

Publishing Your JavaScript App



Steve Buchanan

DevOps Architect

@buchatech | www.buchatech.com

Overview

Exploring the Components Needed for Hosting JavaScript Apps

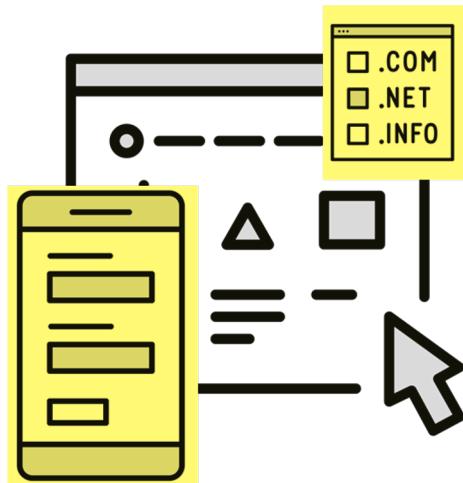
Exploring Options for Hosting JavaScript Apps

Demo: Publish a JavaScript App

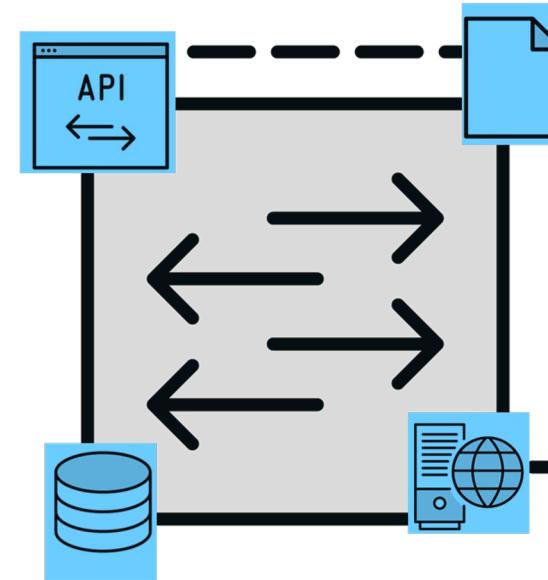


Exploring the Components Needed for Hosting JavaScript Apps

JavaScript App Components



Front end



Back end



JavaScript App Components - the Tech

Front end includes *HTML, CSS, and JavaScript*.

Here are two **CSS frameworks** that can be used on the front end or a web app:

- [Bootstrap](#): A popular CSS framework used for web apps

- [Tailwind](#): A utility-first CSS framework used for accelerated UI development

Here is a list of **JavaScript frameworks** that can be used on the front of the web app:

- [React](#)
- [Vue](#)
- [Angular](#)

Back end includes **JavaScript** running on a **Node.js runtime server**.

Here are some **frameworks** that can be used on the back end of a **JavaScript web app**:

