

# The Quest for Serverless

*Divya Sasidharan*



**HELLO**

MY NAME IS

**Divya Sasidharan**

**HELLO**

MY NAME IS

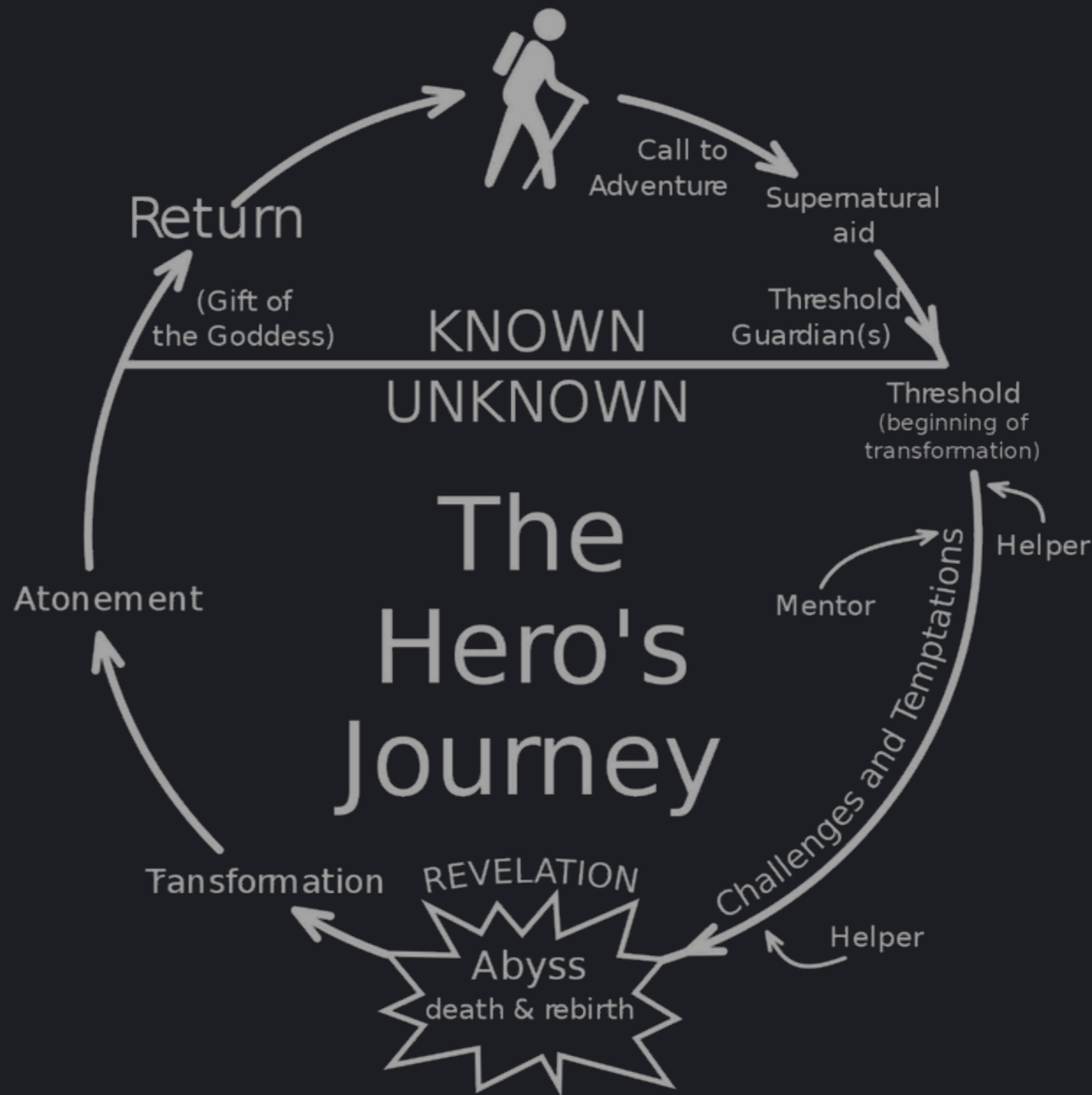
**Divya Sasidharan**

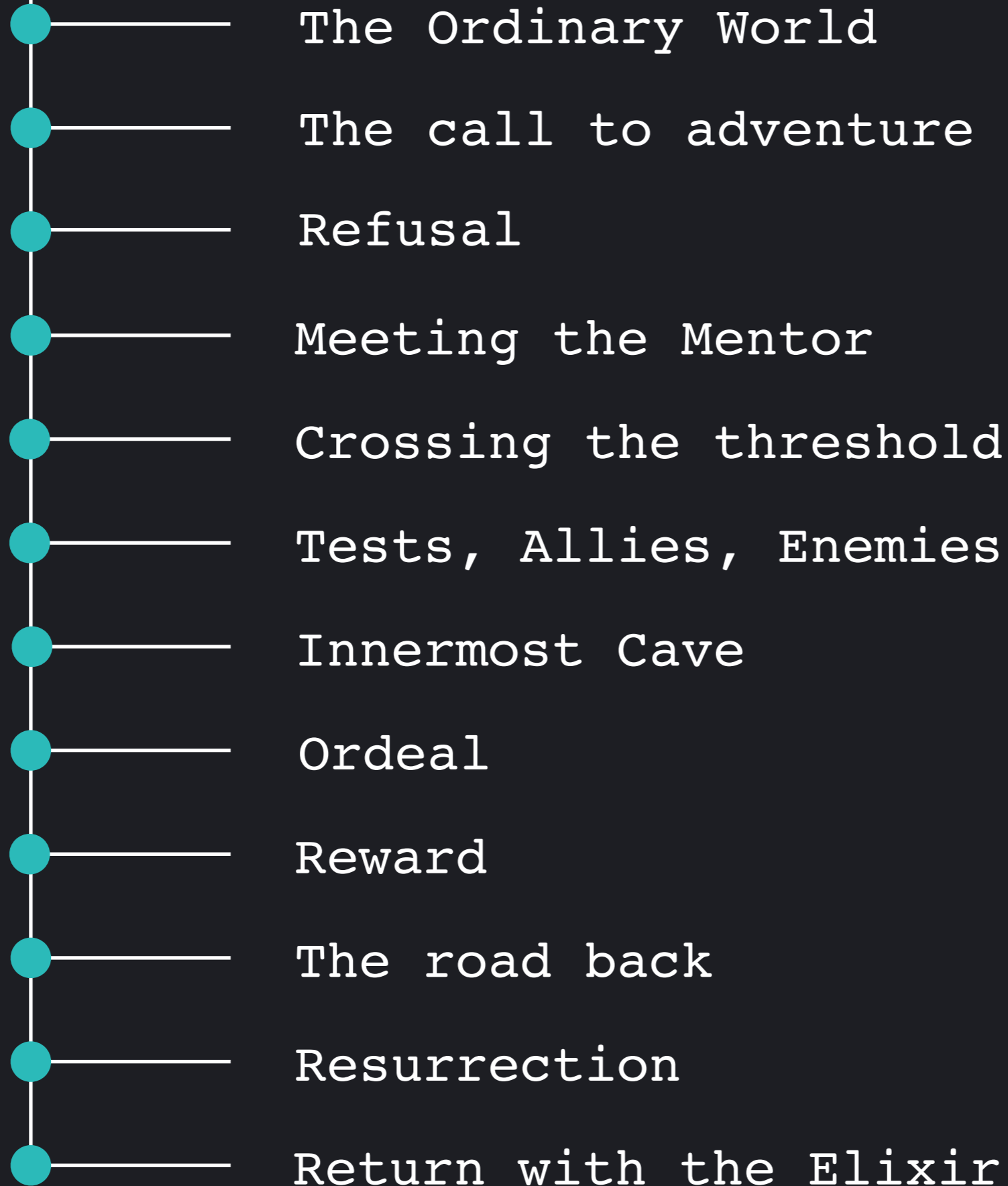
*@shortdiv*



netlify

# THE HERO'S JOURNEY





- 
- The Ordinary World
  - The call to adventure
  - Refusal
  - Meeting the Mentor
  - Crossing the threshold
  - Tests, Allies, Enemies
  - Innermost Cave
  - Ordeal
  - Reward
  - The road back
  - Resurrection
  - Return with the Elixir

**DEPARTURE**



- The Ordinary World
- The call to adventure
- Refusal
- Meeting the Mentor
- Crossing the threshold
- Tests, Allies, Enemies
- Innermost Cave
- Ordeal
- Reward
- The road back
- Resurrection
- Return with the Elixir

**DEPARTURE**

**INITIATION**

- 
- The Ordinary World
  - The call to adventure
  - Refusal
  - Meeting the Mentor
  - Crossing the threshold
  - Tests, Allies, Enemies
  - Innermost Cave
  - Ordeal
  - Reward
  - The road back
  - Resurrection
  - Return with the Elixir

