MLP, AWS and AWS Amplify

Chris Modica
Principal SA, AWS
FSI, ANZ

Chris Modica

Principal SA

Sydney, Australia

🇦🇺🇮🇹🇮🇪👶🚴‍♀️🏃‍♀️

@Chris_Modica_
ln/christophermodica

3 years AWS, working with some of Australia's largest and most complex customers . First 2 years focused on B2B/ISV businesses across AU, NZ, US, UK, CN, ZA.

Ex. CTO, VPoE. Scaled and managed large teams (140-160+) across multiple countries, grown and created new businesses in tech, propTech, classifieds and FSI.

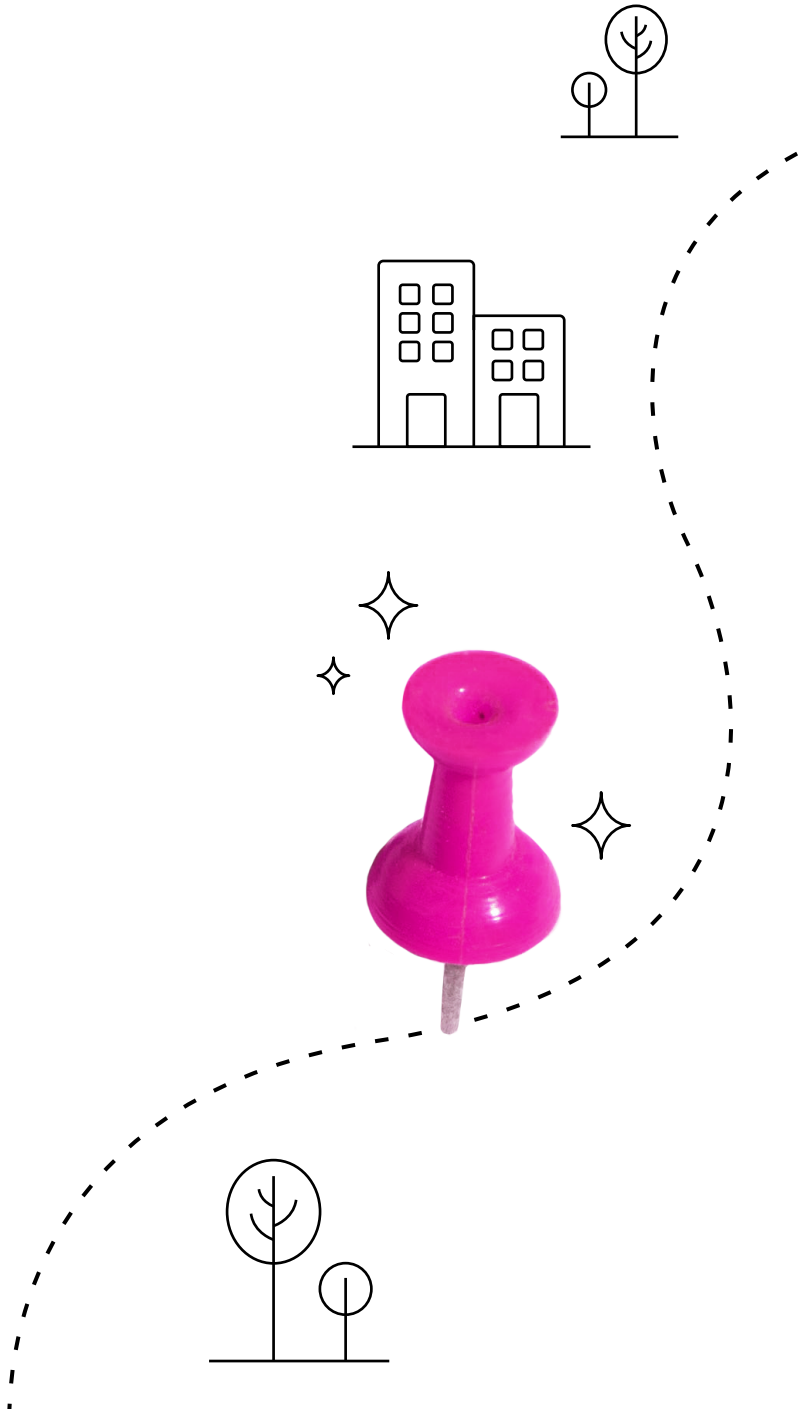
Currently enjoy working on the CBA Team at AWS

aws

Agenda

All In Agile – Product Engineering

1. What is an MVP?
2. Before You Build
3. Leverage Frameworks
4. Flexible Architecture
5. Speed with Safety
6. Surviving the unexpected
7. Understanding your users