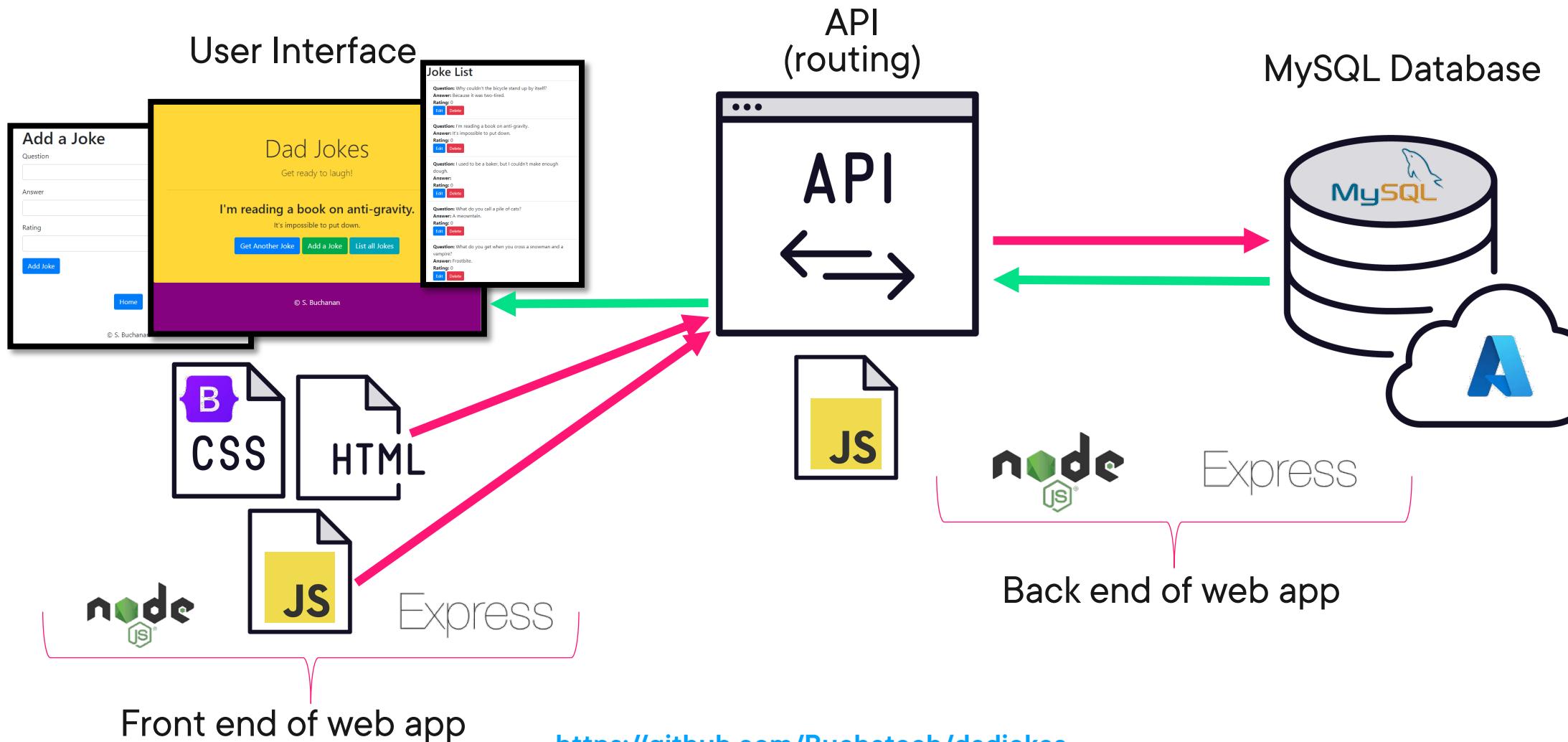
- [Express](#): Fast, unopinionated, minimalist web framework for Node.js.
- [Sails](#): Realtime MVC framework for Node.js.

Databases that can be used with **JavaScript web apps** are:

- [MongoDB](#)
- [MySQL](#)
- [PostgreSQL](#)



Dad Jokes Web App Architecture



Dad Jokes File/Directory Structure

```
- app/  
  - config/  
    - dbinfo.js  
  - features/  
    - addjoke/  
      - addjoke.html  
      - addjoke.js  
    - jokelist/  
      - jokelist.html  
      - jokelist.js  
  - views/  
    - index.html  
- apiserver.js  
- index.js
```



Dad Jokes File/Directory Structure

```
- app/  
  - config/  
    - dbinfo.js  
  - features/  
  - addjoke/  
    - addjoke.html  
    - addjoke.js  
  - jokelist/  
    - jokelist.html  
    - jokelist.js  
- views/  
  - index.html  
- apiserver.js  
- index.js
```



Add a Joke

Question

Answer

Rating

[Add Joke](#)

[Home](#)

© S. Buchanan



Dad Jokes File/Directory Structure

```
- app/  
  - config/  
    - dbinfo.js  
  - features/  
    - addjoke/  
      - addjoke.html  
      - addjoke.js  
    - jokelist/  
      - jokelist.html  
      - jokelist.js  
  - views/  
    - index.html  
- apiserver.js  
- index.js
```