**DEPARTURE**

**INITIATION**

**RETURN**

Act 1

# Departure





*The Ordinary World*

---

DATE	SCENE	ACT
07/09	1	1



*The Ordinary World*

DATE

SCENE

ACT

07/09

1

1



**Sammy Serverus**

BASICALLY

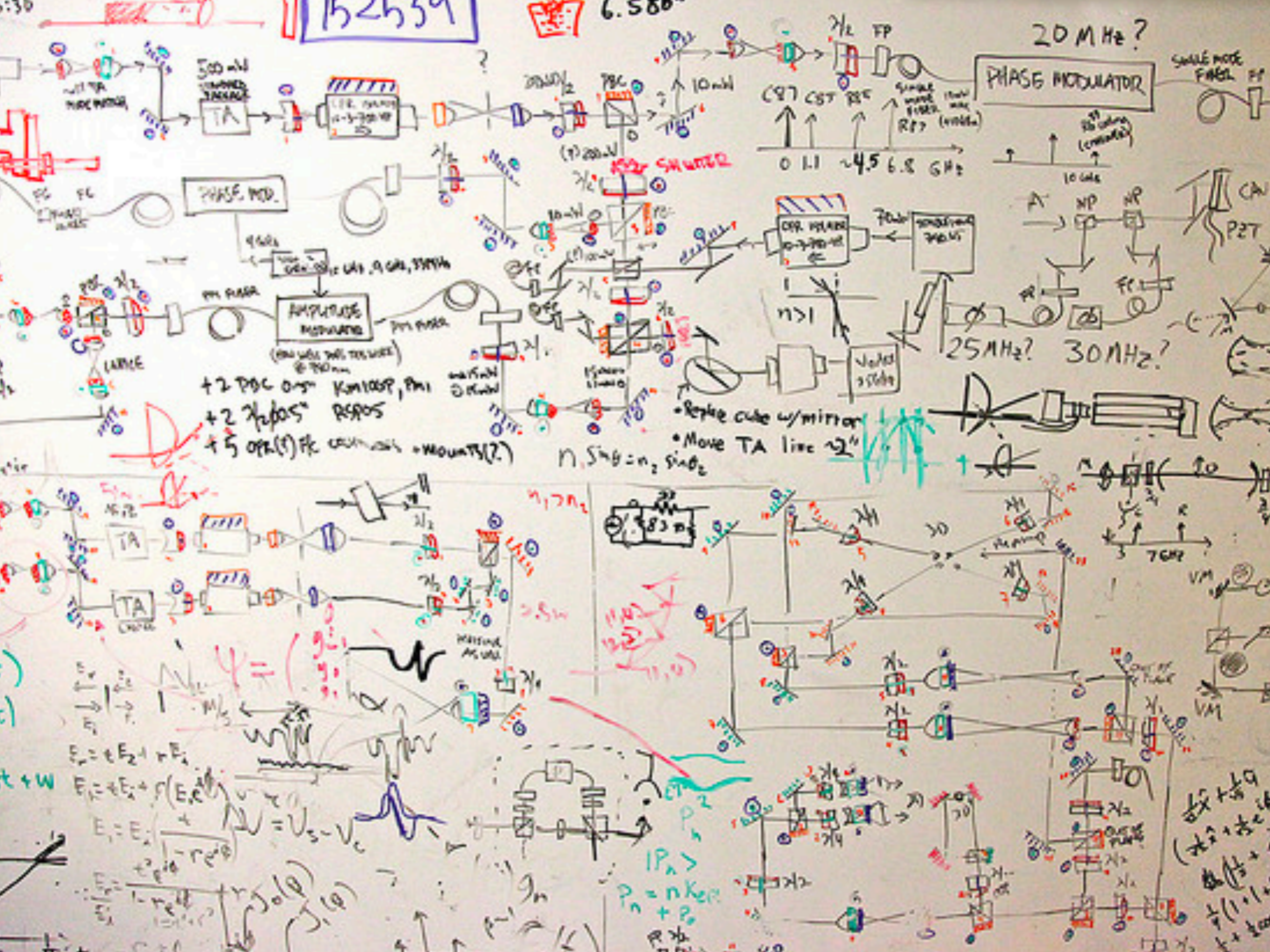


LEARN — TO — COOK

# Spatchcocked Chicken

The best recipe for those of you living under a rock





## Luke's Lobster Back Bay

★★★★★ 4  
1578 Reviews on Yelp

My Rating  
★★★★★ 0

## Neptune Oyster

★★★★★ 4.5  
4740 Reviews on Yelp

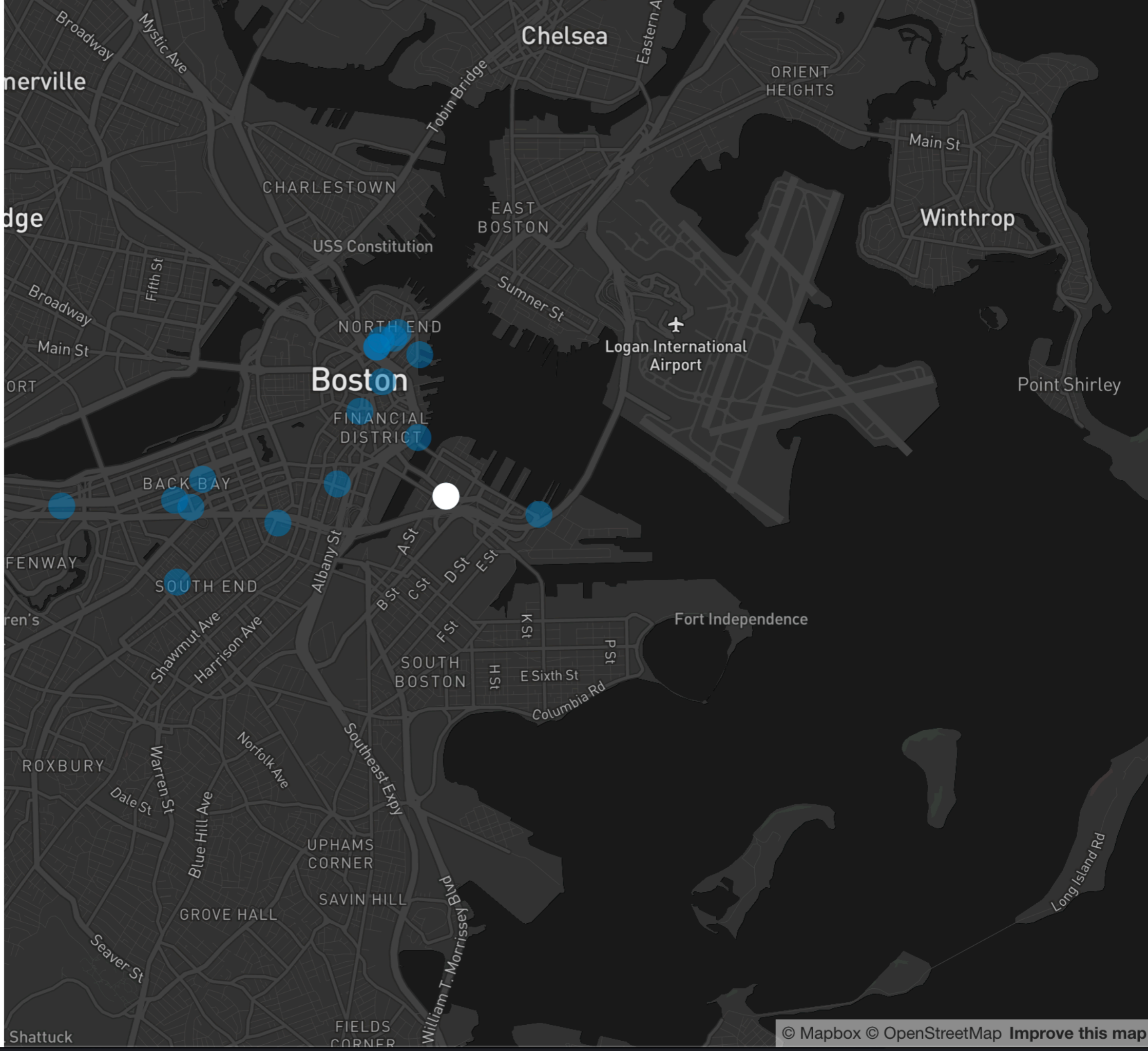
My Rating  
★★★★★ 0

## Yankee Lobster

★★★★★ 4  
1575 Reviews on Yelp

My Rating  
★★★★★ 0

## James Hook & Co







*Call to Adventure*

---

DATE

SCENE

ACT

07/09

2

1



*Call to Adventure*

---

DATE	SCENE	ACT
07/09	2	1



# *Call to Adventure*

---

DATE

SCENE

ACT

07/09

2

1



1. ▲ **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
2. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [1](#) comments
3. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
4. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
5. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
6. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
7. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
8. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
9. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
10. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
11. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
12. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
13. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
14. **Some news (github.com)**  
100000 points by [somedude](#) 3 hours ago | [hide](#) | [100000](#) comments
15. **Some news (github.com)**