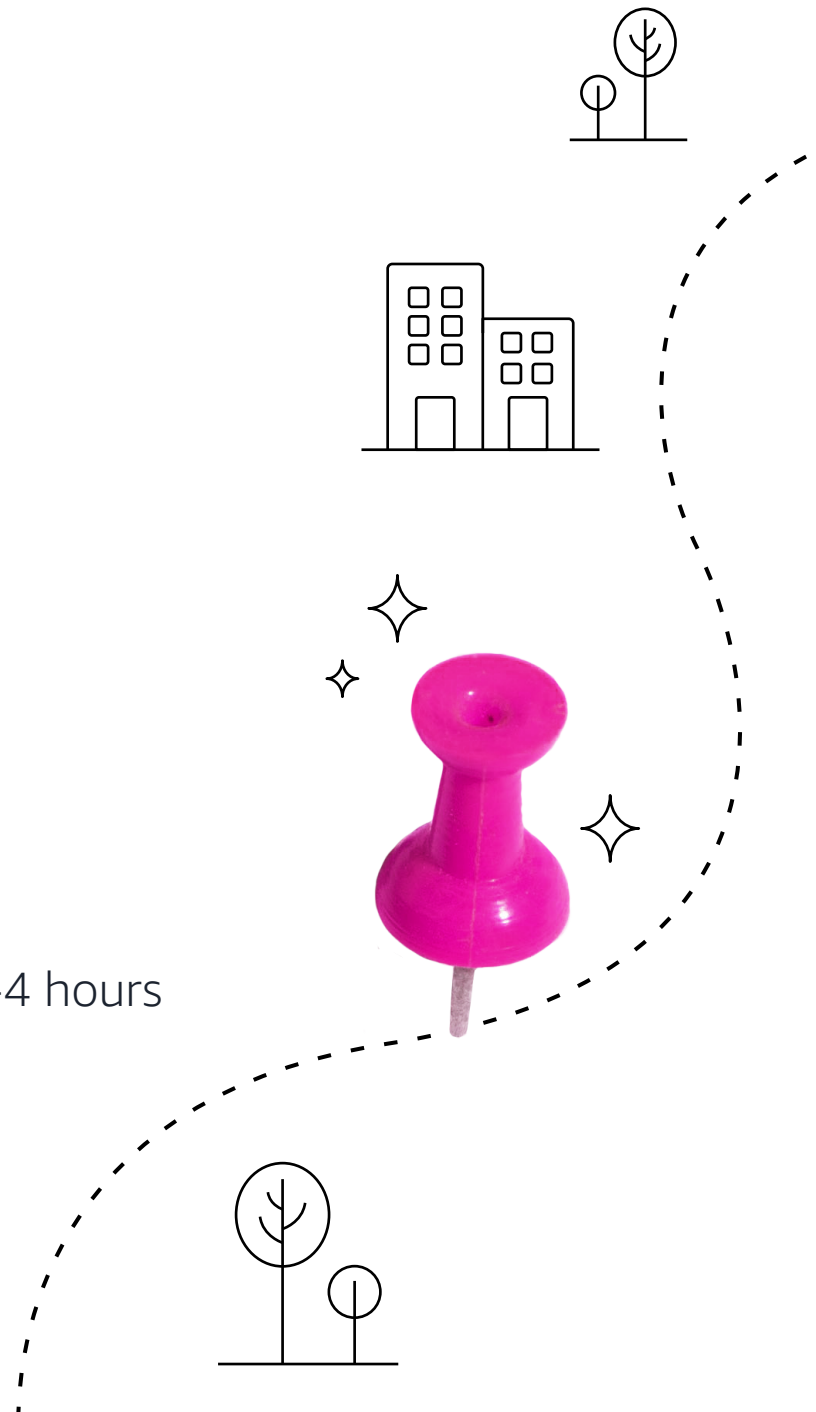
Agenda

Build on Technology

1. JavaScript Frameworks and Trends 2021

2. AWS Amplify overview

3. Learning pathways (*hands on coding*) - Next.js projects
    1. Watch and learn (video) ~ 30 mins
    2. Intermediate option or Advanced/Expert (Existing AWS account) ~ 2-4 hours
    3. Need an AWS account/developer workstation – Future ½ day event



Time to Value

Opportunity cost

The fast companies are **440x**

We found that, compared to low performers, high performers have:

46x more frequent code deployments

440x faster lead time from commit to deploy

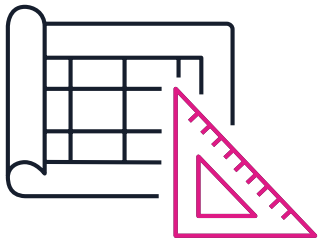
96x faster mean time to recover from downtime

5.0x lower change failure rate (changes are 1/5 as likely to fail)

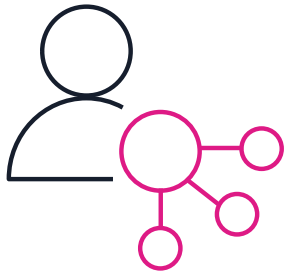
https://puppet.com/resources/report/2021-state-of-devops-report

What Is An MVP?