Joke List

Question: Why couldn't the bicycle stand up by itself?	Answer: Because it was two-tired.
Rating: 0	Edit Delete
Question: I'm reading a book on anti-gravity.	Answer: It's impossible to put down.
Rating: 0	Edit Delete
Question: I used to be a baker, but I couldn't make enough dough.	Answer:
Rating: 0	Edit Delete
Question: What do you call a pile of cats?	Answer: A meowntain.
Rating: 0	Edit Delete
Question: What do you get when you cross a snowman and a vampire?	Answer: Frostbite.
Rating: 0	Edit Delete
Question: A man walked into a bar	Answer: He sad ouch
Rating: 3	Edit Delete



Dad Jokes File/Directory Structure

```
- app/  
- config/  
  - dbinfo.js  
- features/  
  - addjoke/  
    - addjoke.html  
    - addjoke.js  
  - jokelist/  
    - jokelist.html  
    - jokelist.js  
- views/  
  - index.html  
- apiserver.js  
- index.js
```



```
JS dbinfo.js  ×  
code > jokesAPI > v7 > app > config > JS dbinfo.js > ...  
1 // Database configuration  
2 const mysql = require('mysql');  
  
3 const db = mysql.createConnection({  
4   host: 'mysql-1234567890.mysql.database.firebaseio.com',  
5   user: 'root',  
6   password: '#XXXXXXXXXXXXXX',  
7   database: 'jokesdb' 8});  
9  
10 db.connect((error) => {  
11   if (error) {  
12     console.error('Database connection error:', error);  
13   } else {  
14     console.log('Connected to the database');  
15   }  
16 });  
17  
18 module.exports = db;  
19
```



Dad Jokes File/Directory Structure

```
- app/
  - config/
    - dbinfo.js
  - features/
    - addjoke/
      - addjoke.html
      - addjoke.js
    - jokelist/
      - jokelist.html
      - jokelist.js
  - views/
    - index.html
- apiserver.js
-index.js
```

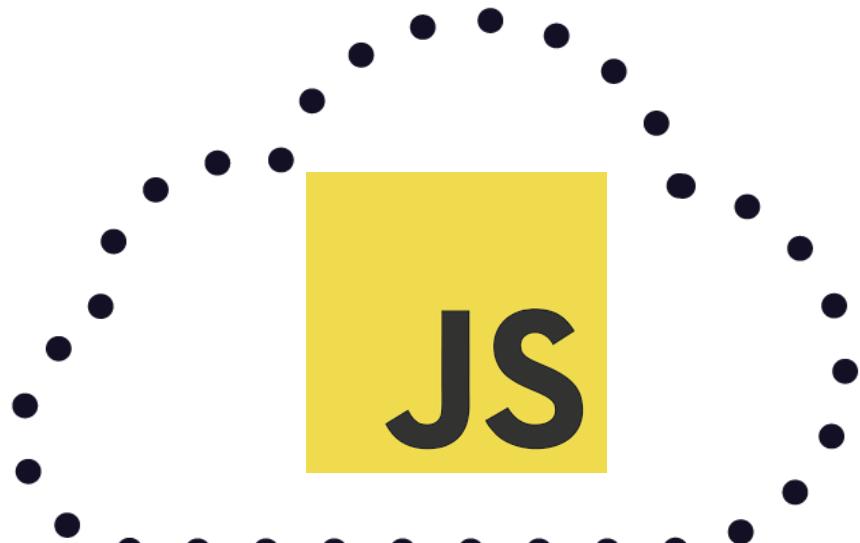
```
js apiserver.js ✘
code > jokesAPI > v7 > app > JS apiserver.js > ↗ router.get('/jokes') callback > ↗ db.query() callback
1  const express = require('express');
2  const db = require('./config/dbinfo');
3  const router = express.Router();
4
5  // Create a joke
6  router.post('/jokes', (req, res) => {
7    let joke = req.body;
8    let sql = 'INSERT INTO jokes (question, answer, rating) VALUES (?, ?, ?)';
9    let values = [joke.question, joke.answer, joke.rating];
10   db.query(sql, values, (err, result) => {
11     if (err) throw err;
12     res.send('Joke has been added...');
13   });
14);
15
16 // All jokes
17 router.get('/jokes', (req, res) => {
18   let sql = 'SELECT * FROM jokes';
19   db.query(sql, (err, results) => {
20     if (err) throw err;
21     res.send(results);
22   });
23);
24
25 // A single joke
26 router.get('/jokes/:id', (req, res) => {
27   let sql = `SELECT * FROM jokes WHERE id = ${req.params.id}`;
28   db.query(sql, (err, result) => {
29     if (err) throw err;
30     res.send(result);
31   });
32});
```



