BOOTSTRAPPING APPLICATIONS USING
AWS AMPLIFY
WITH
AWS APPSYNC
Fullstack development

WHO AM I

- Peter
- Twenty years working in software
- Majority for small software companies
- Majority as a technical product manager
- Current focus is fullstack development

BACKEND-AS-A-SERVICE / SERVERLESS

- Interactive software ASAP
- Leverage the work of others
- Humans are expensive
- (At scale, things are different)

MY VIEW OF SERVERLESS

- Authentication
- Database
- Storage
- Functions
- Hosting
- APIs
- AI / ML
- Etc, etc

AWS APPSYNC
AWS GraphQL Service

AWS AppSync is a serverless back-end for mobile, web and enterprise applications

AWS AMPLIFY FRAMEWORK

- Evolution of AWS Mobile Hub
- Amazon's answer to Google Firebase
- Make it easy for application (front-end) developers to use (consume) back-end resources

The foundation for your cloud-powered mobile & web apps

AMPLIFY CATEGORIES

- Analytics
- API (GraphQL via **AppSync** and REST)
- Authentication
- Functions
- Hosting
- Storage
- Notifications

USING AMPLIFY

1.   AWS Account
2.   Amplify CLI
3.   Docs and libs

AMPLIFY DOCS AND LIBS

- iOS - Swift
- Android - Java
- Web - JavaScript - **React**, Angular, Ionic, Vue
- React Native

DEMO

Prereq's

- Amplify CLI installed and configured
- React project created, libraries installed and amplify init

CURRENT STATE - LOCAL

- amplify/backend folder - current local config
- amplify/#current-cloud-backend folder - last push'ed config
- amplify/.config folder - project settings
- .amplifyrc file - amplify project specific config

CURRENT STATE - REMOTE

- 2 new roles (auth and unauth)
- S3 bucket
- Cloudformation Stack

APPSYNC INTEGRATION

1. Setup API endpoint with authentication and schema in client
2. Generate JavaScript (TypeScript) code from schema
3. Write app code to run queries, mutations and subscriptions

DEMO

---

amplify add api
amplify push

DEMO

after amplify push completes

RESULT - LOCAL

- aws-export.js - AWS config (GraphQL endpoint, API key, etc)
- graphql folder - schema and codegen for queries, mutations and subscriptions
- types for Flow or TypeScript

RESULT - REMOTE

- AppSync project created
- Lots of other stuff

DEMO

write some code

CLIENT OPTIONS

- Amplify GraphQL client
- AWS AppSync (SDK) / Apollo client

A WORD ABOUT AUTH

- amplify add auth
- Creates Cognito User and Identiy Pools
- can be integrated with API

DIRECTIVES

- Part of schema (@)
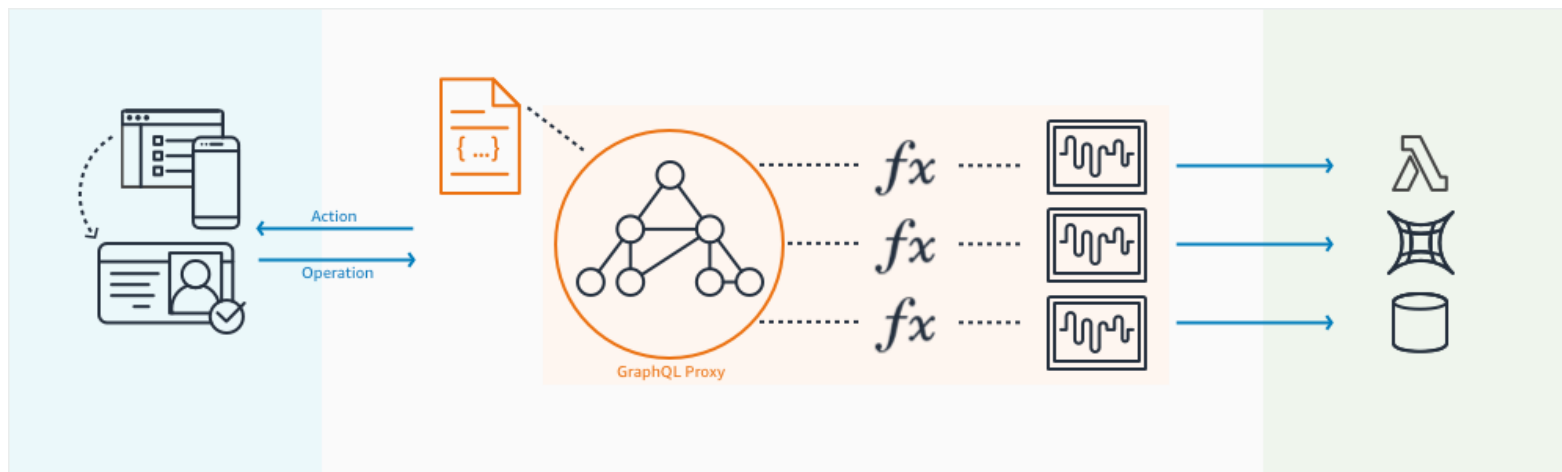- @model
- @auth
- @connection
- @searchable
- @versioned

```graphql
type Blog @model {
  id: ID!
  name: String!
  posts: [Post] @connection(name: "BlogPosts")
}
type Post @model {
  id: ID!
  title: String!
  blog: Blog @connection(name: "BlogPosts")
  comments: [Comment] @connection(name: "PostComments")
}
type Comment @model {
  id: ID!
  content: String
  post: Post @connection(name: "PostComments")
}
```

```
type PrivateNote
  @model
  @auth(rules: [{allow: owner}])
{
  id: ID!
  content: String!
}
```

```
type Task
  @model
  @auth(rules: [
      {allow: groups, groups: ["Managers"], queries: null,
mutations: [create, update, delete]},
      {allow: groups, groups: ["Employees"], queries: [get,
list], mutations: null}
    ])
{
  id: ID!
  title: String!
  description: String
  status: String
}
```

APPSYNC
ARCHITECTURE

Source: https://docs.aws.amazon.com/appsync/latest/devguide/system-overview-and-architecture.html

HOW IT WORKS

- Resolvers "translate" GraphQL to do something with a "Data Source"
- Data Sources = DynamoDB, Lambda, ElasticSearch, Aurora, HTTP
- Translation via VTL (Apache Velocity Template Language)

DEMO

AWS Console - AppSync

TAKEAWAYS

- Teams may struggle with Amplify
- Be ready to edit .gitignore
- Client side codegen is probably not that valuable
- Amplify/Appsync - ownership matters
- If you just want GraphQL API - don't start with Amplify

TRADE-OFF

- Serverless - Quick to Production
- vs
- Traditional - Control

- Prototyping?

Thank you!
@peter_dyer
fullsapps.com