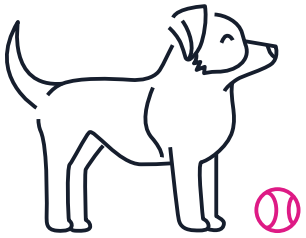
Minimum {X} Product



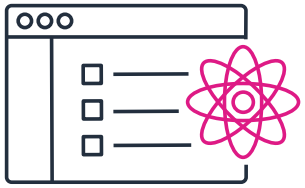
Viable



Usable



Loveable



Testable

”Left to Right” vs “Bottom to Top”

Functionality

Reliability

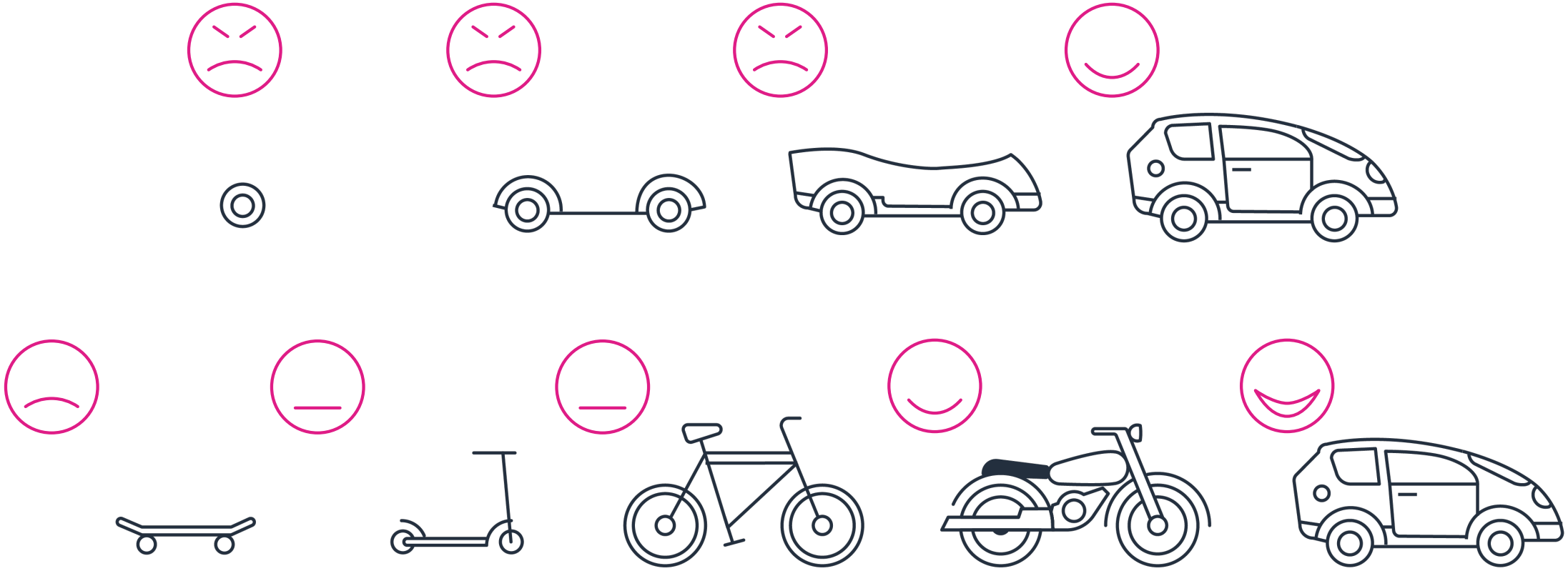
Usability

UX & Design

“If you aren't embarrassed by the first version of your product, you shipped too late.”

