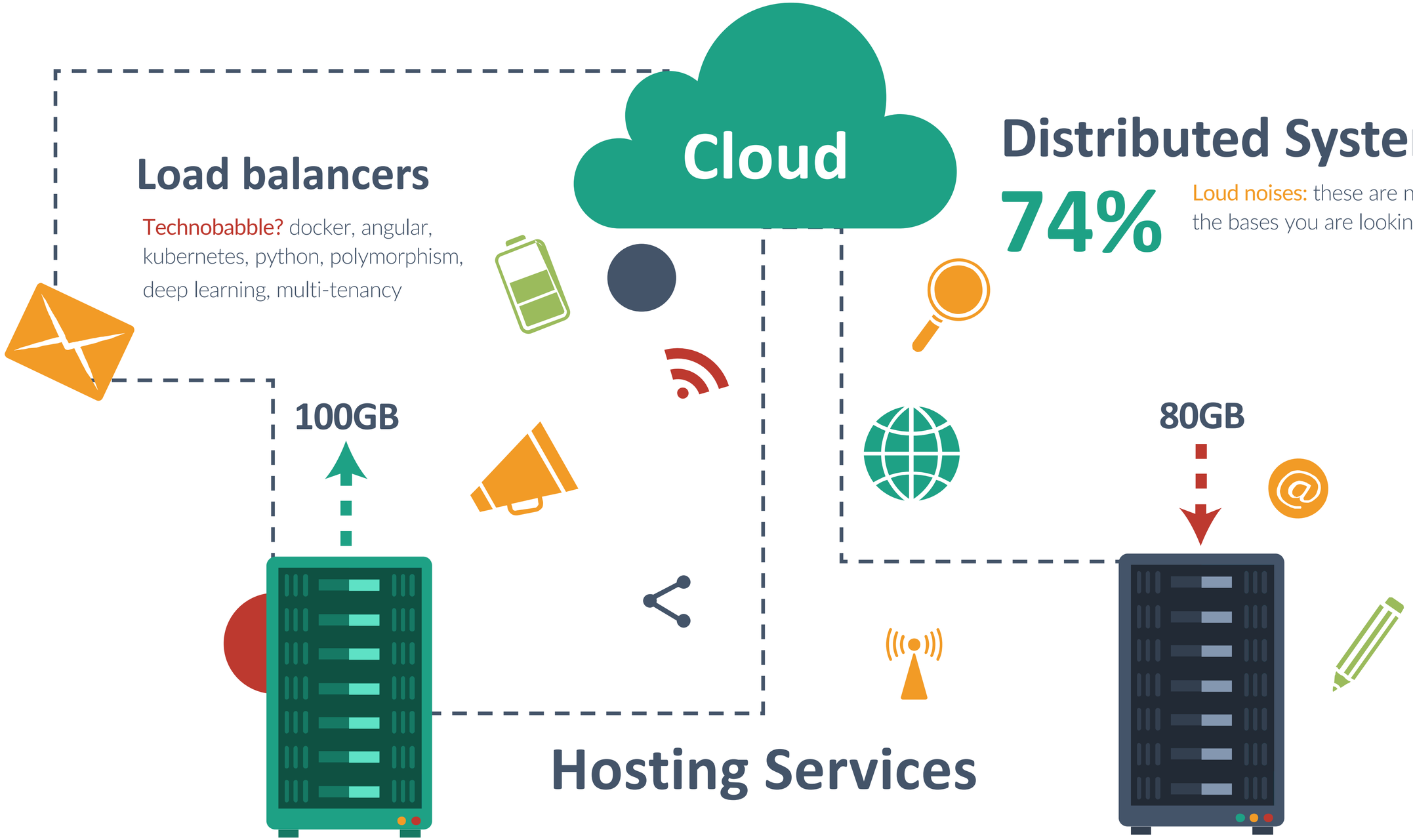
github.com/jongear/serverless-tour-of-heroes