Exploring Options for Hosting JavaScript Apps



What Is JavaScript Hosting



JavaScript hosting refers to web hosting that supports JavaScript-based apps.

For example, Node.js is a popular JavaScript runtime environment that is used to building scalable & fast web apps. To host a Node.js app, you will need a Node.js-friendly host.

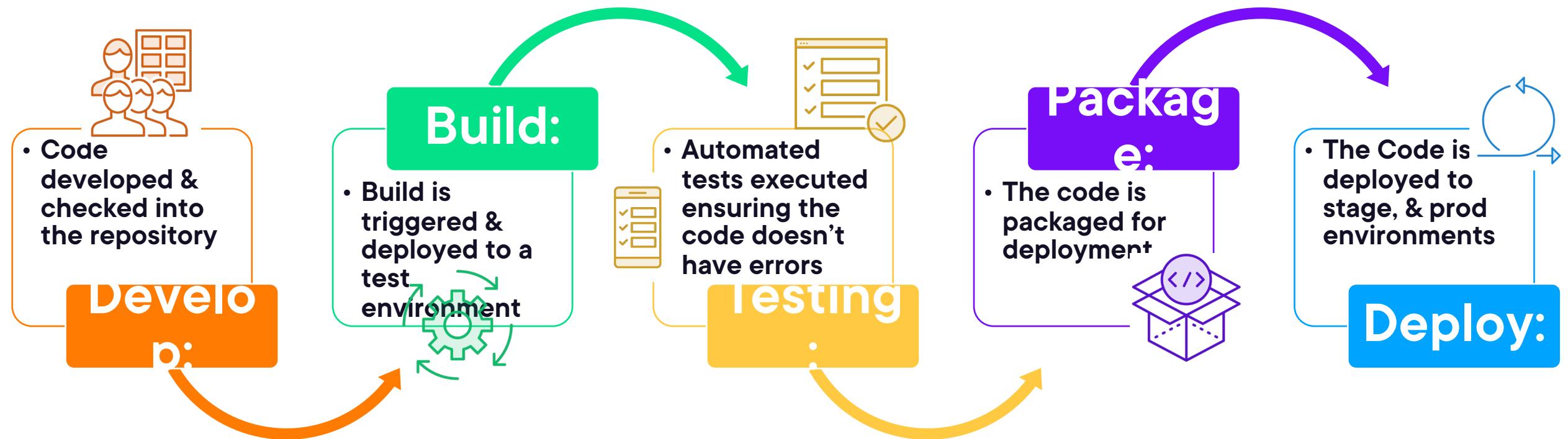
There are many options for JavaScript hosting, including:

- Shared servers
- Virtual private servers (VPS)
- Dedicated servers
- Containers



CI/CD for JavaScript

CI/CD stands for **Continuous Integration & Continuous Deployment**. It's a software engineering practice that helps teams to collaborate better and improve the software overall.