got-lobstah.com



# 503 Service Temporarily Unavailable

cloudflare





~~How about that "JAM"~~

I'll lose the bleeps, the sweeps and all those  
creeps



# *Refusal*

---

DATE

SCENE

ACT

07/09

3

1



*Refusal*

---

DATE	SCENE	ACT
07/09	3	1



*Refusal*

---

DATE

SCENE

ACT

07/09

3

1



We're gonna need a bigger server rack

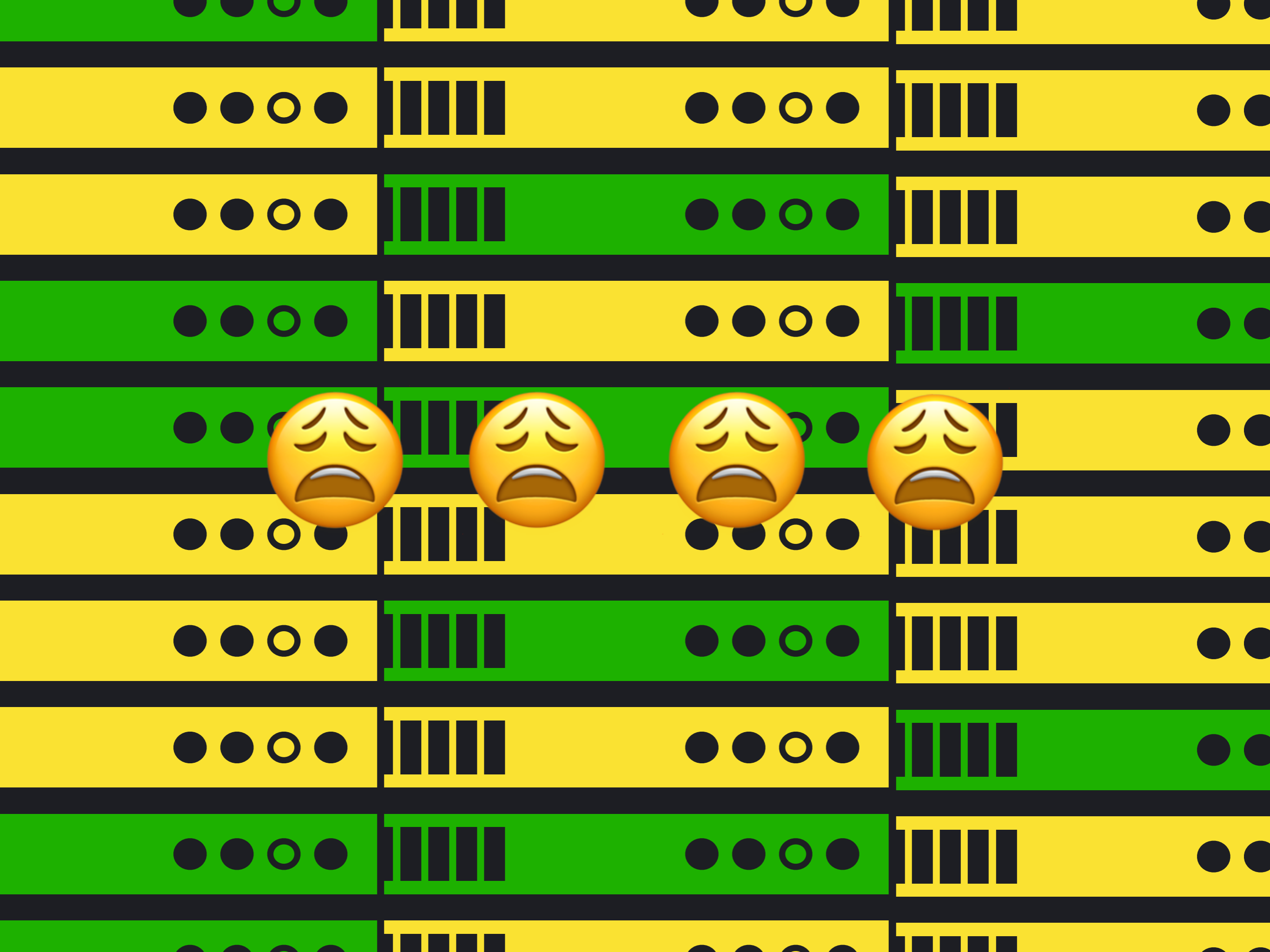


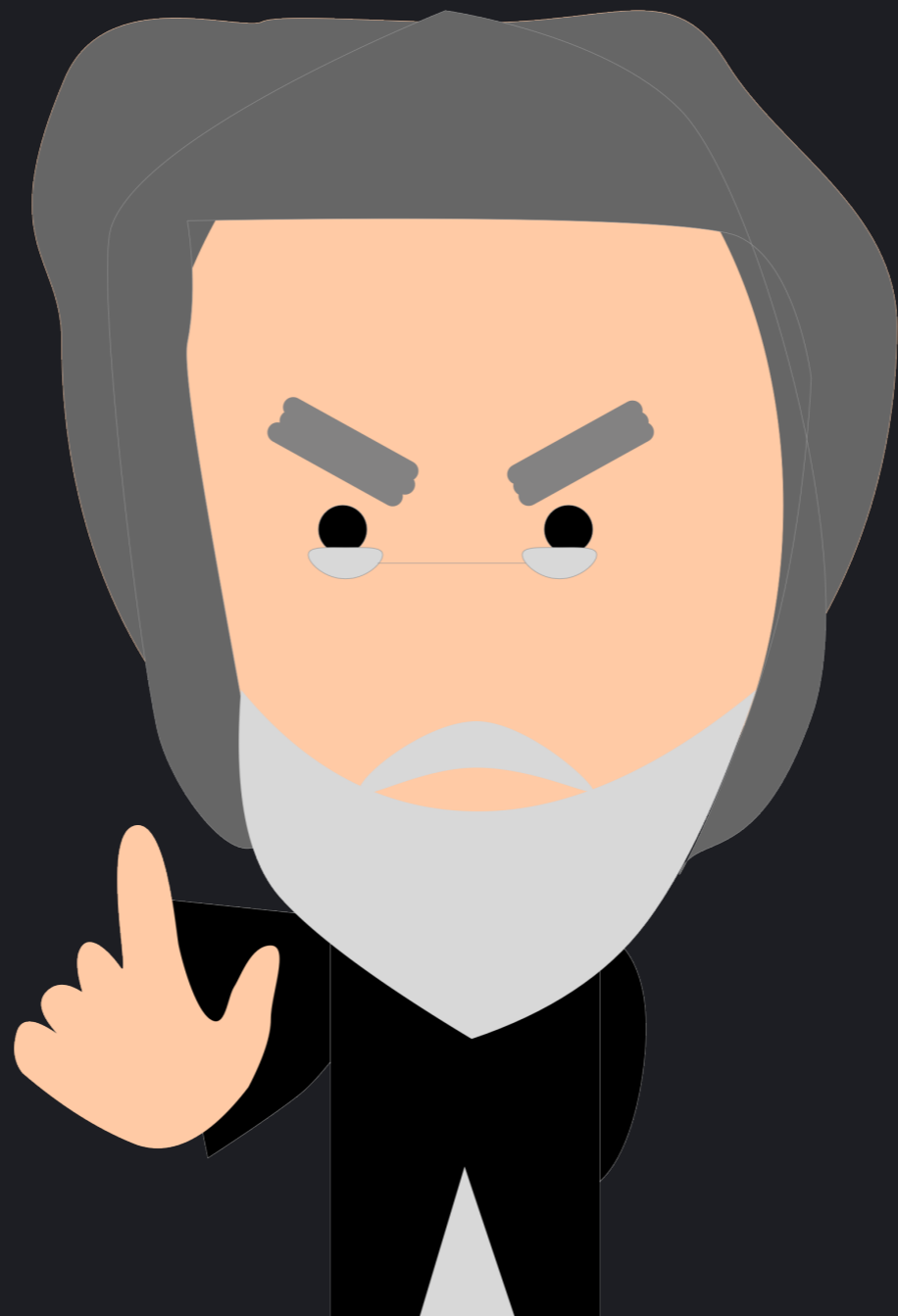


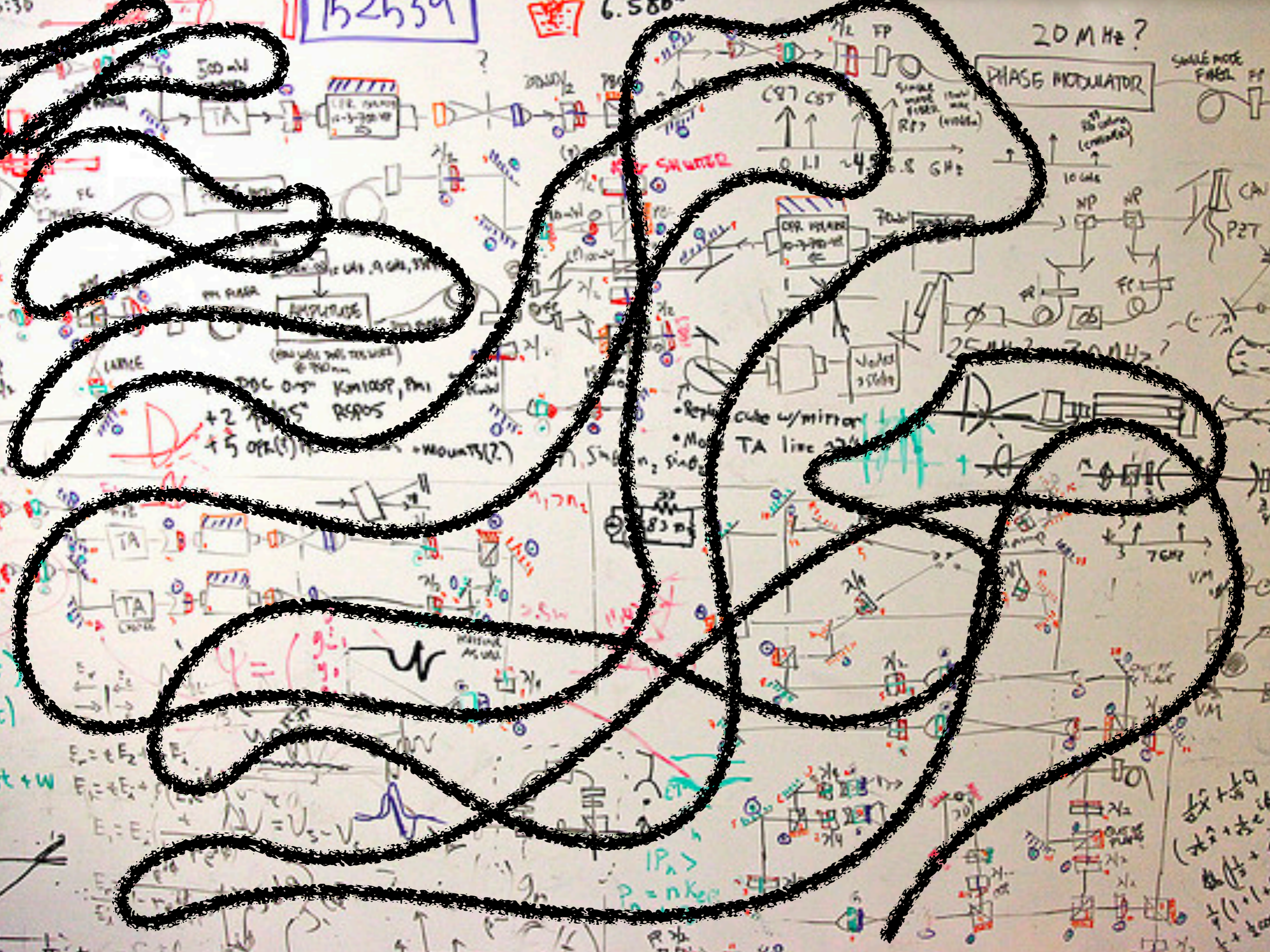












152539

6.588

20 MHz?

PHASE MODULATOR

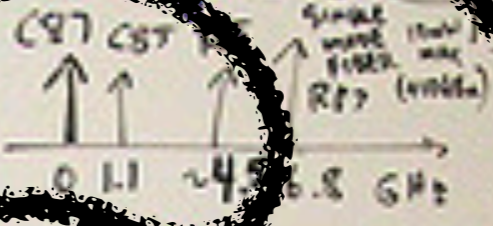
SCALE NOTE  
FIBER

500 mW

TA



SWITCH



10 uW

NP

NP

AMP

AMPLITUDE

DIG 0-5

Km100P, Phi

RESPOS

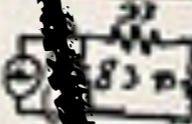
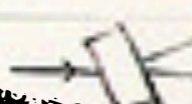
+2

OPK(t)

+MOUNTS(?)