Evolution of computing

The quest to abstract limitations



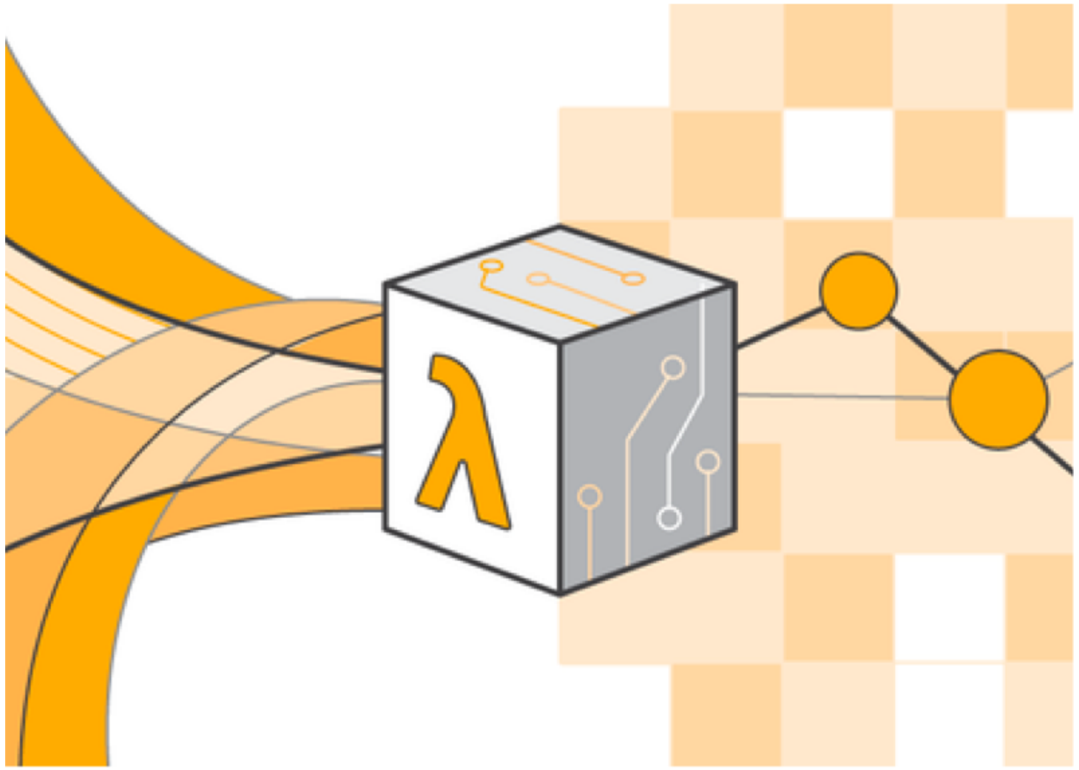
github.com/jongear/serverless-tour-of-heroes



github.com/jongear/serverless-tour-of-heroes

Serverless means unit of work

Serverless is a mindset shift



Serverless means just your code and nothing else

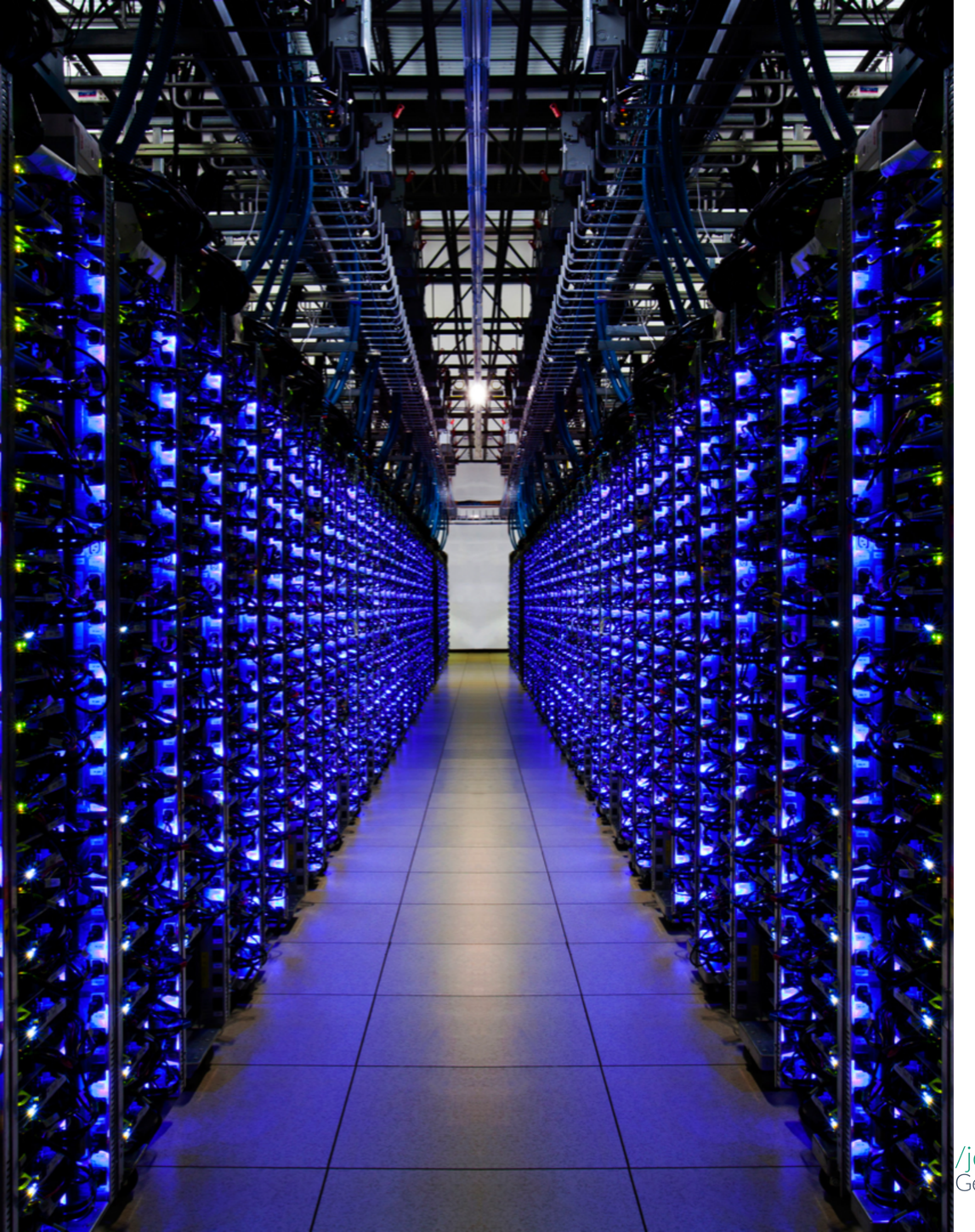
No VM startup scripts, infrastructure is managed.