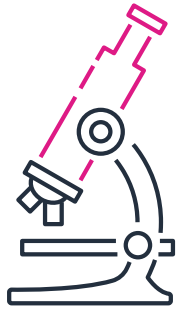
Reid Hoffman
Co-Founder LinkedIn

Product Iteration

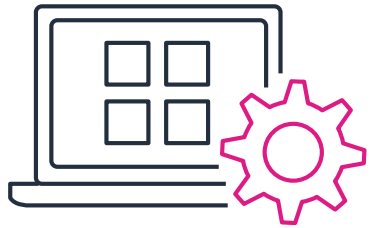


Optimize For Learning

Learn

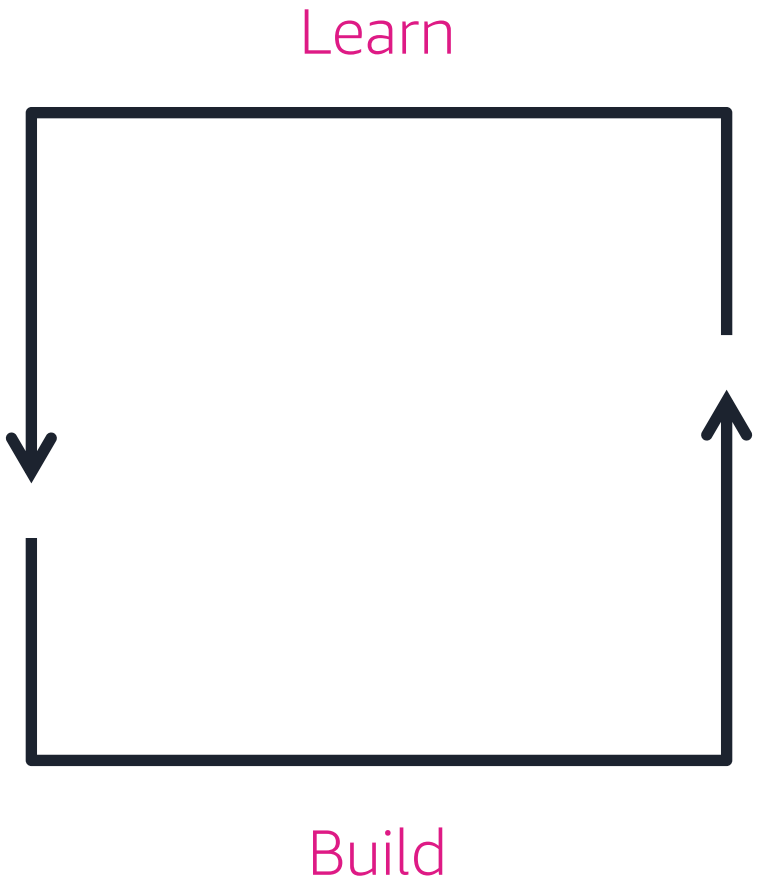


Experiment



Prototype

The Learn-Build-Learn Cycle



Before You Build

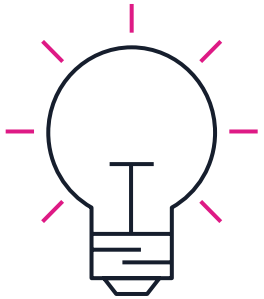
Are You Ready To Build an MVP?

Problem Definition

A clear understanding of the problem you are solving will ease communication of your idea with both stakeholders and users.

Problem Metrics

The number of people that have this problem, and the impact of the problem to those users.



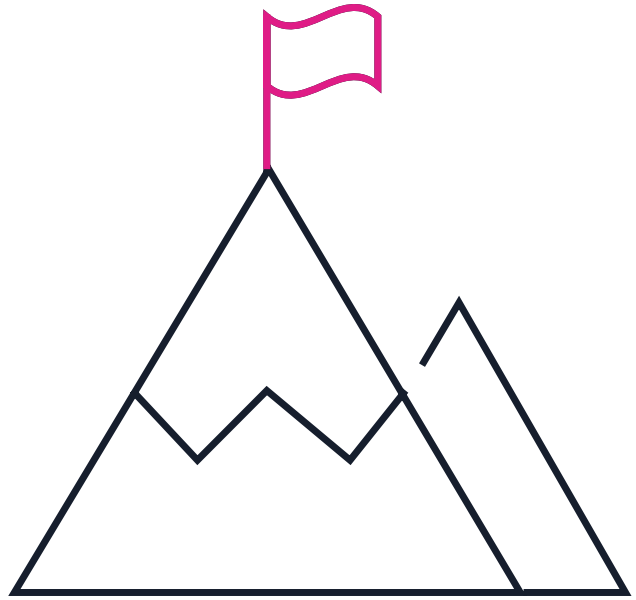
Solution Hypothesis

Initial hypothesis of how the problem will be solved for your users.

aws

MVP Is a State of Mind

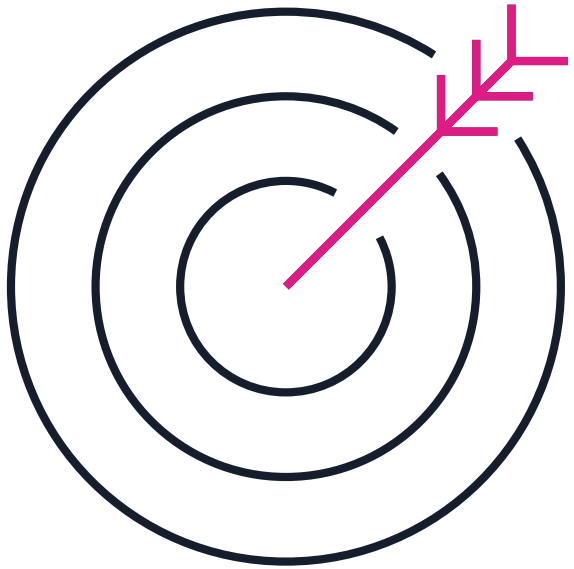
- Process not product
- Speed is paramount
- Set the right focus (10-100 customers)
- Customer feedback is critical
- Expect to change/pivot/adapt
- Weigh technical debt vs other kinds of debt



aws

MVP Feature Prioritization & Scope

- Focus on building and validating the strongest use case first
- In most cases any building should take < 3 person months
- Distinguish between being busy and creating value
- Minimize friction wherever possible for your users



aws

“Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.”

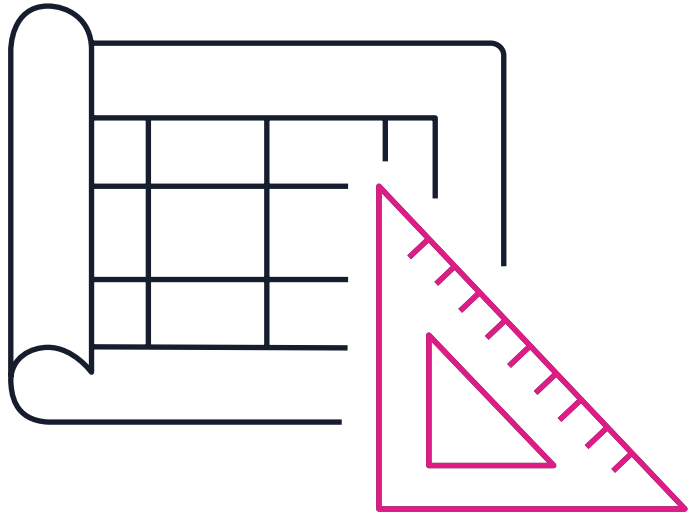
Antoine de Saint-Exupéry
Airman's Odyssey

Leverage Frameworks

What Architecture Is Right For Me?