JavaScript Hosting Options



Render
www.render.com



Cloud:
 Azure
 Google Cloud Platform
 Amazon Web Services



heroku

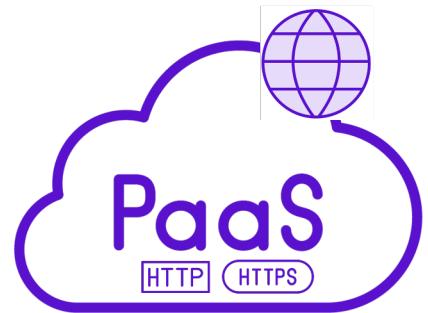
 **DigitalOcean**

Heroku
www.heroku.com

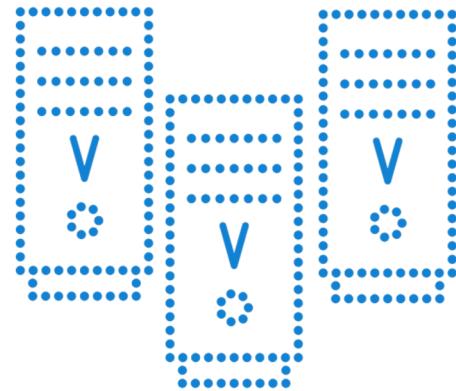
Digital Ocean
www.digitalocean.com



JavaScript Hosting Options - Azure



Azure App Service
(Managed web servers)



Virtual Machines
(Scalable VMs)



Azure Kubernetes Service AKS
(Managed Kubernetes Clusters)



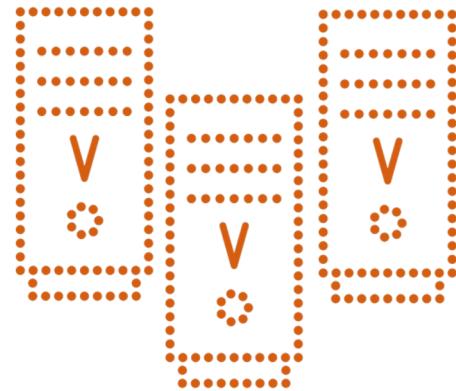
Azure Databases
MySQL Postgres



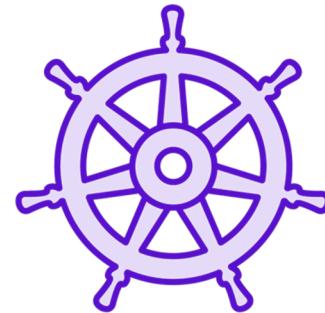
JavaScript Hosting Options - GCP



App Engine
(Managed web servers)



Compute Engine
(Scalable VMs)



Google Kubernetes Engine GKE
(Managed Kubernetes Clusters)



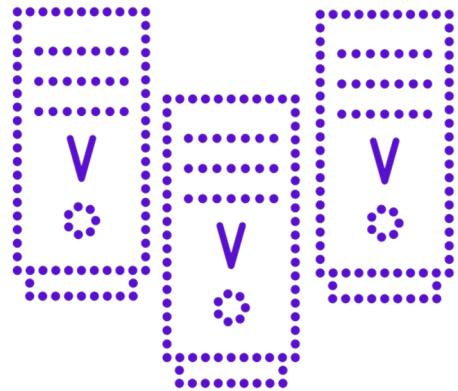
Cloud SQL
MySQL
PostgresSQL



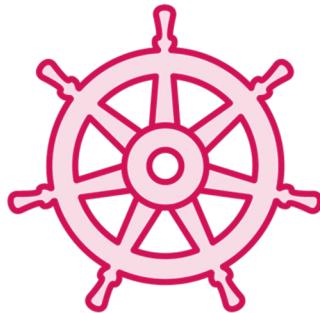
JavaScript Hosting Options - AWS



Elastic Beanstalk
(Managed web servers)



EC2 Instances
(Scalable VMs)