Serverless means automatic scalability

Infrastructure scales under load and disposes when idle

Serverless means stateless

No session sharing. No context recall. Functions are idempotent



Yes there are servers!

We trade explicit server control for ease of deployment and scalability

/jongear/serverless-tour-of-heroes

Serverless Platforms

Molding the future of on demand compute power

"Data is the competitive differentiator. The quality of data and the questions you are able to ask of the data have become the major differentiator among companies."

DR WERNER VOGELS

CTO, Amazon.com

[AWS re:Invent 2016 Keynote]

AWS

Lambda

Nov 2014

Google Cloud

Functions

Feb 2016

Azure

Functions

Mar 2016

github.com/jongear/serverless-tour-of-heroes

A Compute Currency

Limitations of a unit of work

128MB – 1.5GB

Memory Allocation

Acts as a dial. Adjusted in 64MB increments

1 - 300 Seconds

Execution Duration

Default timeout is 3 seconds

50MB / 250MB

Deployment Package Size

50MB compressed deployment package limit.
250MB uncompressed package limit.

***1000 Limit**

Concurrent Executions

Default: 1000 executions

Select regions default 3000

Can request to have safety limit increased

Compute Integration

Integrating with a unit of work

API Gateway

IoT

Alexa Skills Kit

Alexa Smart Home Skill

CloudFront

CloudWatch Events

CloudWatch Logs

CodeCommit

Cognito Sync Trigger

DynamoDB

Kinesis

S3

SNS

github.com/jongear/serverless-tour-of-heroes

Serverless Tour of Heroes

Dashboard

Detail view

github.com/jongear/serverless-tour-of-heroes

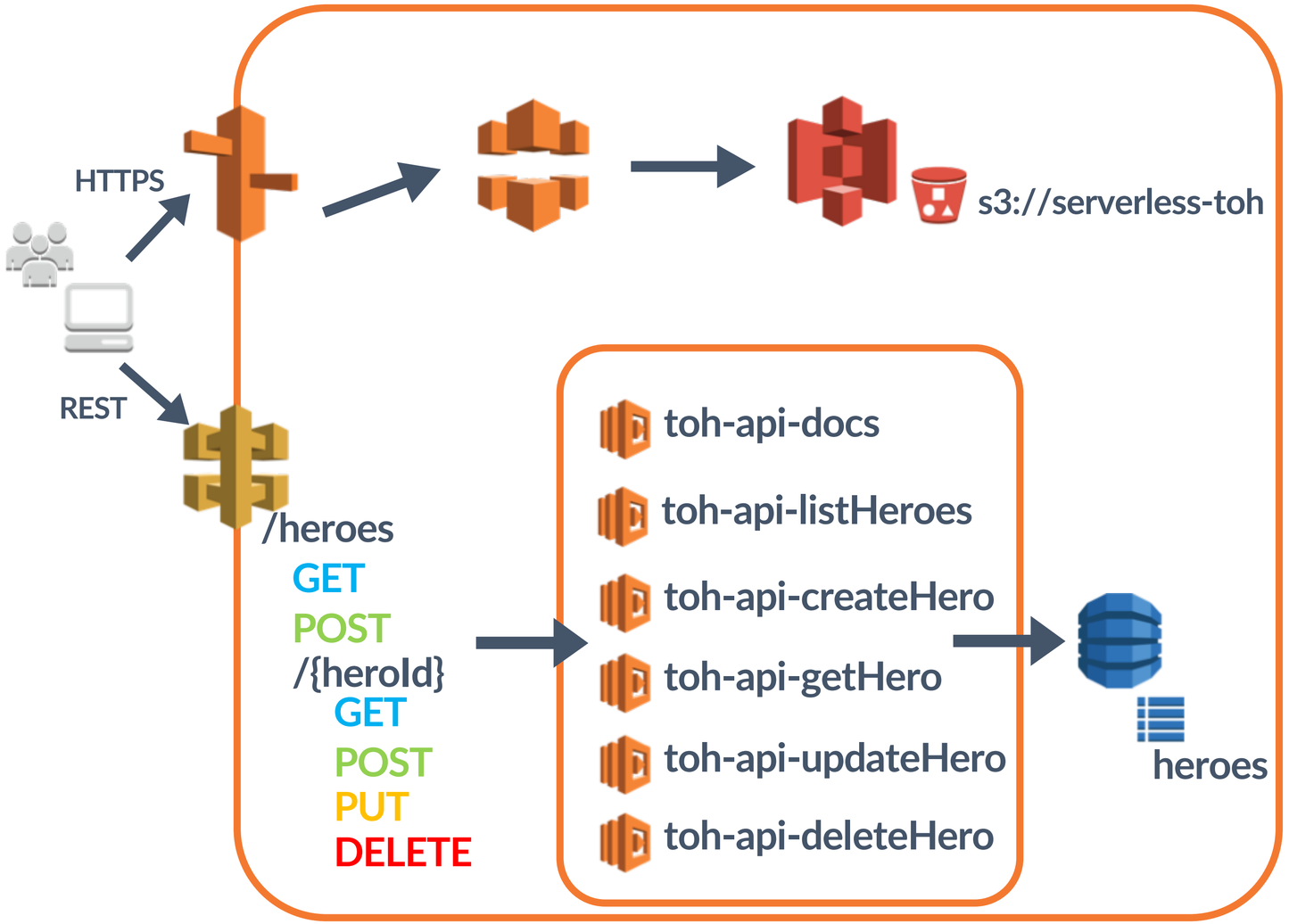
Serverless Tour of Heroes

Highly responsive web applications of the future

github.com/jongear/serverless-tour-of-heroes

Serverless Tour of Heroes

Highly responsive web applications of the future



Highly responsive

Zero downtime

Endpoint level deployment granularity

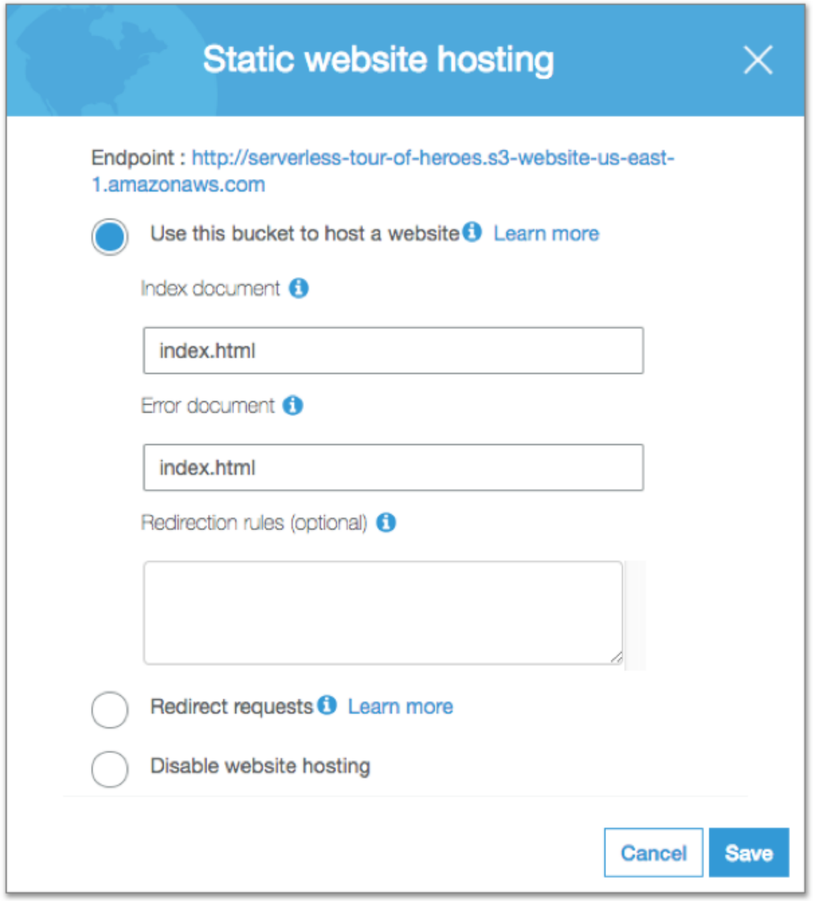
Bonus

- API Gateway adds DDOS prevention
- Swagger API export capabilities

github.com/jongear/serverless-tour-of-heroes

Website

Highly responsive web applications of the future



github.com/jongear/serverless-tour-of-heroes

Website

Highly responsive web applications of the future

General | Origins | Behaviors | Error Pages | Restrictions | Invalidations | Tags