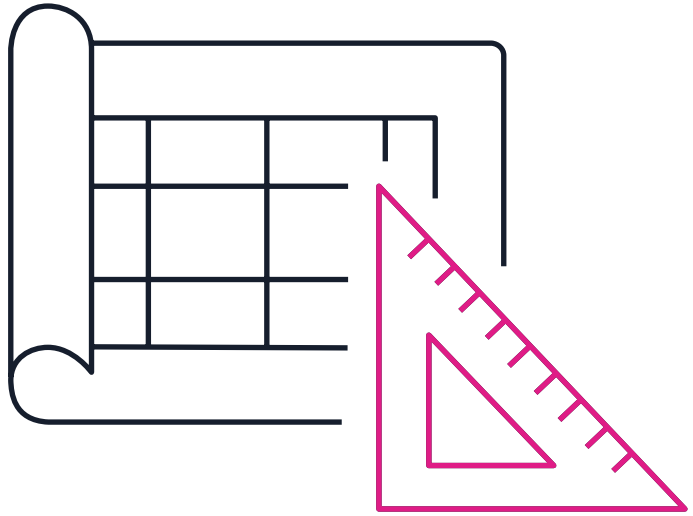
Ask yourself some important questions:

- What are you building?
- What are you optimizing for?
- What are your existing skill sets? What skill sets are available to you in the market?



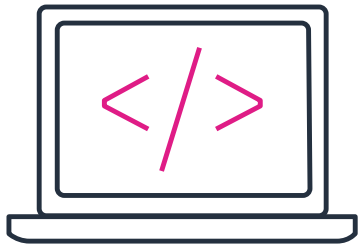
What Architecture Is Right For Me?

- The monolith is a viable option
- Leverage frameworks to increase your speed and confidence
- Very little of what you build in your MVP will be recognizable a year later
- Whatever you choose: leverage infrastructure as code!

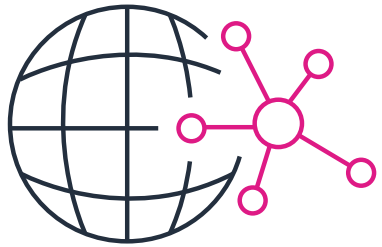


aws

The Code Continuum



Consider where
no code, and low code
solutions fit in

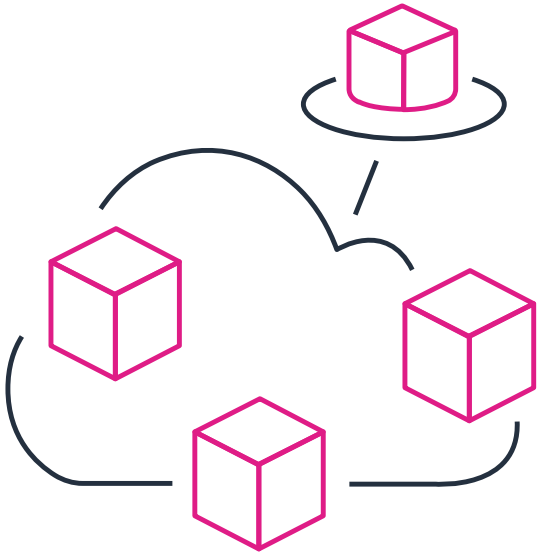


Abstraction
vs
flexibility tradeoffs

AWS Amplify

AWS Amplify is a set of tools and services that can be used together or on their own, to help front-end web and mobile developers build scalable full stack applications, powered by AWS.

- Configure and deploy backends quickly
- Seamlessly connect frontends
- Streamlined deployment of frontends
- Easily manage your users and content



https://aws.amazon.com/amplify/

aws

AWS Copilot

AWS Copilot is a command line interface (CLI) that enables customers to quickly launch and easily manage containerized applications on AWS.

- Architecture, not infrastructure
- Simple and powerful config
- Develop/Release/Operate

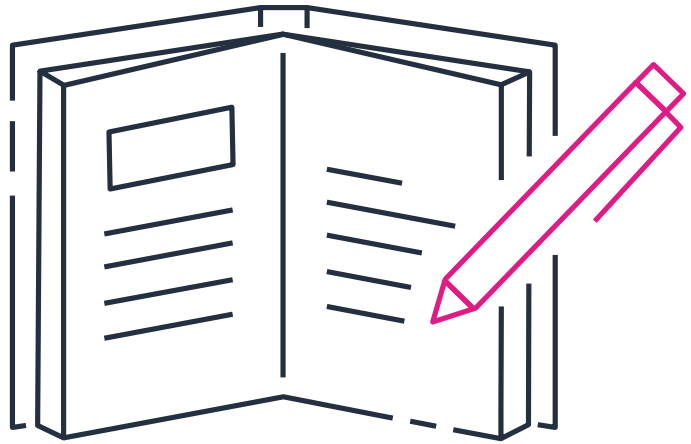
https://aws.amazon.com/blogs/containers/introducing-aws-copilot/

aws

AWS Solutions Library

The AWS Solutions Library offers a collection of cloud-based solutions for dozens of technical and business problems, vetted for you by AWS.

