serverless

How Serverless Is Changing the World for Developers

serverless

Gareth Mc Cumskey

gareth.mccumskey@serverless.com

Current: Developer Advocate and Customer Success Engineer at Serverless Inc.

Former: Building sites since early 2000’s, first professional role in 2008, building serverless apps since 2016

serverless

What is Serverless?

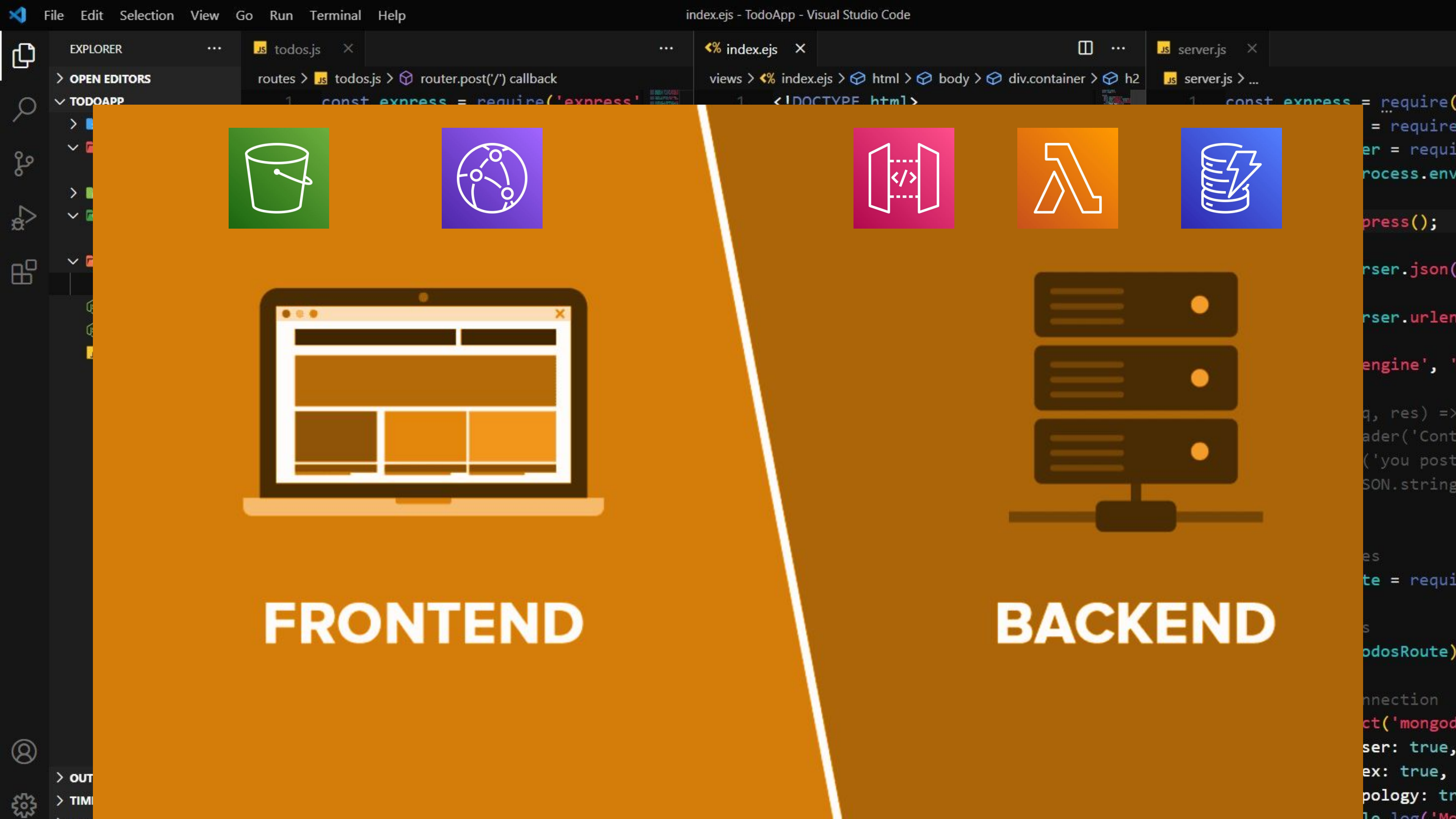
Your application is just a collection of code

Break it up in a way to use cloud services instead

Its is not just taking code and dropping it into Lambda

Architecting an application to make the best of the cloud

Cloud Native


File Edit Selection View Go Run Terminal Help
index.ejs - TodoApp - Visual Studio Code
EXPLORER
todos.js
index.ejs
server.js
OPEN EDITORS
TODOAPP
routes > todos.js > router.post('/') callback
views > index.ejs > html > body > div.container > h2
server.js > ...
FRONTEND
BACKEND