Edit

Distribution ID	[redacted]
ARN	arn:aws:cloudfront::[redacted]:distribution/[redacted]
Log Prefix	-
Delivery Method	Web
Cookie Logging	Off
Distribution Status	InProgress
Comment	-
Price Class	Use All Edge Locations (Best Performance)
AWS WAF Web ACL	-
State	Enabled
Alternate Domain Names (CNAMEs)	-
SSL Certificate	Default CloudFront Certificate (*.cloudfront.net)
Domain Name	[redacted].cloudfront.net
Custom SSL Client Support	-
Security Policy	TLSv1
Supported HTTP Versions	HTTP/2, HTTP/1.1, HTTP/1.0
IPv6	Enabled
Default Root Object	index.html
Last Modified	2018-06-19 22:09 UTC-5
Log Bucket	-

github.com/jongear/serverless-tour-of-heroes

Website

Highly responsive web applications of the future

General | Origins | Behaviors | **Error Pages** | Restrictions | Invalidations | Tags

You can configure CloudFront to respond to requests using a custom error page when your origin returns an HTTP 4xx or 5xx status code. For example, when your custom origin is unavailable and returning 5xx responses, CloudFront can return a static error page that is hosted on Amazon S3. You can also specify a minimum TTL to control how long CloudFront caches errors. For more information, see Customizing Error Responses in the *Amazon CloudFront Developer Guide*.

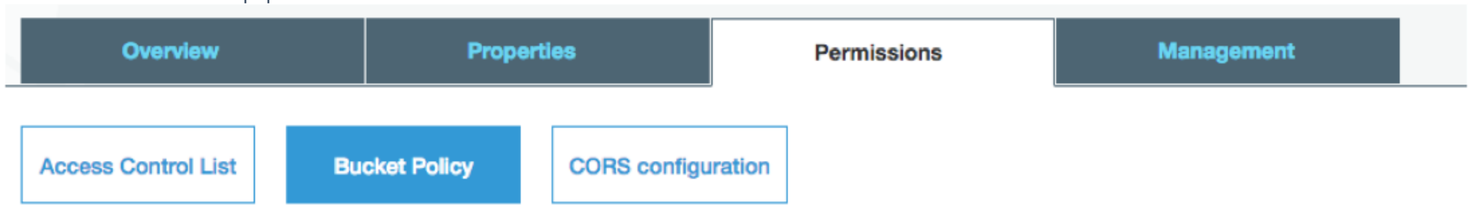
Create Custom Error Response | Edit | Delete

	HTTP Error Code	Error Caching Minimum TTL	Response Page Path	HTTP Response Code
☐	403	300	/index.html	403
☐	404	300	/index.html	200