- Automatically deploys directly into your AWS account
- Library of AWS-vetted architecture diagrams



https://aws.amazon.com/solutions/

aws

AWS Cloud Development Kit

The AWS Cloud Development Kit (AWS CDK) is an open source software development framework to define your cloud application resources using familiar programming languages.

- Easier cloud onboarding
- Faster development process
- Customizable and shareable
- No context switching

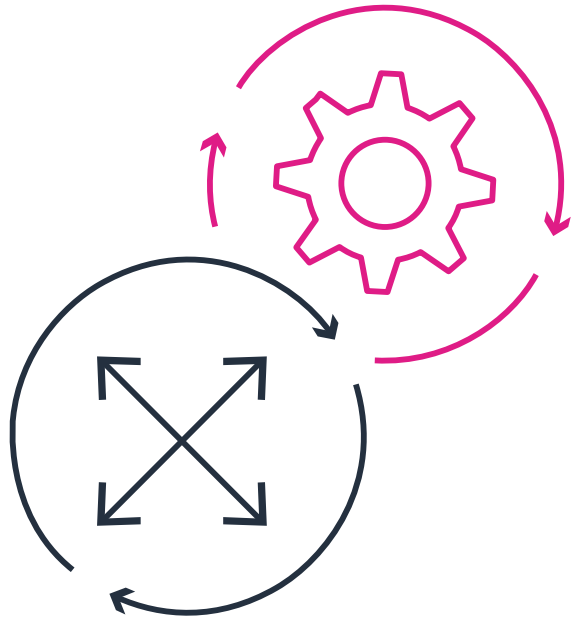
https://aws.amazon.com/cdk/

aws

AWS Solutions Constructs

AWS Solutions Constructs are vetted architecture patterns, available as an open-source extension of the AWS Cloud Development Kit, that can be easily assembled to create a production-ready workload.

- Infrastructure-as-code
- Pre-built, well-architected multi-service patterns
- Combine to create your own solutions
- Speed up your development cycle
- Consistently deliver Well-Architected apps



https://aws.amazon.com/solutions/constructs/

aws

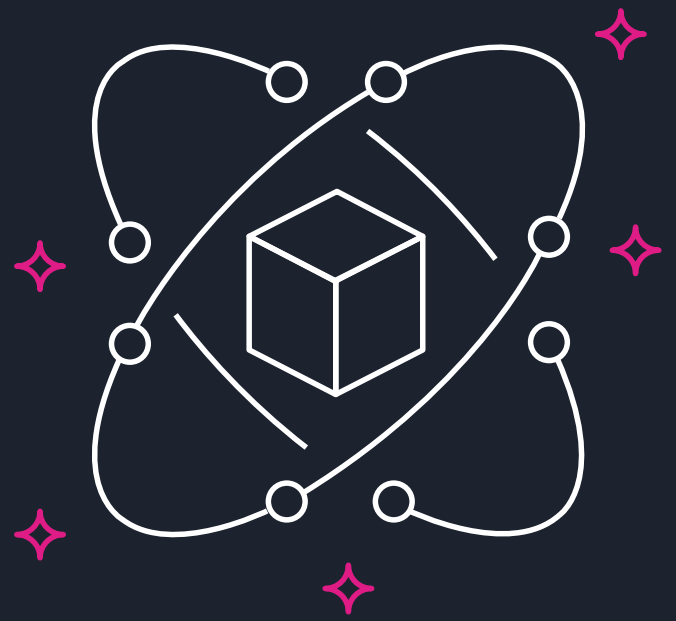
Flexible Architecture

Technology/Architecture Philosophy

- Avoid “undifferentiated heavy lifting” at the beginning
- Resist the temptation to add complexity unnecessarily
- Don’t introduce endogenous churn
- De-risk by transferring that risk to your provider
- Don’t be intimidated by the unknown

Prioritize Managed Services

- Choose the highest level of abstraction available to you that gets the job done
- You can always choose to de-abstract down the road, the reverse is much harder
- Non-functional requirements are baked in to managed services
- Operations burden is significantly reduced
- Security risks are significantly (but not completely) reduced



aws

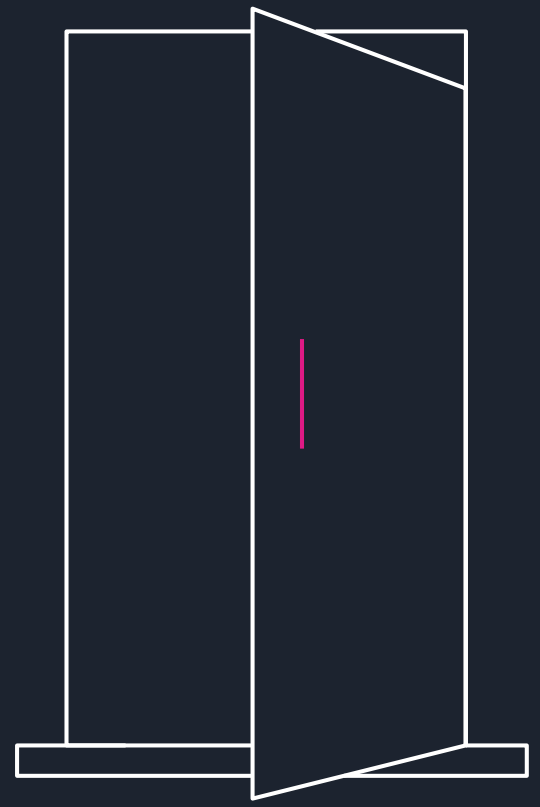
Minimize Throwaway Work

- Minimize, but don’t expect to eliminate re-work
- PMF can be described as a point where your product is breaking under growth
- Let your user feedback and desired outcomes drive changes, not technical debt
- The iterative learning cycle can mean evolution over revolution