cube w/mirror

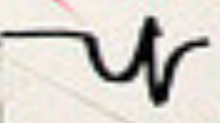
TA line



TA

TA

$$Y = \left( \frac{g_1}{g_2} \right)$$



$$E_1 = E_2$$

$$E_1 = E_2 + \dots$$

$$E_1 = E_2$$

$$V = V_3 - V_4$$

$$IP_n >$$

$$P = nKop$$

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2}$$



*Meeting the Mentor*

---

DATE                      SCENE                      ACT

07/09

4

1



# *Meeting the Mentor*

---

DATE	SCENE	ACT
07/09	4	1





*Meeting the Mentor*

---

DATE                      SCENE                      ACT

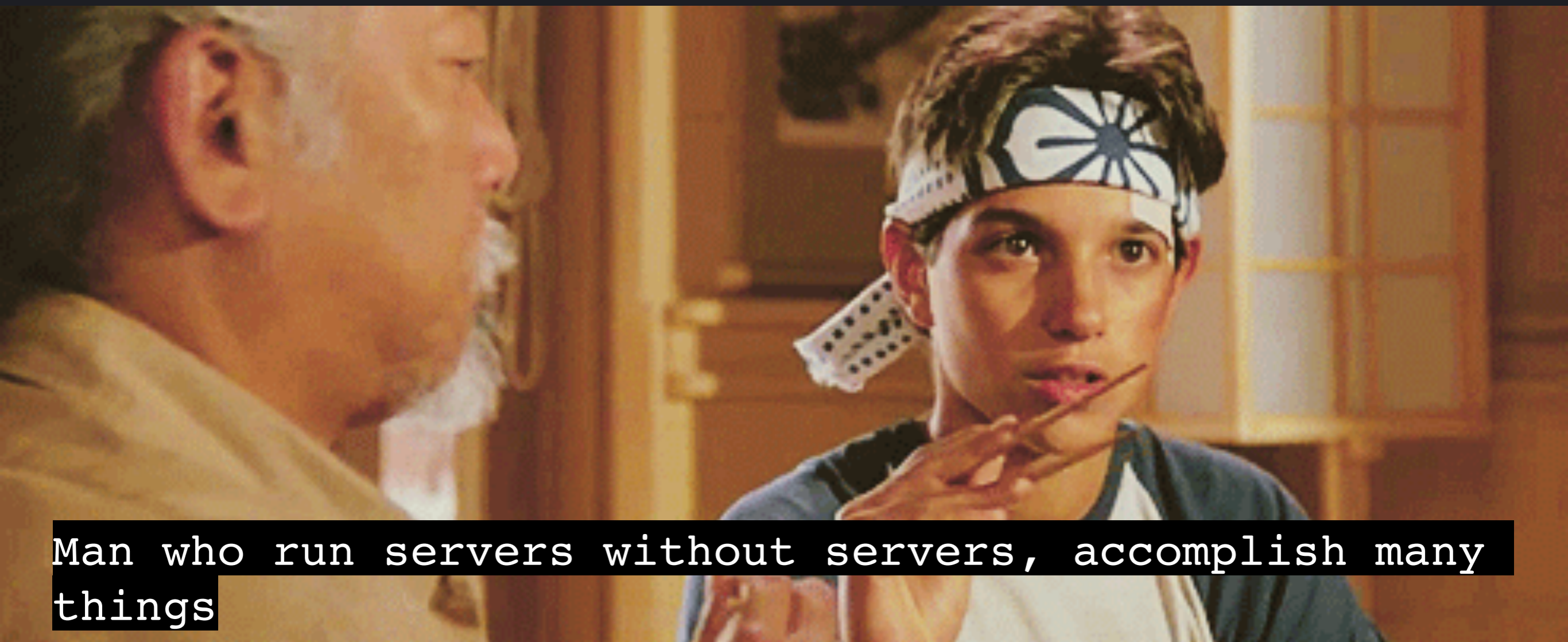
07/09

4

1



**Swami Scale-a-lot**



Man who run servers without servers, accomplish many things

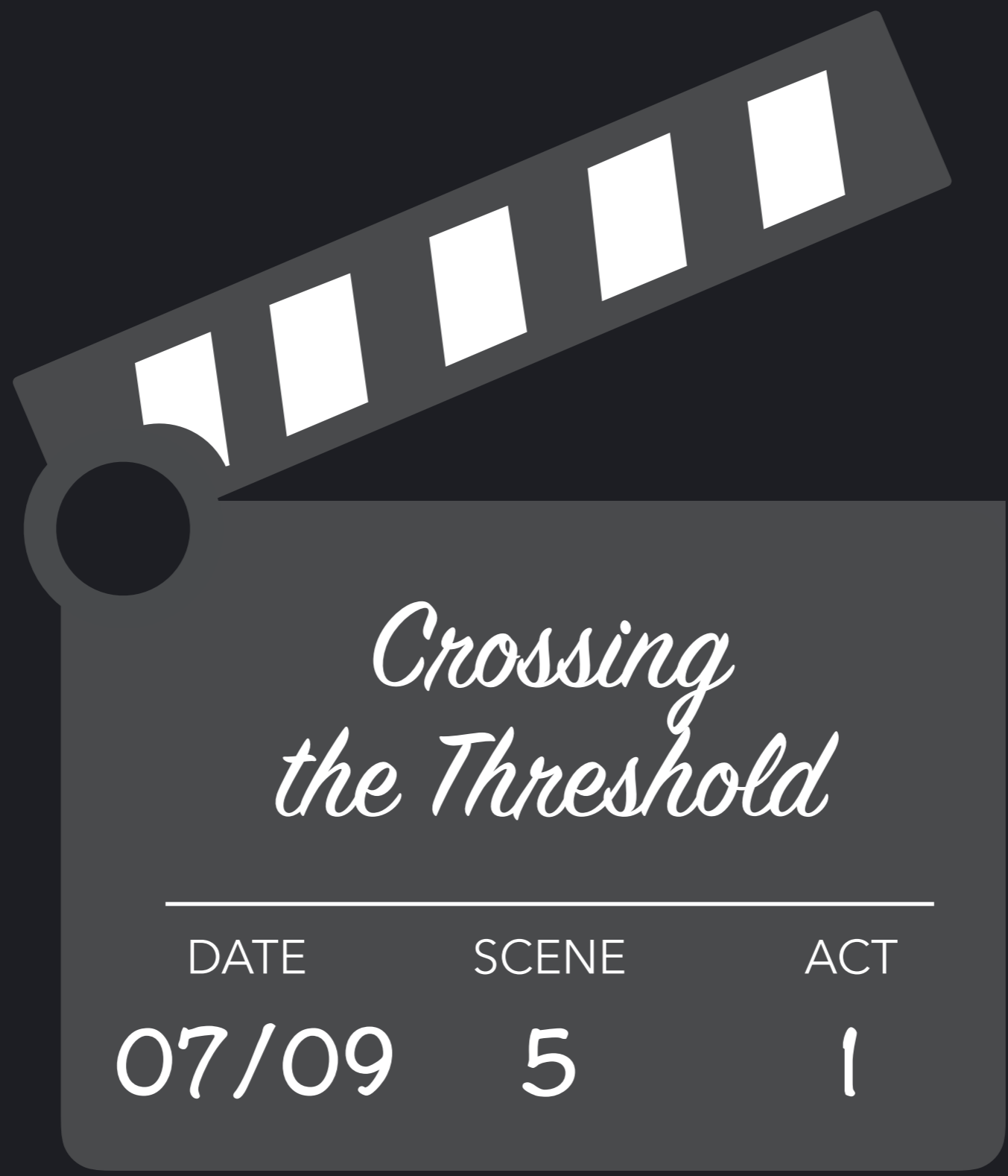


*Crossing  
the Threshold*

---

DATE                  SCENE                  ACT

07/09                  5                  1



*Crossing  
the Threshold*

---

DATE	SCENE	ACT
07/09	5	1



*Crossing  
the Threshold*

---

DATE                  SCENE                  ACT

07/09                  5                  1









Act 2

A silhouette of a person walking away from the viewer down a long, perspective-lined tunnel. The tunnel walls are textured with faint, repeating circular patterns. The lighting is dim, creating a sense of depth and mystery.

# Initiation



*Tests, Allies, Enemies*

---

DATE

SCENE

ACT

07/09

1

2



*Tests, Allies, Enemies*

---

DATE

SCENE

ACT

07/09

1

2



*Tests, Allies, Enemies*

---

DATE

SCENE

ACT

07/09

1

2

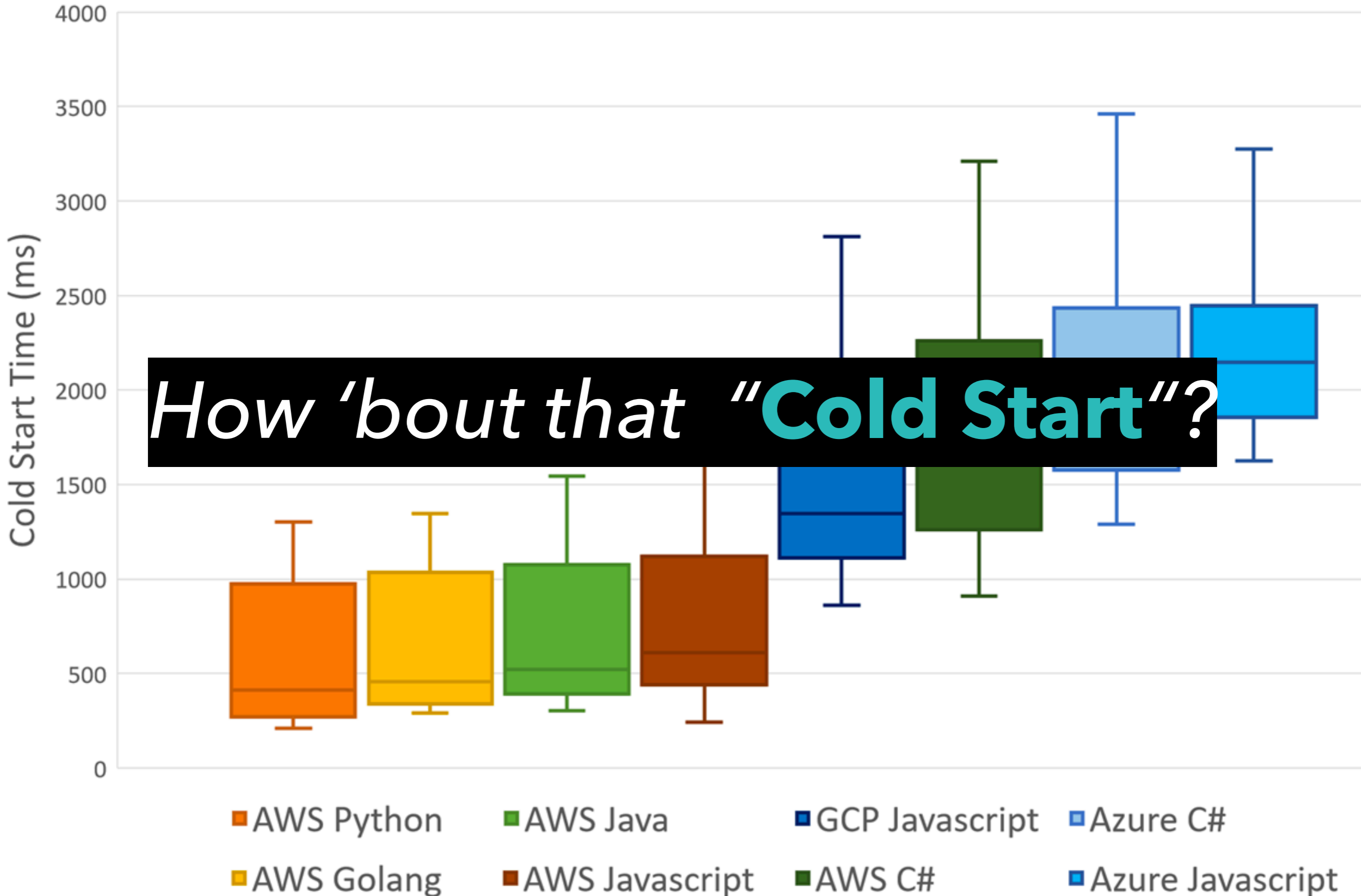
-----  
**THERE  
ARE  
STILL  
SERVERS  
IN  
SERVERLESS**  
-----