The usual suspects

No infrastructure management

Accelerated time to market

Only pay for what you use

Traffic management built in

The not so usual suspects

We’ve figured out after years of building Serverless applications

Spent time working with teams

Experiencing it ourselves building Serverless solutions

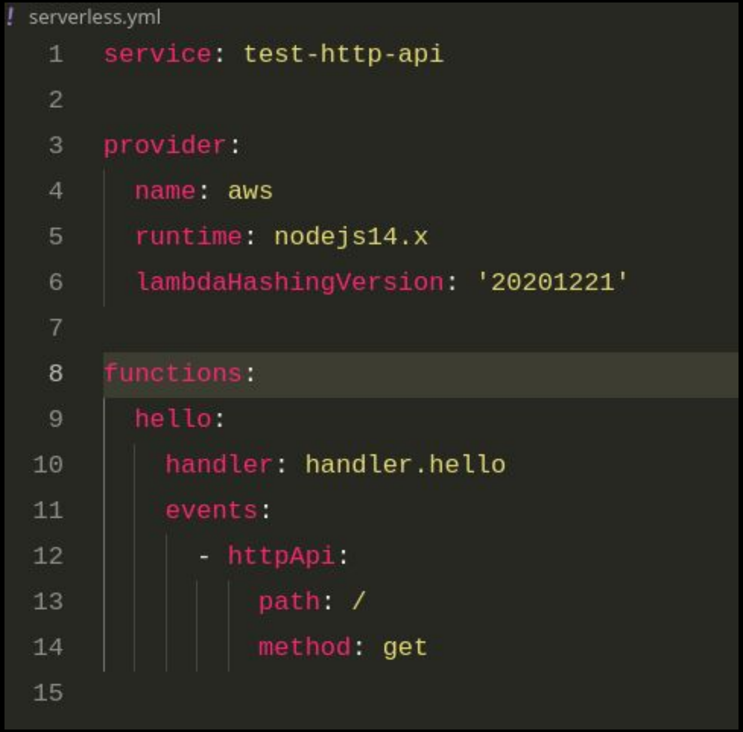
Frameworks help a lot!

Single configuration file

Automates creation and setup of all cloud primitives

Sync is maintained in the cloud

Can share with team and all work together to maintain


! serverless.yml

```yaml
service: test-http-api

provider:
  name: aws
  runtime: nodejs14.x
  lambdaHashingVersion: '20201221'

functions:
  hello:
    handler: handler.hello
    events:
      - httpApi:
          path: /
          method: get

```

serverless

Closer to production

serverless

The good ole days

Develop and test locally

After a few weeks team gets a release together

Tosses it over the wall to DevOps (or System Administrators)

App breaks because there is are differences in the two environments

Teams had been moving to better release configurations but still local versus production

Serverless Days

Because of using cloud services so extensively local is difficult

A better development model is to push changes to the Cloud and test

May seem initially slow benefits are multiple

You are on production essentially

IAM, latencies, concurrency, API validation and a lot more

serverless deploy function -f functionName

serverless

Lambda forgiving of “bad code”

The good ole days

Write “bad code” it can bring down an entire application

Shared resources between threads

Clusters help but eventually all machines can get swamped

Containerisation only helps so much

Serverless Days

A single Lamba executes within its own context

Built in timeouts means even if code is bad it eventually stops

Tests have shown that noisy neighbour issues are minimal at best

Ephemeral nature of Lambda is a savior here

Less focus on optimisation and more on what pays the bills

Less code, less errors

The good ole days

A web application needed to perform all processing of the HTTP request in code

From routing based on paths, authentication, CORS, responses and more

That meant a lot more code had to be written

The single largest causes of failures in platforms is code written by developers

Makes you wonder why low/no code solutions have been getting so popular

serverless

Serverless Days

API Gateway handles pathing, CORS and responses

DynamoDB simplifies database interaction codewise

S3 replaces file system interactions

Serverless Framework doesn’t add any additional code to your application

serverless

Continuous delivery much easier

The good ole days

As we said before, releases were tossed over the wall

Friction between infrastructure teams and developers

Lead to big bang releases with a ton of changes

Has recently started changing but can be difficult to manage cleanly

Biggest cause is usually large clusters of infrastructure that are slow to update

Serverless Days

Infrastructure is a part of the service

Changes can be pushed to production with a single command

Or a merge to GitHub

Lots of small services decreases surface area of all changes

Automation with CI/CD is relatively simple

serverless

Thanks.
Questions?