Website BONUS

Highly responsive web applications of the future

Lockdown to CloudFront



Bucket policy editor ARN: arn:aws:s3:::serverless-tour-of-heroes

Type to add a new policy or edit an existing policy in the text area below.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "2",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity [redacted]",
                    "arn:aws:iam::[redacted]:user/serverless-admin"
                ]
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::serverless-tour-of-heroes/*"
        }
    ]
}
```

Origin Domain Name and Path	Origin ID	Origin Type	Origin Access Identity
serverless-tour-of-heroes.s3.amazonaws.com	S3-serverless-tour-of-heroes	S3 Origin	origin-access-identity/cloudfront/[redacted]

github.com/jongear/serverless-tour-of-heroes

Website Deployment

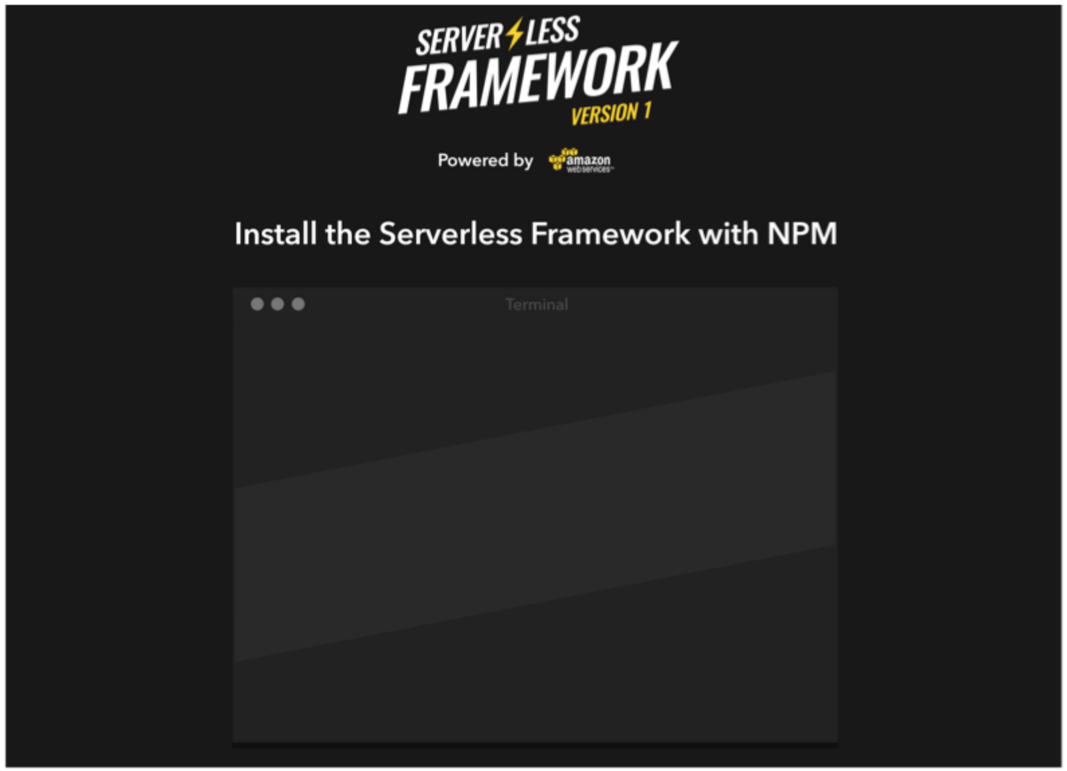
Highly responsive web applications of the future

```yaml
# .travis.yml

language: python
python:
  - '2.7'
sudo: required

cache:
  pip: true
  directories:
    - node_modules

branches:
  only:
    - master

install:
  - nvm install 9.4.0
  - npm install
  - pip install awscli

script:
  - npm run lint
  - npm test
  - npm run-script ng build -- --prod --aot --configuration=dev

deploy:
  - provider: s3
    skip_cleanup: true
    local_dir: dist
    access_key_id: "$AWS_ACCESS_KEY_ID"
    secret_access_key:
      secure: "$AWS_SECRET_ACCESS_KEY"
    bucket: "$AWS_BUCKET"
    on:
      branch: master

after_deploy:
  - aws cloudfront create-invalidation --distribution-id $AWS_CLOUDFRONT_DISTRIBUTION_ID --paths "/*"
```

github.com/jongear/serverless-tour-of-heroes

Serverless Framework

One deployment specification for all platforms



Specification is written in YML

Attempts to abstract CloudFormation complexities away

Granular and full stack deployment capabilities

Capable or rolling a single endpoint or an entire service

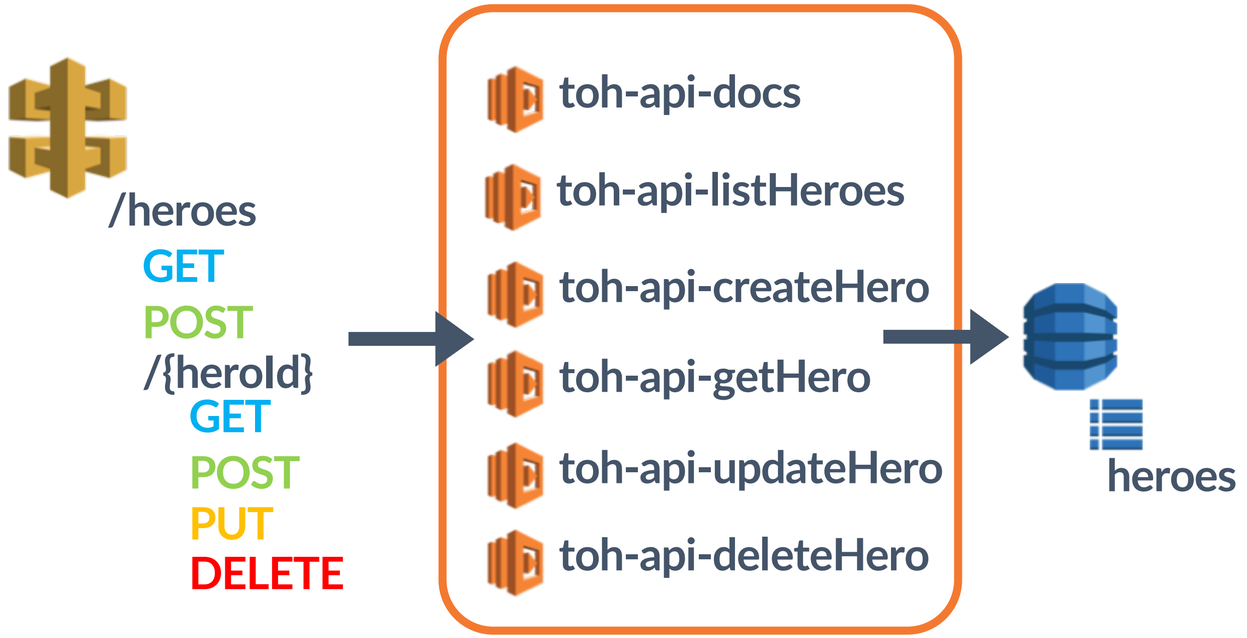
Environment variable and tagging support

Manage Dev, QA, Prod deployments effortlessly

**Learn more about the Serverless Framework*
https://serverless.com

API

Highly responsive web applications of the future

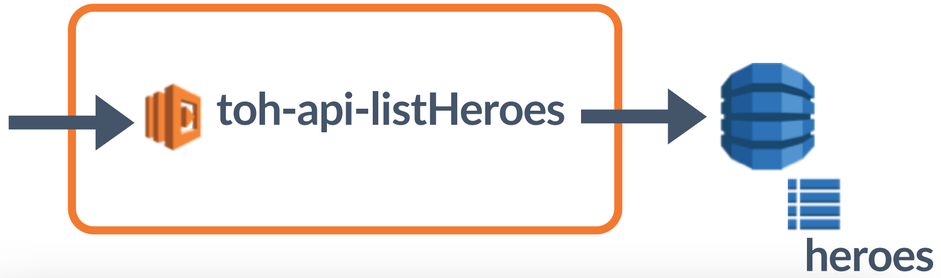


github.com/jongear/serverless-tour-of-heroes

API

Highly responsive web applications of the future

/heroes
GET




