

SERVERLESS SERVER-SIDE  
RENDERING



Hello 

I work with...

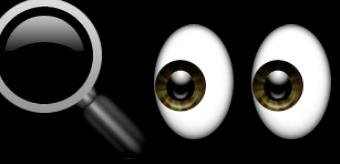
- React
- Typescript
- Cloud things on AWS
- Node.js
- Terraform

**THIS IS ME WAITING**

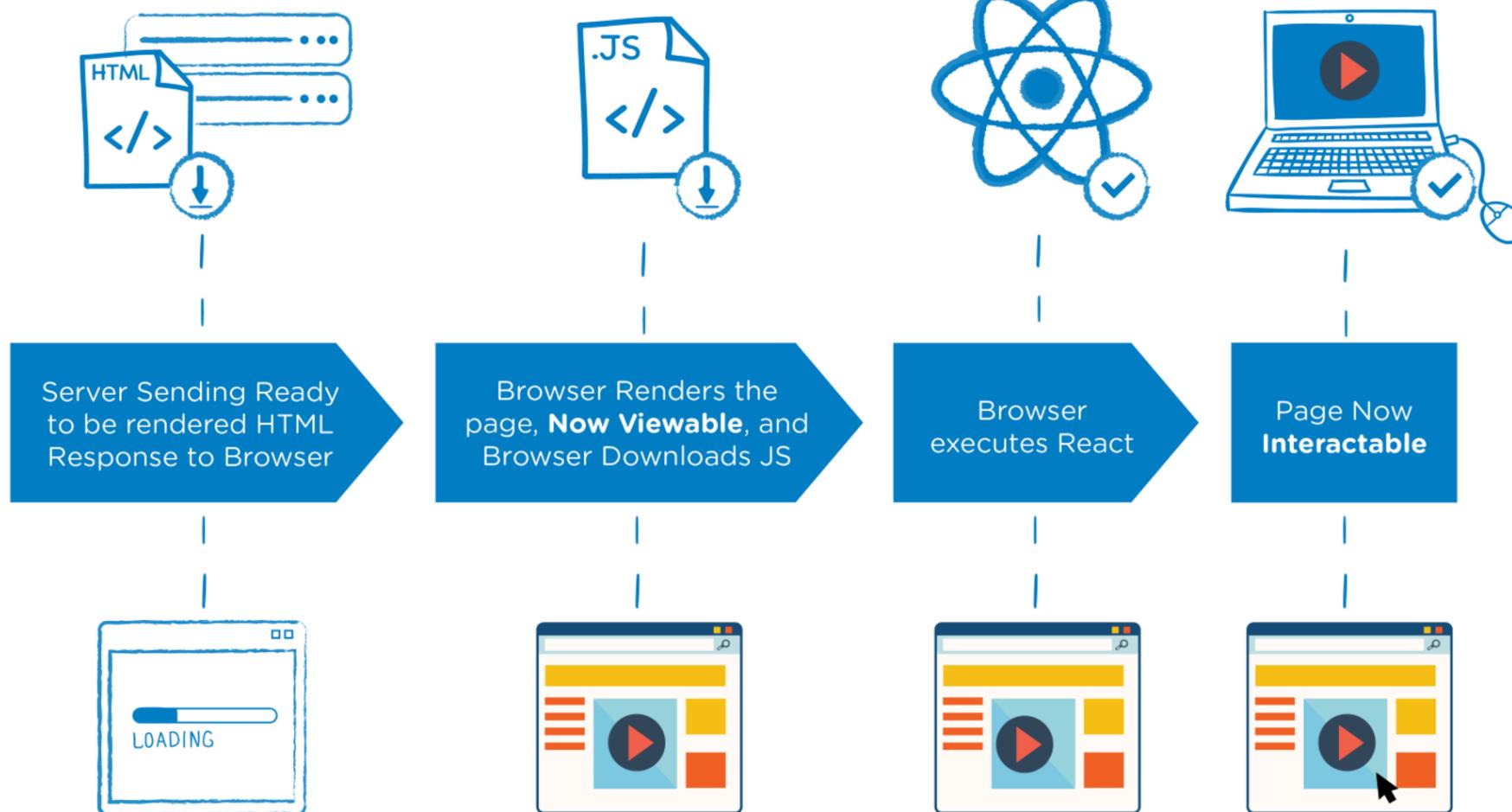