aws

Make Reversible Decisions Quickly

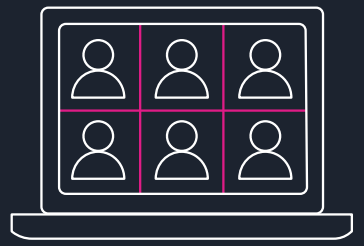
- At Amazon we call these decisions “two way doors”
- What is the cost/difficulty of reversing the decision?
- How do I measure the outcome?
- How do revert the decision?
- Be prepared to be wrong a lot initially



Externalize State

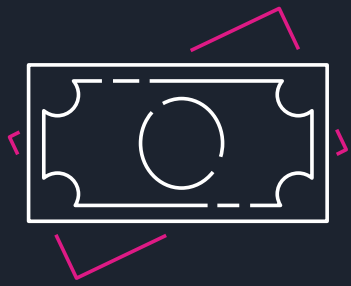
- Separating state from logic gives you optionality, scaling, and flexibility
- It also introduces new challenges and risks you need to manage

Externalize Authentication



User management is the pinnacle of “undifferentiated heavy lifting”

One of the highest risk components in the architecture



Payments is another prime example of this

aws

Building Guardrails

Build Guardrails, Not Gates

- Includes but not limited to both security and operations
- Teams should spend time solving problems that move the business ahead
- Experimentation should be encouraged
- Seeing the results of a change should be immediate
- Time spent firefighting, waiting on people or processes, or doing rework is waste
- Hiring talent is hard enough

Onboarding

- As you grow your team the ease and safety of onboarding new developers is critical
- You will have varying levels of trust
- A single source of truth for identity will make your life much easier
- Start documenting expected standards and processes early

aws

Onboarding

Consider the differences in:

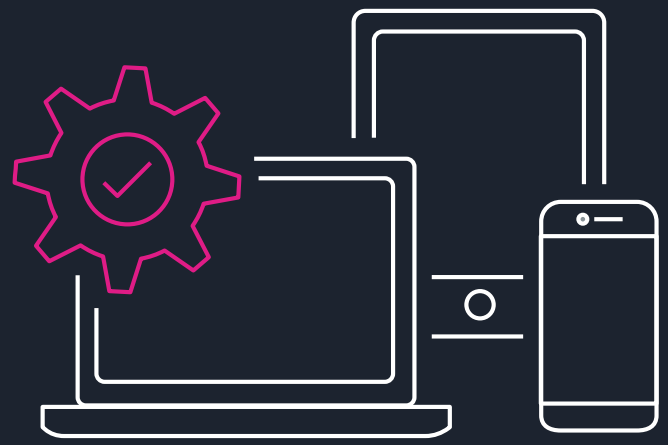
the ability to push code

access production systems

access production data

CI/CD

- Leverage the benefits of small, fast units of deployment
- If you aren't shipping those updates, you are missing the primary benefit
- Working on an MVP without CI/CD is like trying to "roll a cube"
- How much CI/CD is right initially?
- Does anyone really write tests?



aws

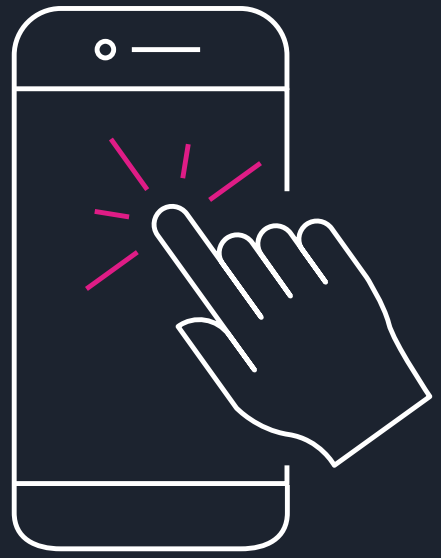
Observability

- Observability is what brings confidence to move quickly
- How do you know the impact of your changes?
- When do you need to rollback quickly?
- Key components:
  - Metrics
  - Logging
  - Tracing

aws

Synthetics

- You want to find issues **before** your customers/users do
- Continually verify your customer experience (even when you don't have any customer traffic)



aws

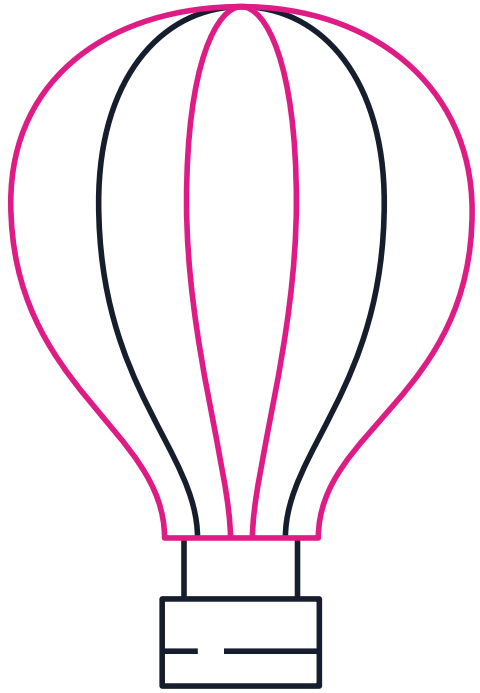
What level of confidence do you have in quickly making changes to your application today?