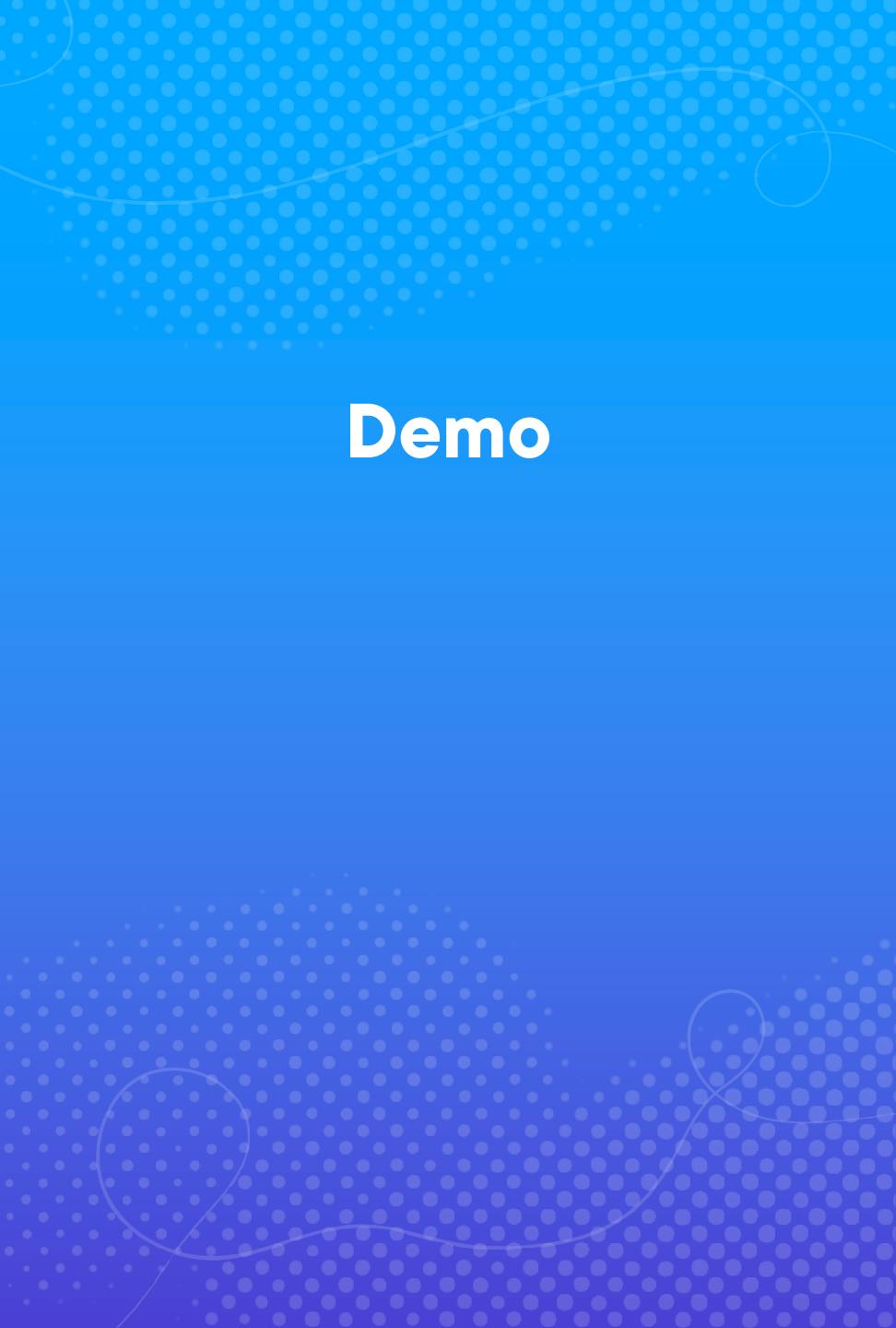


Elastic Kubernetes Service EKS
(Managed Kubernetes Clusters)