( \\_ \_ / ) ||

( • 人 • ) ||

/ づ

# Languages / Cloud Vendors Comparison





**APPROVED**





*Innermost Cave*

---

DATE

SCENE

ACT

07/09

2

2



*Innermost Cave*

---

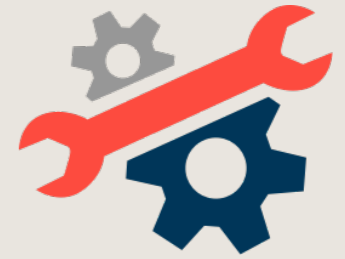
DATE	SCENE	ACT
07/09	2	2



*Innermost Cave*

---

DATE	SCENE	ACT
07/09	2	2





## Luke's Lobster Back Bay

★★★★★ 4  
1578 Reviews on Yelp

My Rating  
★★★★★ 0

## Neptune Oyster

★★★★★ 4.5  
4740 Reviews on Yelp

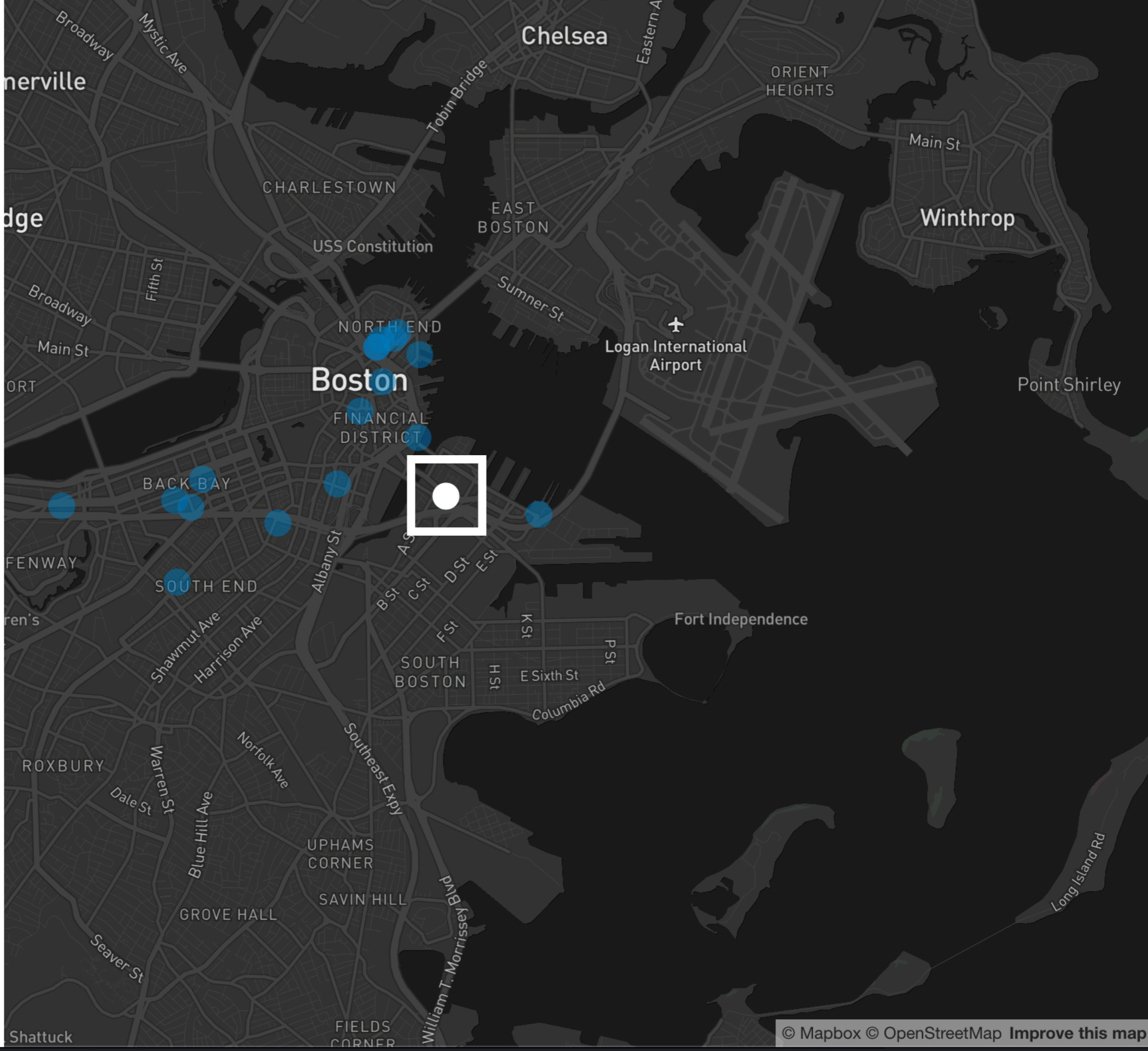
My Rating  
★★★★★ 0

## Yankee Lobster

★★★★★ 4  
1575 Reviews on Yelp

My Rating  
★★★★★ 0

## James Hook & Co



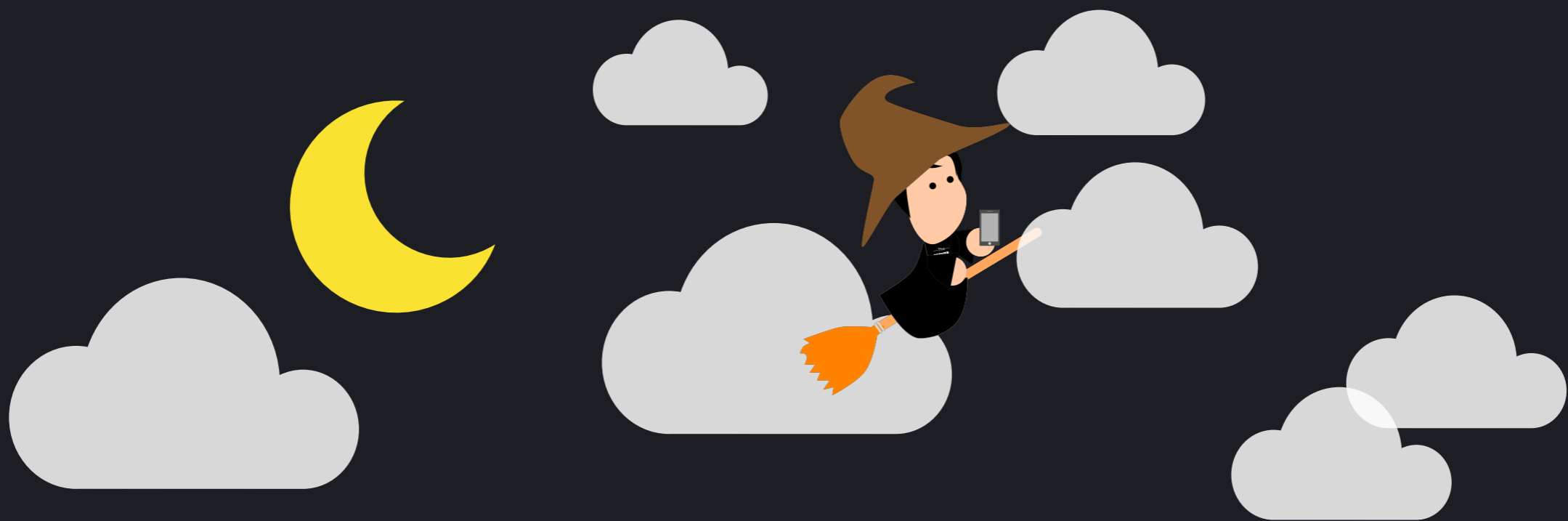
```
1  const express = require('express'),
2     yelp = require('yelp-fusion'),
3     port = 8000,
4     app = express();
5
6  app.use((req, res, next) => {
7    res.header("Access-Control-Allow-Origin", "*");
8    res.header("Access-Control-Allow-Headers", "Origin, X-Requested
9    -with, Content-Type, Accept");
10   next();
11 })
12
13 app.get('/yelpit/:term/:location', (req, res) => {
14   const clientId = CLIENT_ID;
15   const clientSecret = CLIENT_SECRET;
16
17   yelp.accessToken(clientId, clientSecret)
18     .then((response) => {
19     const client = yelp.client(response.jsonBody.access_token);
20     const searchRequest = {
21       term: req.params.term,
22       limit: 50,
23       location: req.params.location
24     }
25     client.search(searchRequest).then(response => {
26     const firstResult = response.jsonBody.businesses;
27     res.send(firstResult)
28   })
29 })
```

```
1  const express = require('express'),
2     yelp = require('yelp-fusion'),
3     port = 8000,
4     app = express();
5
6  app.use((req, res, next) => {
7    res.header("Access-Control-Allow-Origin", "*");
8    res.header("Access-Control-Allow-Headers", "Origin, X-Requested
9    -with, Content-Type, Accept");
10   next();
11 })
12
13 app.get('/yelpit/:term/:location', (req, res) => {
14   const clientId = CLIENT_ID;
15   const clientSecret = CLIENT_SECRET;
16
17   yelp.accessToken(clientId, clientSecret)
18     .then((response) => {
19     const client = yelp.client(response.jsonBody.access_token);
20     const searchRequest = {
21       term: req.params.term,
22       limit: 50,
23       location: req.params.location
24     }
25     client.search(searchRequest).then(response => {
26       const firstResult = response.jsonBody.businesses;
27       res.send(firstResult)
28     })
29   })
30 }
```



```
10  })
11
12
13  app.get('/yelpit/:term/:location', (req, res) => {
14    const clientId = CLIENT_ID;
15    const clientSecret = CLIENT_SECRET;
16
17    yelp.accessToken(clientId, clientSecret)
18      .then((response) => {
19        const client = yelp.client(response.jsonBody.access_token);
20        const searchRequest = {
21          term: req.params.term,
22          limit: 50,
23          location: req.params.location
24        }
25        client.search(searchRequest).then(response => {
26          const firstResult = response.jsonBody.businesses;
27          res.send(firstResult)
28        })
29      })
30  })
31
32  app.listen(port, 'localhost', function(err) {
33    if(err) {console.log(err)}
34    console.info(`Listening on port ${port}`)
35  });
36
```