```yaml
# serverless.yml

service: toh-api

plugins:
  - serverless-webpack
  - serverless-offline

package:
  individually: true

custom:
  stage: "${opt:stage, self:provider.stage}"
  webpack:
    webpackConfig: ./webpack.config.js
    includeModules: true
  serverless-offline:
    port: 3000

provider:
  name: aws
  runtime: nodejs8.10
  region: us-east-1
  environment: ${file(env.yml):${self:custom.stage}}

  iamRoleStatements:
    - Effect: Allow
      Action:
        - dynamodb:DescribeTable
        - dynamodb:Query
        - dynamodb:Scan
        - dynamodb:GetItem
        - dynamodb:PutItem
        - dynamodb:UpdateItem
        - dynamodb:DeleteItem
      Resource:
        - "Fn::GetAtt": [ HeroesTable, Arn ]

functions:
  listHeroes:
    handler: src/heroes/list.handler
    events:
      - http:
          path: /heroes
          method: GET
          cors: true

resources:
  - ${file(resources/dynamodb-table.yml)}
```

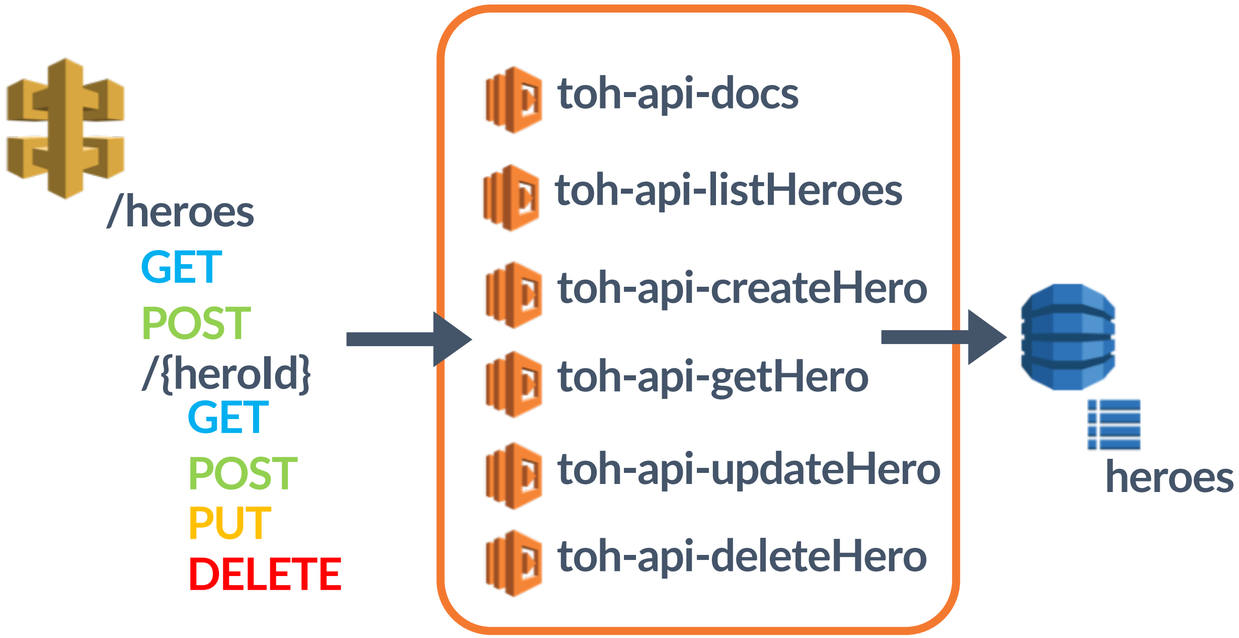
```yaml
# resources/dynamodb-table.yml