RDS
MySQL
PostgreSQL





Demo

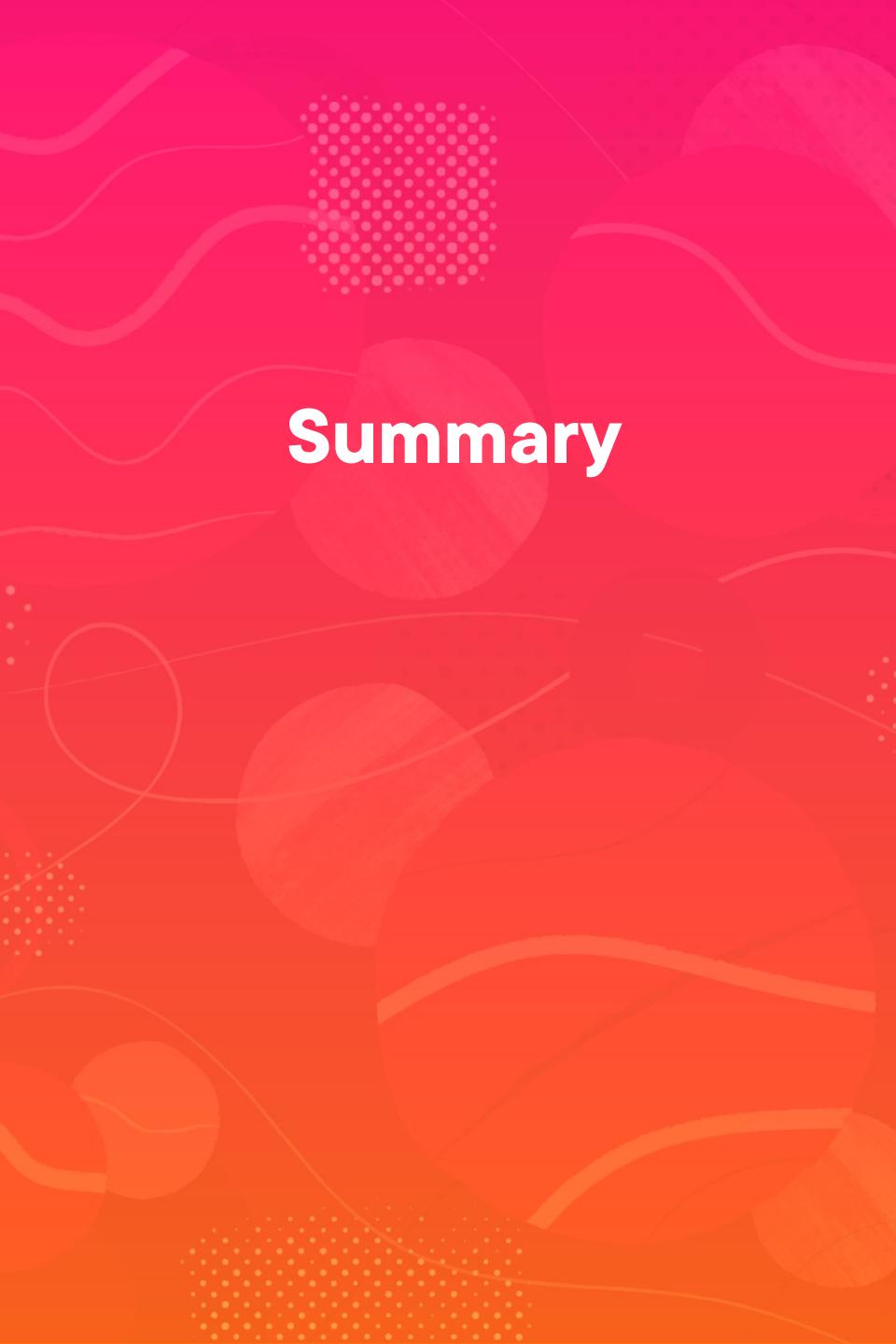
Demo: Publish a JavaScript App to Hosting Service



Demo

**Demo: Publish a JavaScript App to
Kubernetes in Cloud**





Summary

In this module we covered:

- ❑ Components Needed for Hosting JavaScript Apps
- ❑ Walked through the Dad Jokes Web App Architecture & File/Directory Structure
- ❑ What JavaScript Hosting is
- ❑ How CI/CD works with a JavaScript app
- ❑ Options for Hosting JavaScript Apps
- ❑ We saw how to Publish a JavaScript App to Kubernetes in Cloud

Why this is important:

- ❑ Up to this point we have planned, designed, and built the functionality of our JavaScript app. This final module covers the final step in your JavaScript app journey - making it available for users.
- ❑ Beyond building a JavaScript app you need to understand how to publish it. This knowledge will serve you time and time again.