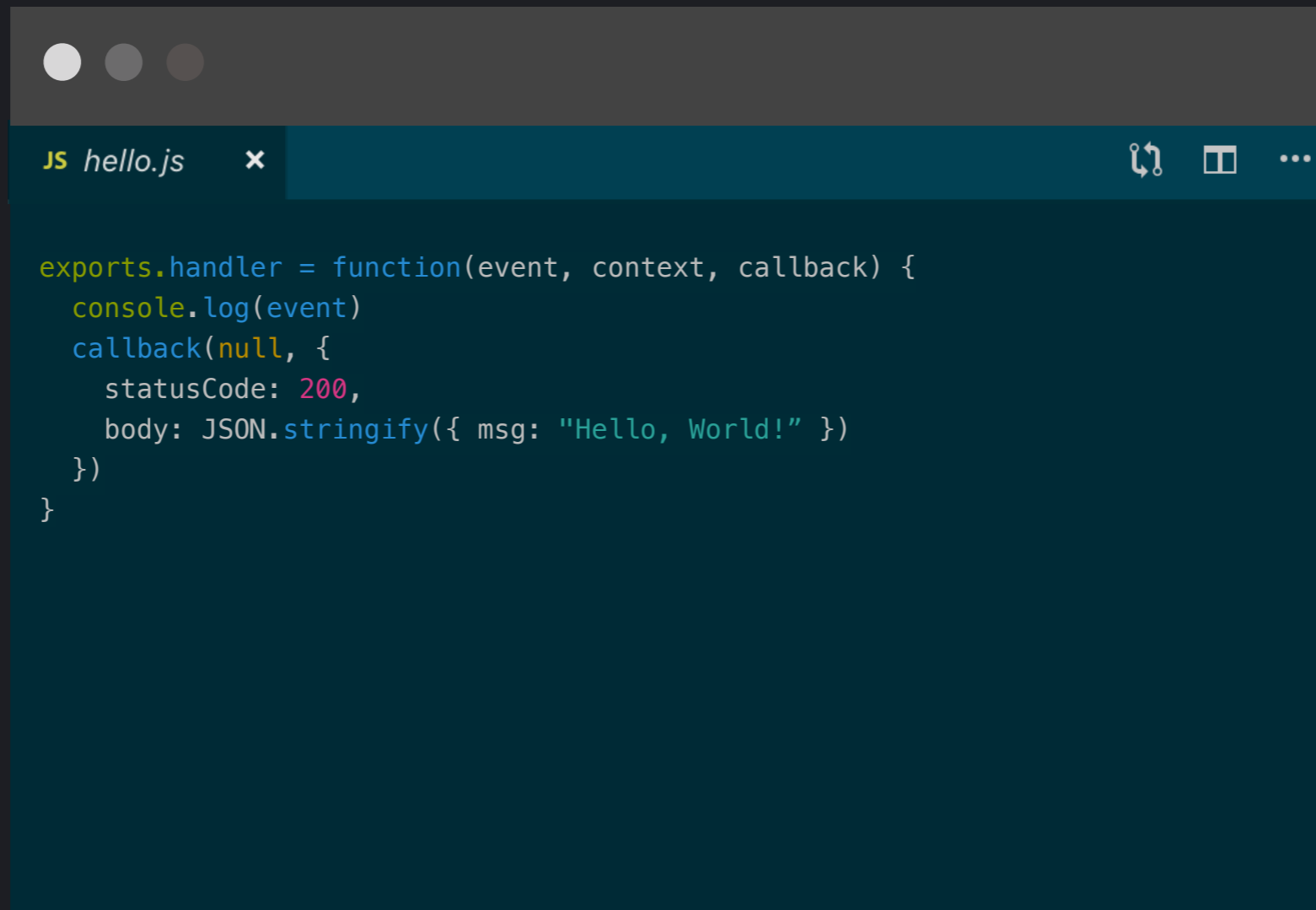
```
1  module.exports = {
2    devServer: {
3      proxy: {
4        '^/yelpit/*': {
5          target: 'http://localhost:8000/yelpit',
6          secure: false
7        }
8      }
9    }
10  }
11
```



# The Chilling Adventures of Functions as a Service







The image shows a code editor window with a dark teal background. The window title bar at the top contains three window control buttons (red, yellow, green) on the left and three icons (refresh, window, and menu) on the right. The tab title is 'JS hello.js' with a close button 'x'. The code is as follows:

```
exports.handler = function(event, context, callback) {  
  console.log(event)  
  callback(null, {  
    statusCode: 200,  
    body: JSON.stringify({ msg: "Hello, World!" })  
  })  
}
```

**functions/hello.js**

```
[build]
  command = "yarn build"
  functions = "functions"
  publish = "dist"
```

**netlify.toml**



## Functions

Functions Free

5 Lambda functions actively running in production.

[Learn more about Functions in the docs](#) ↗

⚙ Settings and usage

🔍 Search by branch or Deploy Preview

**handle-signup.js**

Created on Mar 14 (19 days ago)



**hello.js**

Created on Mar 14 (19 days ago)



**identity-login.js**

Created on Mar 14 (19 days ago)



**lock-site.js**

Created on Mar 14 (19 days ago)



**super-special-function.js**



`/.netlify/functions/yelp-it`



```
1  const yelp = require("yelp-fusion");
2
3  exports.handler = async event => {
4    const client = yelp.client(process.env.YELP_API_KEY);
5    const { term, location } = event.queryStringParameters;
6    const searchRequest = {
7      term: term,
8      location: location
9    };
10   var geojsonify = async res => {
11     let features = [];
12     res.map(item => {
13       features.push({
14         type: "Feature",
15         geometry: {
16           type: "Point",
17           coordinates: [item.coordinates.longitude, item.coordinates.l
18         ],
19         properties: {
20           ...item
21         }
22       });
23     });
24   });
25   return {
26     type: "FeatureCollection",
27     features
28   };
29 };
30
```

```
16         coordinates: [item.coordinates.longitude, item.coordinates.latitude],
17     },
18     properties: {
19         ...item
20     }
21 });
22 });
23 return {
24     type: "FeatureCollection",
25     features
26 };
27 };
28 try {
29     let res = await client.search(searchRequest);
30     let geojsonified = await geojsonify(res.jsonBody.businesses);
31     return {
32         statusCode: 200,
33         body: JSON.stringify({
34             results: geojsonified
35         })
36     };
37 } catch (err) {
38     return { statusCode: 500, body: err.toString() };
39 }
40 };
41 };
42
43
44
```

```
1 <template>
2   <div class="home">
3     <div id="sidebar" v-if="dataLoaded">
4       <Slider
5         :yelpData="yelpData.features"
6       />
7     </div>
8   </div>
9 </template>
10
11 <script>
12 import { value, onCreated } from "vue-function-api";
13
14 export default {
15   name: "home",
16   setup() {
17     const yelpData = value(null);
18     const getYelpData = () => {
19       axios
20         .get("/.netlify/functions/yelp-it", {
21           params: {
22             location: "boston, ma",
23             term: "lobster"
24           }
25         })
26         .then(async res => {
27           yelpData.value = res.data.results;
28         })
29     }
```

```
14 export default {
15   name: "home",
16   setup() {
17     const yelpData = value(null);
18     const getYelpData = () => {
19       axios
20         .get("/.netlify/functions/yelp-it", {
21           params: {
22             location: "boston, ma",
23             term: "lobster"
24           }
25         })
26         .then(async res => {
27           yelpData.value = res.data.results;
28         })
29         .catch(err => {
30           console.log(err);
31         });
32     };
33     onCreate(async () => {
34       getYelpData();
35     });
36   }
37 }
38 </script>
```

```
1 <template>
2   <div class="home">
3     <div id="sidebar" v-if="dataLoaded">
4       <Slider
5         :yelpData="yelpData.features"
6       />
7     </div>
8   </div>
9 </template>
10 <script>
11 import { value, onCreated } from "vue-function-api";
12
13 export default {
14   name: "home",
15   setup() {
16     const yelpData = value(null);
17     const getYelpData = () => {
18       axios
19         .get("/.netlify/functions/yelp-it", {
20           params: {
21             location: "boston, ma",
22             term: "lobster"
23           }
24         })
25         .then(async res => {
26           yelpData.value = res.data.results;
27         })