**FOR YOUR SLOW WEBSITE TO  
LOAD**

PERFORMANCE ⚡

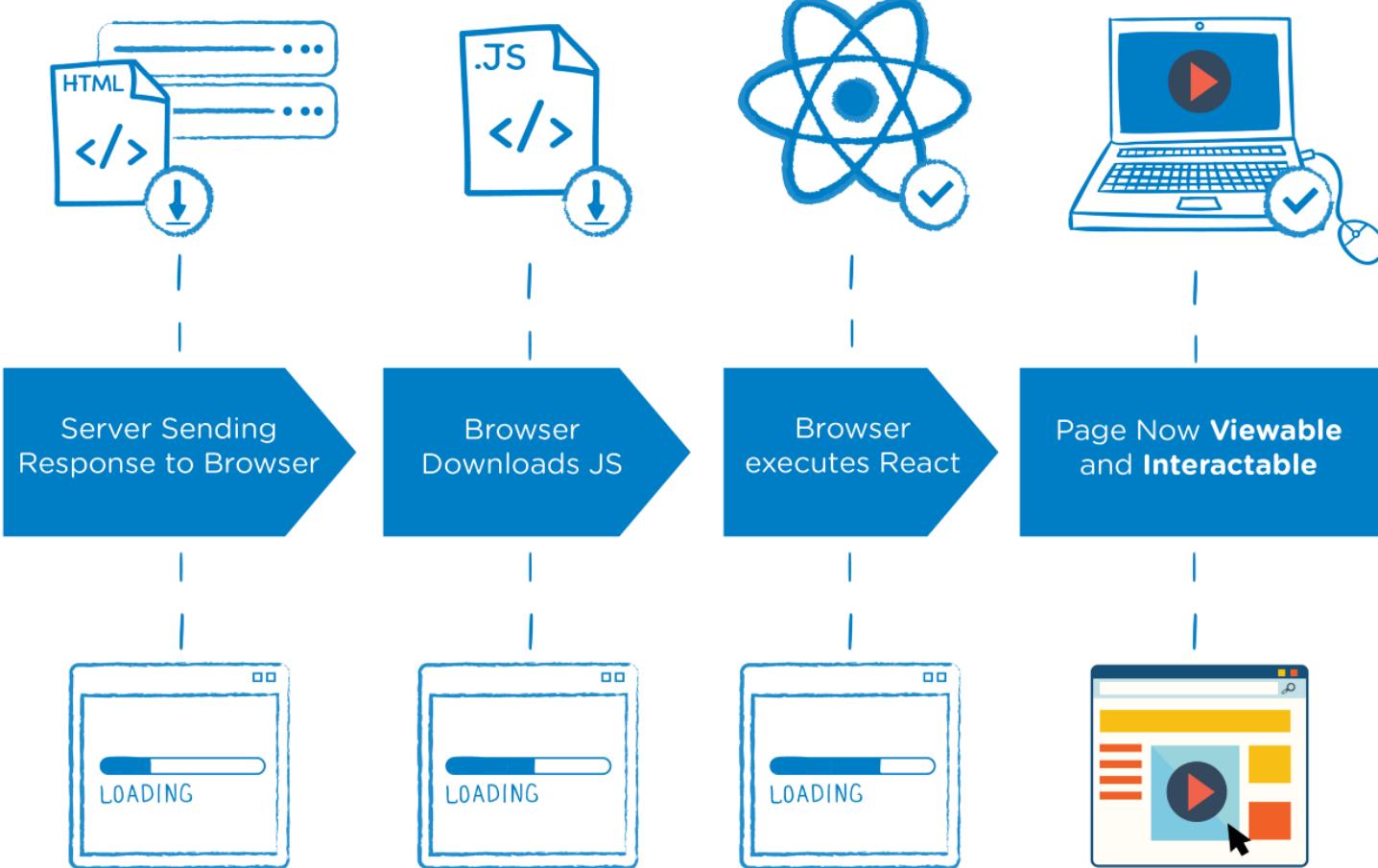
USER EXPERIENCE ✨

SEO 

# SSR



# CSR



# SERVER-SIDE RENDERING

PROS:

SEO

Performance

CONS:

Added complexity

TTFB slower than client-side rendering

```
1 import * as e from 'express';
2 import { Application } from 'express';
3 import * as ServerlessExpress from 'aws-serverless-express';
4 import { MarkupHandler } from './src/handlers/MarkupHandler';
5
6 let express: any = (<any>e).default || e;
7 const app: Application = express()
8
9 app.use('/', MarkupHandler);
10
```

```
1 import { createElement } from 'react';
2 import { renderToString } from 'react-dom/server';
3 import HelloWorld from '../components/HelloWorld';
4 import { Router, Request, Response, NextFunction } from 'express';
5 const MarkupHandler = Router();
6
7 MarkupHandler.get('/', function(request: Request, response: Response, next: NextFunction) {
8     const html = renderToString(createElement(HelloWorld as any));
9     request = request;
10    response.send(html);
11    next();
12 });
13
14 export { MarkupHandler };
```

```
1 import * as React from 'react';
2
3 interface HelloWorldState {
4   buttonClicked: boolean;
5 }
6
7 class HelloWorld extends React.Component<void, HelloWorldState> {
8   constructor(props: any) {
9     super(props);
10    this.state = {
11      buttonClicked: false
12    };
13  }
14
15  render() {
16    const backgroundStyles = {
17      backgroundColor: 'black',
18      height: '100%',
19      width: '100%'
20    };
21    const textStyles = {
22      color: 'white',
23      textAlign: 'center',
24      paddingTop: '25%',
25      paddingBottom: '25%'
26    };
27
28    return (
29      <div style={backgroundStyles}>
30        <h1 style={textStyles}>hello boston!</h1>
31      </div>
32    );
33  }
34}
35
36 export default HelloWorld;
```

# LAMBDA



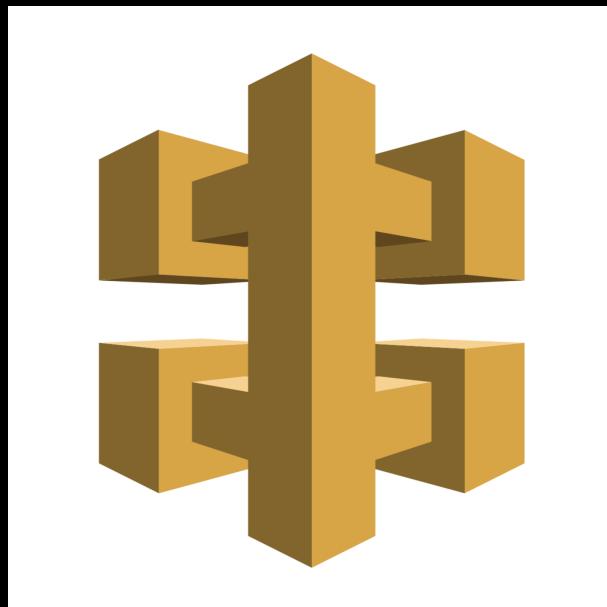
## PROS:

- On-demand execution
- Lower cost than an instance (in some cases)
- Hassle-free scalability

## CONS:

- Initial load time may be slow, but subsequent invocations will be faster
- Stage environments

# API GATEWAY



## API

- Endpoints
- Methods
- Event Object
- Proxy to Lambda

# The Recipe

1. Create zip file of React app
2. Upload zip file to Lambda
3. Create API using API Gateway
4. Create an endpoint in the API
5. Create a GET method for that endpoint
6. Go back to Lambda and choose your new API Gateway as trigger
7. Stage the API
8. Make a request to that endpoint
9. Woohoo!



*SERVERLESS*

The word "SERVERLESS" is displayed in a large, white, sans-serif font. The letter "S" is capitalized and italicized. A bright yellow lightning bolt graphic is positioned between the "V" and "L" of the word, partially overlapping both letters. The background of the text area is solid black.

# Terraform Lambda Config

From  
[https://www.terraform.io/docs/providers/aws/r/lambda\\_function.html](https://www.terraform.io/docs/providers/aws/r/lambda_function.html)

## Example Usage

```
resource "aws_iam_role" "iam_for_lambda" {
  name = "iam_for_lambda"

  assume_role_policy = <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "sts:AssumeRole",
      "Principal": {
        "Service": "lambda.amazonaws.com"
      },
      "Effect": "Allow",
      "Sid": ""
    }
  ]
}
EOF
}

resource "aws_lambda_function" "test_lambda" {
  filename        = "lambda_function_payload.zip"
  function_name   = "lambda_function_name"
  role            = "${aws_iam_role.iam_for_lambda.arn}"
  handler         = "exports.test"
  source_code_hash = "${base64sha256(file("lambda_function_payload.zip"))}"
  runtime          = "nodejs4.3"

  environment {
    variables = {
      foo = "bar"
    }
  }
}
```

TWITTER: @natqab

EMAIL: [natalieqabazard@gmail.com](mailto:natalieqabazard@gmail.com)

## GITHUB

Code: <https://github.com/natqab/ssr-react-lambda>

Slides: <https://github.com/natqab/react-boston-2017>