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
- Outsource what I can
- Humans are expensive
- (At scale, things are different)

SERVERLESS

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

AWS APPSYNC

AWS managed 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

Amplify CLI installed and configured

DEMO #1

create-react-app
npm install aws-amplify aws-amplify-react
amplify init

DEMO #2

after amplify init completes

RESULT - LOCAL

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

RESULT - 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 #3

amplify add api

LEVERAGING THE AMPLIFY - APPSYNC INTEGRATION

- Edit schema as desired in client
- Update back-end via CLI

- amplify push
- amplify publish
- amplify codegen
- amplify api gql-compile

DEMO #4

amplify push (first time)

DEMO #5

after amplify push comletes

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

DIRECTIVES

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

CLIENT OPTIONS

- Amplify GraphQL client
- AWS AppSync SDK

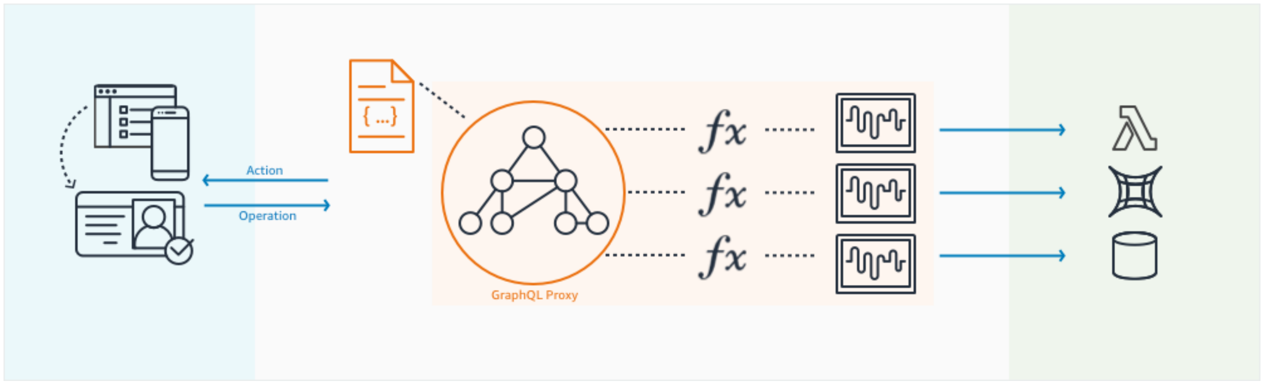
DEMO #6

coding

DEMO #7

AppSync

ARCHITECTURE



HOW IT WORKS

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

DEMO #8

new AppSync API using AWS Lambda (as Data Source)

TAKEAWAYS (PART 1)

- Nudged/pushed towards AWS DynamoDB
- Nudged/pushed towards AWS Cognito
- Teams may struggle with Amplify
- Be ready to edit .gitignore
- If you just want GraphQL API - don’t start with Amplify

TAKEAWAYS (PART 2)

- Amplify/Appsync - ownership matters
- If you already have Resolvers (or a backend) Appsync may not be a good fit
- Client side codegen is probably not that valuable
- Be aware of codegen goodies (filter, limit, nextToken)

TRADE-OFF

- Quick to Production
- Control (flexibility, understandability)

OTHER SERVERLESS OPTIONS

- AWS only big 3 doing GraphQL as a Service
- Google Firebase
- Prisma Cloud
- Others...

FINAL THOUGHTS

- Generous free tiers
- Consider for prototyping
- Reduce decision fatigue

Thank you!
@peter_dyer
fullsapps.com