```



### Luke's Lobster Back Bay

★★★★☆ 4  
1578 Reviews on Yelp

My Rating  
☆☆☆☆☆ 0

### Neptune Oyster

★★★★☆ 4.5  
4740 Reviews on Yelp

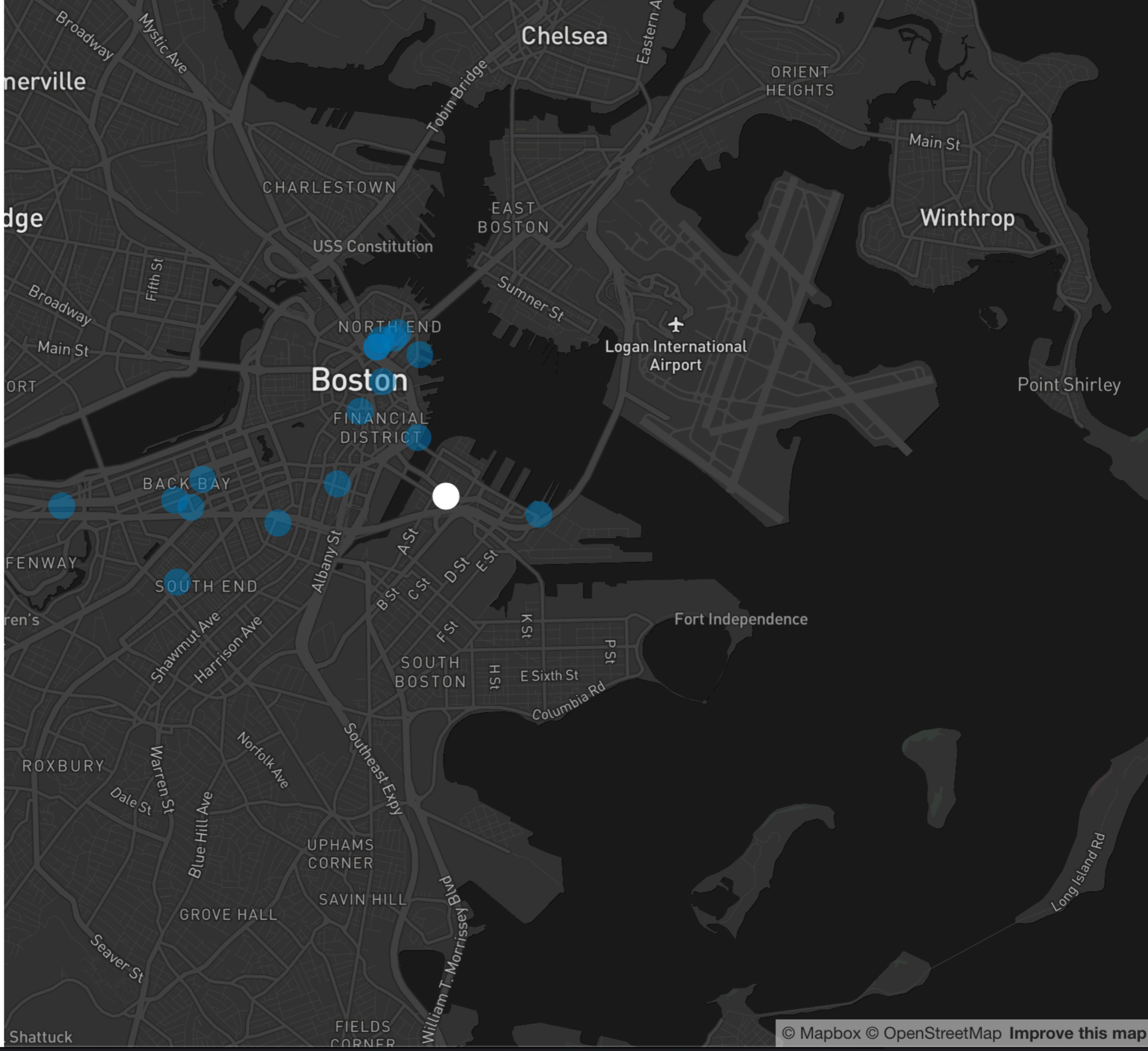
My Rating  
☆☆☆☆☆ 0

### Yankee Lobster

★★★★☆ 4  
1575 Reviews on Yelp

My Rating  
☆☆☆☆☆ 0

### James Hook & Co







```
netlify addons:create fauna
```

```
1 const faunadb = require("faunadb");
2 const q = faunadb.query;
3
4 export default async function useFauna() {
5   const createClient = async () => {
6     const secret = await axios.get("/.netlify/functions/fauna-user")
7     const { faunaSecret } = secret.data;
8     const client = new faunadb.Client({ secret: faunaSecret });
9     return client;
10  };
11
12  const client = await createClient();
13
14  const setRating = ref => async title => {
15    try {
16      if (ref === null) {
17        client
18          .query(
19            q.Create(q.Class("ratings"), {
20              data: title
21            })
22          )
23          .then(ret => console.log(ret))
24          .catch(err => console.log(err));
25      }
26      const listicle = await client
27        query(
```

```
12 const client = await createClient();
13
14 const setRating = ref => async title => {
15   try {
16     if (ref === null) {
17       client
18         .query(
19           q.Create(q.Class("ratings"), {
20             data: title
21           })
22         )
23         .then(ret => console.log(ret))
24         .catch(err => console.log(err));
25     } else {
26       client
27         .query(
28           q.Update(q.Ref(q.Class("ratings"), ref), {
29             data: title
30           })
31         )
32         .then(ret => console.log(ret));
33     }
34     return listicle;
35   } catch (err) {
36     console.log(err);
37   }
38 };
39
```

```
37     console.log(err);
38   }
39 };
40
41 const getRatings = async () => {
42   try {
43     const res = await client.query(
44       q.Map(q.Paginate(q.Match(q.Ref("indexes/all_ratings"))),
45         ref =>
46           q.Get(ref)
47         )
48     );
49     return res.data;
50   } catch (err) {
51     console.log(err);
52   }
53 };
54
55 return {
56   client,
57   getRatings,
58   setRating
59 };
60 }
61
62
```



*Ordeal*

---

DATE

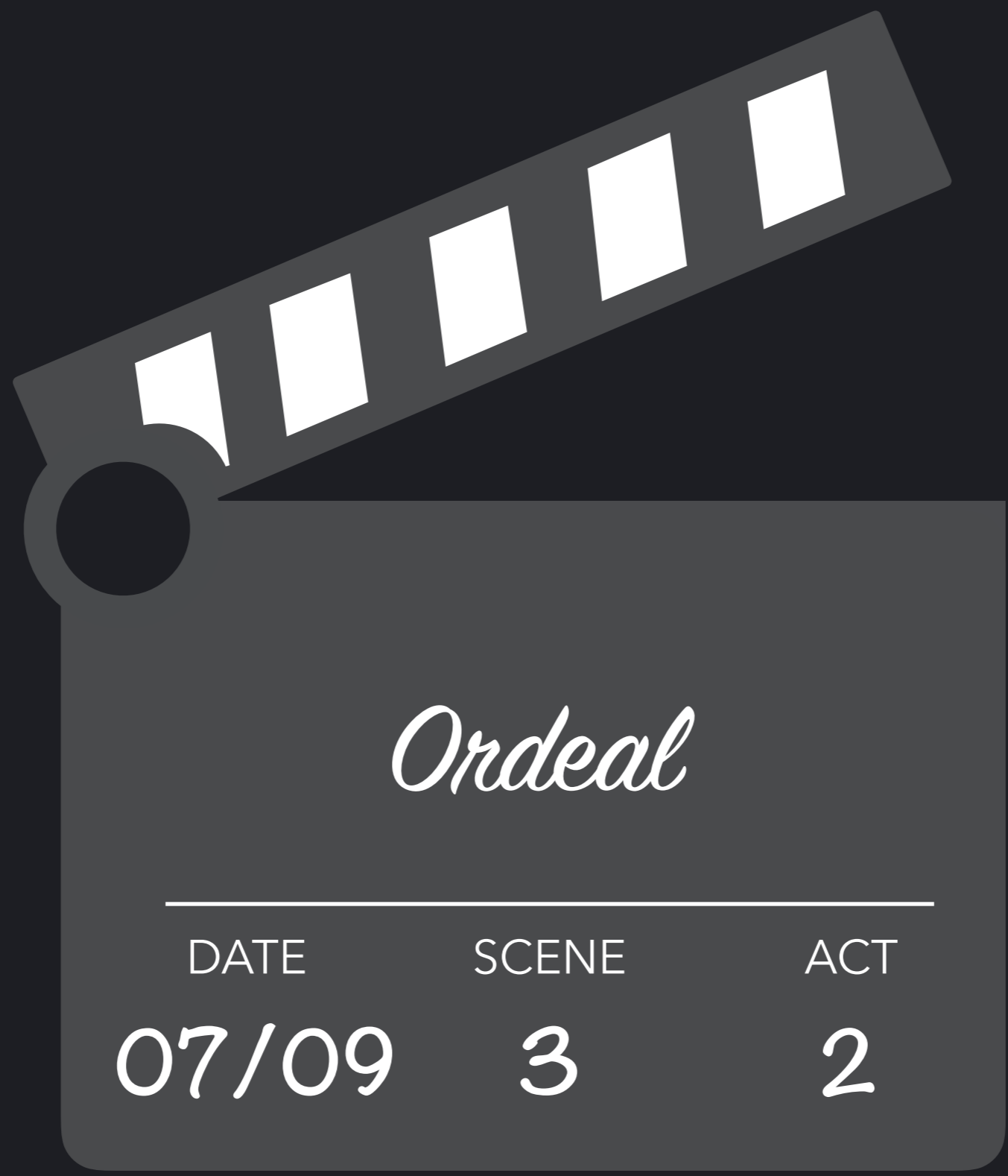
SCENE

ACT

07/09

3

2



*Ordeal*

---

DATE	SCENE	ACT
07/09	3	2



*Ordeal*

---

DATE

SCENE

ACT

07/09

3

2





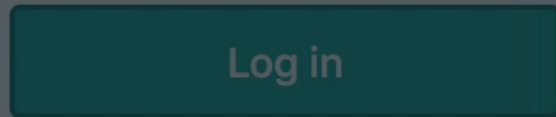
Auth as a **Service**

Netlify **Identity**

Netlify Identity

# Auth v

Single button example:



Click  
the w

✕

Sign up

Log in

---

✉ divya@netlify.com

divya@netlify.com

hello@shortdiv.com

divified@gmail.com

shortdiv

name@email.com

jane@example.com

Clear localhost URL

---

OR

G

Continue with Google





## you-auth-to-know

• <https://you-auth-to-know.netlify.com>

Deploys from [GitHub](#). Last published on Mar 20.



⚙ Site settings

⚙ Domain settings

### 🔗 Production deploys >

Production: master@7f7d453 **PUBLISHED** >

Mar 20: typo

Production: master@1839c4d **FAILED** >

Mar 20: fixes

Production: master@58ddec1 >

Mar 20: add dashboard page

Production: master@cc17049 >

Mar 20: add page and asset

### 🔗 Deploy Previews

#### Auto-preview your PR or branch

Every time you open a pull request, or push new changes to a branch, Netlify automatically builds a preview with a unique URL. Like a staging environment for every PR or branch, previews are perfect for testing and collaboration.