Resources:
  HeroesTable:
    Type: AWS::DynamoDB::Table
    Properties:
      # Generate a name based on the stage
      TableName: ${self:custom.stage}-heroes
      AttributeDefinitions:
        - AttributeName: id
          AttributeType: S
      KeySchema:
        - AttributeName: id
          KeyType: HASH
      # Set the capacity based on the stage
      ProvisionedThroughput:
        ReadCapacityUnits: 4
        WriteCapacityUnits: 1
```

```javascript
// src/heroes/list.js

import * as dynamo from '../libs/dynamoLib'
import { failure, success } from '../libs/responseLib'

export async function handler(event, context, callback) {
  const params = {
    TableName: process.env.DYNAMODB_TABLE,
  }

  try {
    const result = await dynamo.call('scan', params)
    callback(null, success(result.Items))
  } catch (e) {
    console.log(e)
    callback(null, failure({ status: false }))
  }
}
```

github.com/jongear/serverless-tour-of-heroes

API

Highly responsive web applications of the future




```yaml
# serverless.yml

functions:
  docs:
    handler: src/index.handler
    events:
      - http:
          path: /
          method: GET
          cors: true
  listHeroes:
    handler: src/heroes/list.handler
    events:
      - http:
          path: /heroes
          method: GET
          cors: true
  createHero:
    handler: src/heroes/create.handler
    events:
      - http:
          path: /heroes
          method: POST
          cors: true
  getHero:
    handler: src/heroes/get.handler
    events:
      - http:
          path: /heroes/{id}
          method: GET
          cors: true
  updateHero:
    handler: src/heroes/update.handler
    events:
      - http:
          path: /heroes/{id}
          method: PUT
          cors: true
  deleteHero:
    handler: src/heroes/delete.handler
    events:
      - http:
          path: /heroes/{id}
          method: DELETE
          cors: true

resources:
  - ${file(resources/dynamodb-table.yml)}
```

github.com/jongear/serverless-tour-of-heroes

API

Highly responsive web applications of the future

```
// src/heroes/create.js

import uuid from 'uuid'
import * as dynamo from '../libs/dynamoLib'
import { failure, success } from '../libs/responseLib'

export async function handler(event, context, callback) {
  const data = JSON.parse(event.body)
  const params = {
    TableName: process.env.DYNAMODB_TABLE,
    Item: {
      id: uuid.v1(),
      name: data.name,
      createdAt: Date.now(),
    },
  }

  try {
    await dynamo.call('put', params)
    callback(null, success(params.Item))
  } catch (e) {
    console.log(e)
    callback(null, failure({ status: false }))
  }
}
```

```
// src/heroes/get.js

import * as dynamoDbLib from '../libs/dynamoLib'
import { failure, success } from '../libs/responseLib'

export async function handler(event, context, callback) {
  const params = {
    TableName: process.env.DYNAMODB_TABLE,
    Key: {
      id: event.pathParameters.id,
    },
  }

  try {
    const result = await dynamoDbLib.call('get', params)
    if (result.Item) {
      // Return the retrieved item
      callback(null, success(result.Item))
    } else {
      callback(null, failure({ status: false, error: 'Item not found.' }))
    }
  } catch (e) {
    callback(null, failure({ status: false }))
  }
}
```

github.com/jongear/serverless-tour-of-heroes

API Deployment

Highly responsive web applications of the future

```yaml
# .travis.yml

language: python
python:
  - '2.7'
sudo: required

cache:
  pip: true
  directories:
    - node_modules

branches:
  only:
    - master

install:
  - nvm install 9.4.0
  - npm install -g serverless
  - npm install
  - pip install awscli

script:
  - npm run lint
  - npm test
  - sls --version
  - echo "script finished"

deploy:
  - provider: script
    script: sls deploy --stage dev --verbose --force
    on:
      branch: master
```

github.com/jongear/serverless-tour-of-heroes


Jon Gear
@GearedUpTech
Github.com/JonGear