[Learn more about Deploy Previews](#) ➔



```
1 const netlifyIdentity = require("netlify-identity-widget");
```

```
2  
3 export default function useAuth() {  
4   const netlifyIdentityInit = () => {  
5     return netlifyIdentity.init({  
6       container: ".account-profile"  
7     });  
8   };  
9
```

```
10   netlifyIdentityInit();
```

```
11  
12   return {  
13     netlifyIdentity  
14   };  
15 };  
16 };
```

`/.netlify/functions/handle-signin`

```
1 'use strict';
2 const faunadb = require('faunadb');
3 const generator = require('generate-password');
4
5 /* configure faunaDB Client with our secret */
6 const q = faunadb.query;
7 const client = new faunadb.Client({
8   secret: process.env.FAUNADB_SERVER_SECRET
9 });
10
11 /* create a user in FaunaDB that can connect from the browser */
12 function createUser(userData, password) {
13   return client.query(
14     q.Create(q.Class('users'), {
15       credentials: {
16         password: password
17       },
18       data: {
19         id: userData.id,
20         user_metadata: userData.user_metadata
21       }
22     })
23   );
24 }
25
26 function obtainToken(user, password) {
27   return client.query(q.Login(q.Select('ref', user), { password }));
28 }
29
```

```
29 }
30 }
31 export function handler(event, context, callback) {
32   var payload = JSON.parse(event.body);
33   var userData = payload.user;
34
35   const password = generator.generate({
36     length: 10,
37     numbers: true
38   });
39
40   createUser(userData, password)
41     .then(user => obtainToken(user, password))
42     .then(key =>
43       callback(null, {
44         statusCode: 200,
45         body: JSON.stringify({
46           app_metadata: {
47             faunadb_token: key.secret
48           }
49         })
50     })
51   )
52   .catch(e => {
53     console.error(e);
54     callback(null, {
55       statusCode: 500,
56       body: JSON.stringify({
57         error: e
58
```





# *Reward*

---

DATE

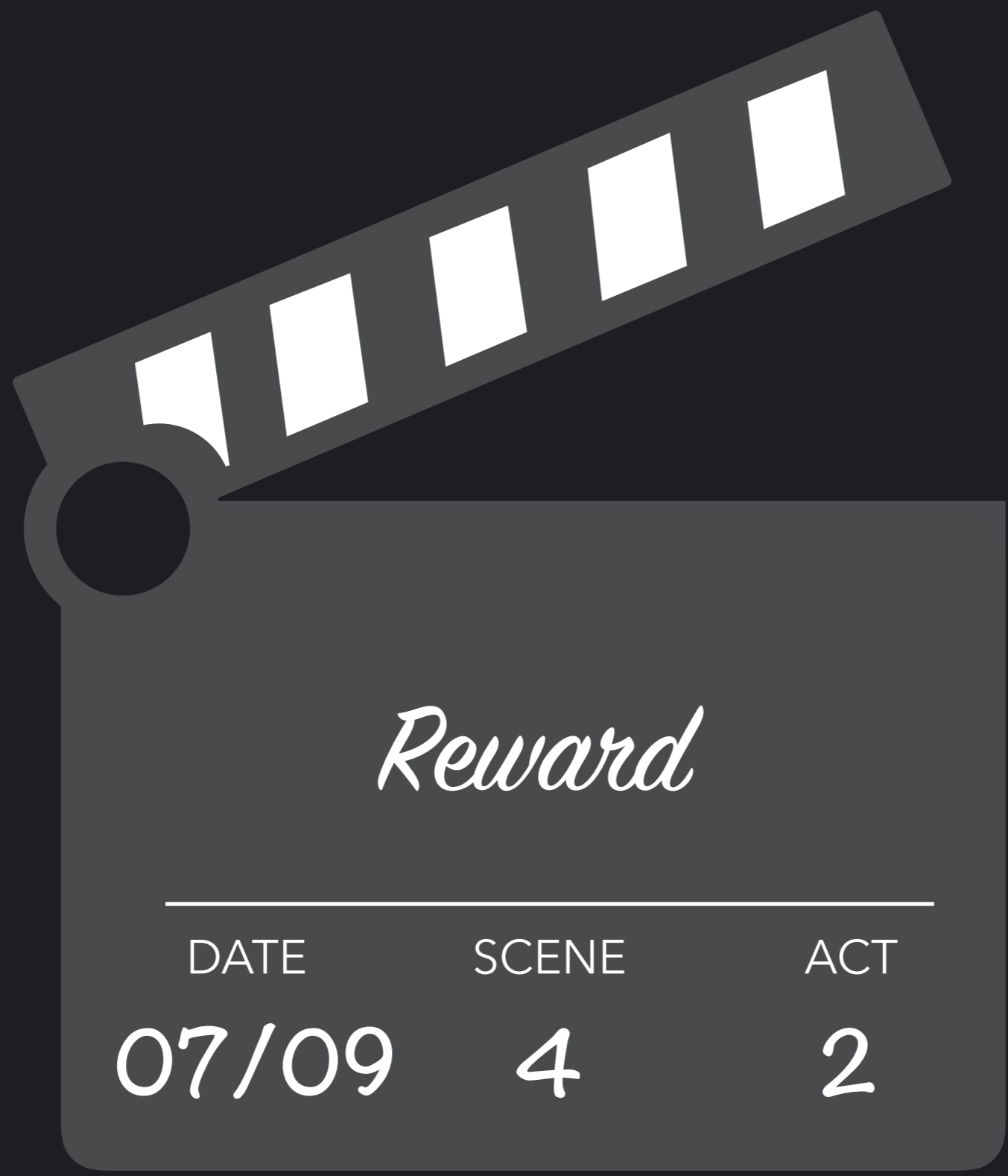
SCENE

ACT

07/09

4

2



*Reward*

---

DATE	SCENE	ACT
07/09	4	2



# *Reward*

---

DATE

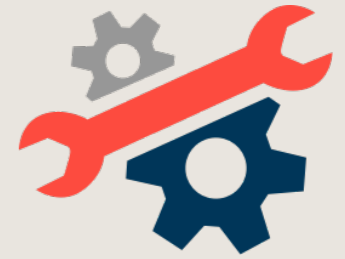
SCENE

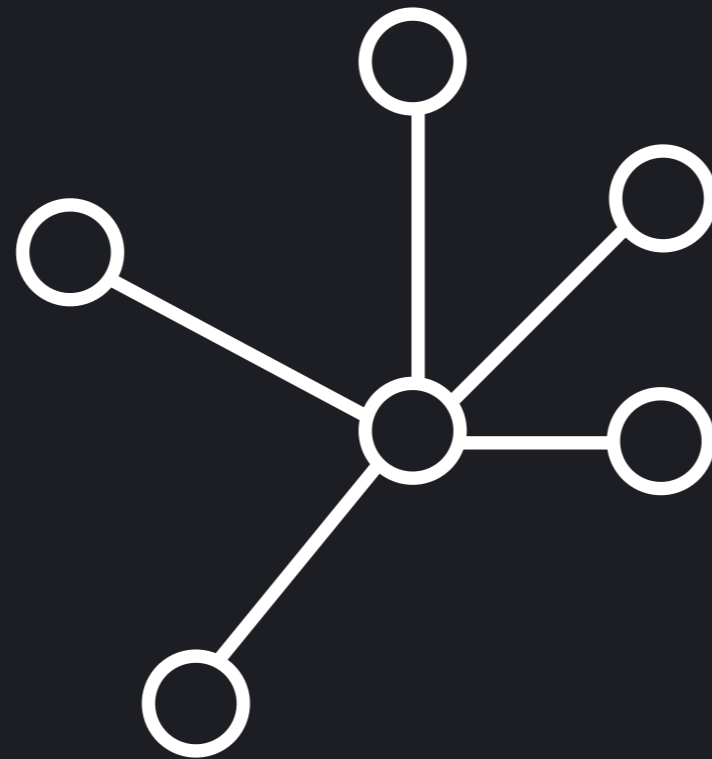
ACT

07/09

4

2





**JAMstack ecosystem**

Form Handling

Search

Push Notifications

Authentication

Database

Static Hosting

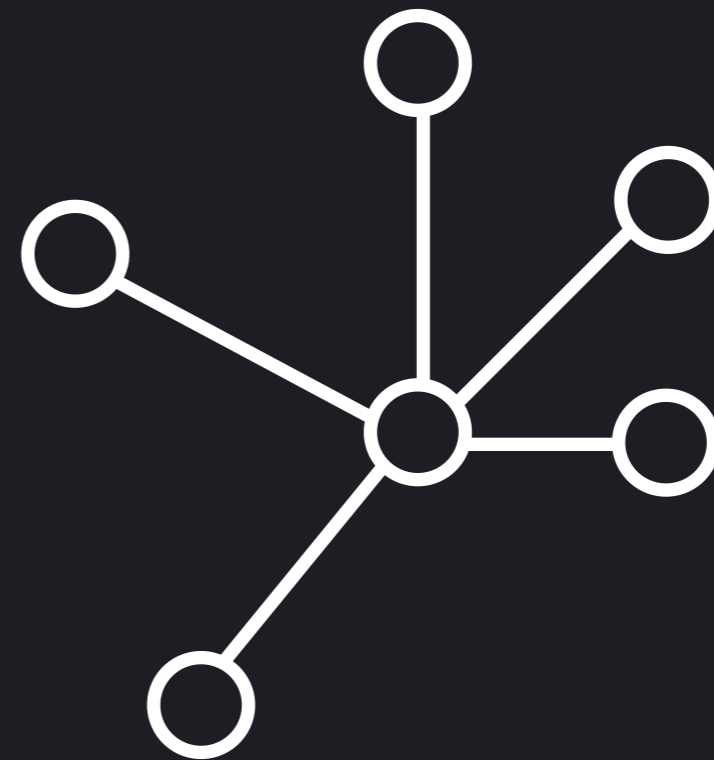
Payments

Media Storage

Headless CMS

Monitoring

FAAS Providers



**JAMstack**  
**ecosystem**



secure



fast



flexible



scalable

Act 3

# Return







*The Road Back*

---

DATE	SCENE	ACT
07/09	1	3



*The Road Back*

---

DATE	SCENE	ACT
07/09	1	3



*The Road Back*

---

DATE

SCENE

ACT

07/09

1

3



There is no cloud, just someone else's server



*Resurrection/  
Return with Elixir*

---

DATE

SCENE

ACT

07/09

2

3



*Resurrection/  
Return with Elixir*

---

DATE	SCENE	ACT
07/09	2	3



*Resurrection/  
Return with Elixir*

---

DATE

SCENE

ACT

07/09

2

3



The Power of  
**SERVERLESS**  
for Front-End Developers



What?



Services



Ideas



Resources

**S**erverless. It's kind of a phenomenon. Let's get one thing out of the way: it still involves servers, so that word *serverless* might feel a bit disingenuous. It's actually a new way to pay for and work with servers that, in many cases, is *cheaper* and *easier* than buying and managing your own servers.

Web developers of all kinds can take advantage of it, but **this site is particularly interested in how it relates to and helps front-end web developers**. Here's one way to think about it: you can take your front-end skills and do things that typically only a back-end can do. You can write a JavaScript

[thepowerofserverless.info](http://thepowerofserverless.info)





**Fullstack**



**JAMstack**



all the things  
~~Radar~~ about to be jammed.



**SERVERLESS**  
4 Lyfe

[gotlobstah.netlify.com](http://gotlobstah.netlify.com)



[github.com/shortdiv/got-lobstah](https://github.com/shortdiv/got-lobstah)