aws

Surviving The Unexpected

Every Once In Awhile, Look Up

- It’s easy to get tunnel vision when focused on the product
- There are important ways to minimize existential risk in your environment
- Don’t wait until the loss would be catastrophic
- **Minimize, not eliminate**



aws

Managed Backups

- Managed services help guard against failure
- They don’t protect you from accidents, errors, and malice
- Centralize and automate data protection
- Covers storage volumes, databases, and filesystems
- Validate and test the process before you need it

aws

Cost Visibility

- Ensuring you have visibility into your costs is critical to planning process
- Understanding your costs helps you reason about feature viability
- Unexpected spikes are indicators of a problem that you want to be aware of quickly
- Track your AWS credit usage
- Use budget alarms

Understanding Your Users

Validated Learning Drives MVP Iteration

- Learning what resonates with your users is a core purpose of your MVP
- Determine how users are actually using your MVP
- The only users that have value are active users
- User metrics are different than operational metrics
- Investors will expect you to be an expert on your users and their experience with your solution

aws

Measuring User Activity

- What are unexpected high use features?
- Where do you see unexpected drop off/abandonment?
- How do you need to update your hypothesis?
- If you can log it, you can build metrics around it.

Integrating Offline Tasks

Fake It Until You Make It

- Your MVP make look different to your users than from the inside
- Manual processes are normal and to be expected
- You can still track them to ensure good customer experience
- Lay the groundwork for future automation

Tracking Manual Work with Step Functions

- You can leverage workflow co-ordination between systems
- You can loop in humans with yes/no decisions via email
- Applies to any back office process

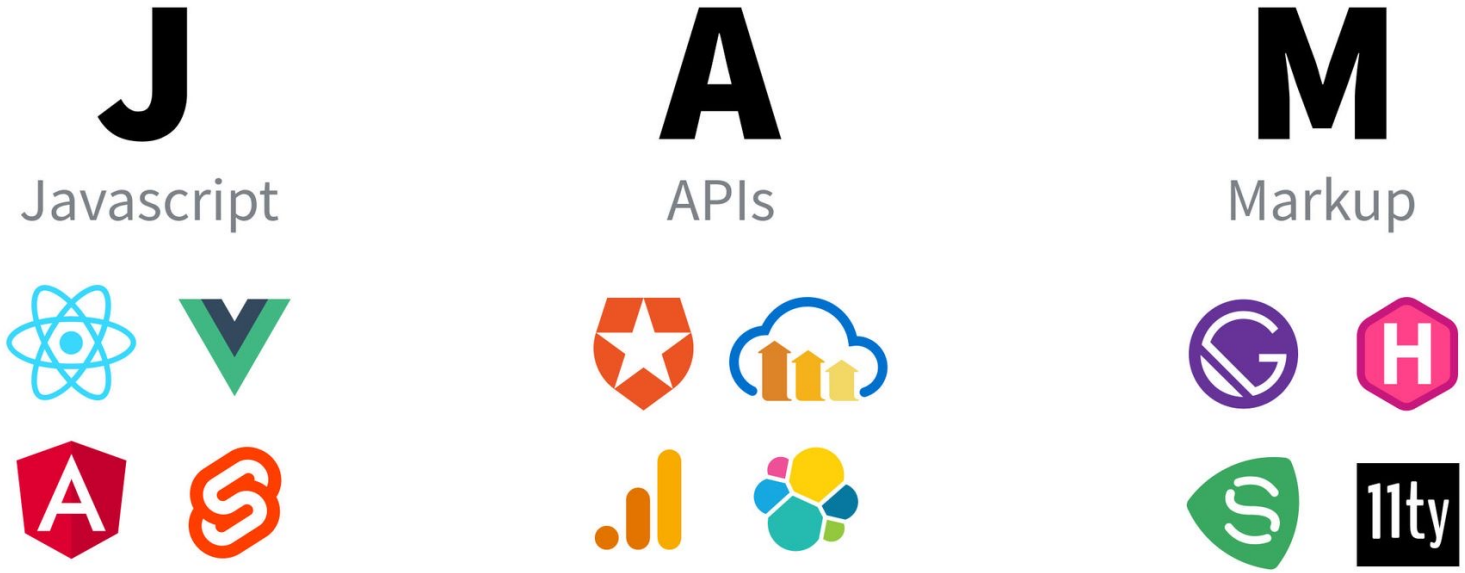
JavaScript Frameworks and Trends 2021

From LAMP to MEAN to JAM Stack

LAMP Stack	MEAN Stack	JAM Stack
Linux	mongoDB.	JS
APACHE HTTP SERVER	express	
MySQL	ANGULARJS by Google	{ api }
php	node js	</> Markup

https://mryap.github.io/From-LAMP-to-JAM/

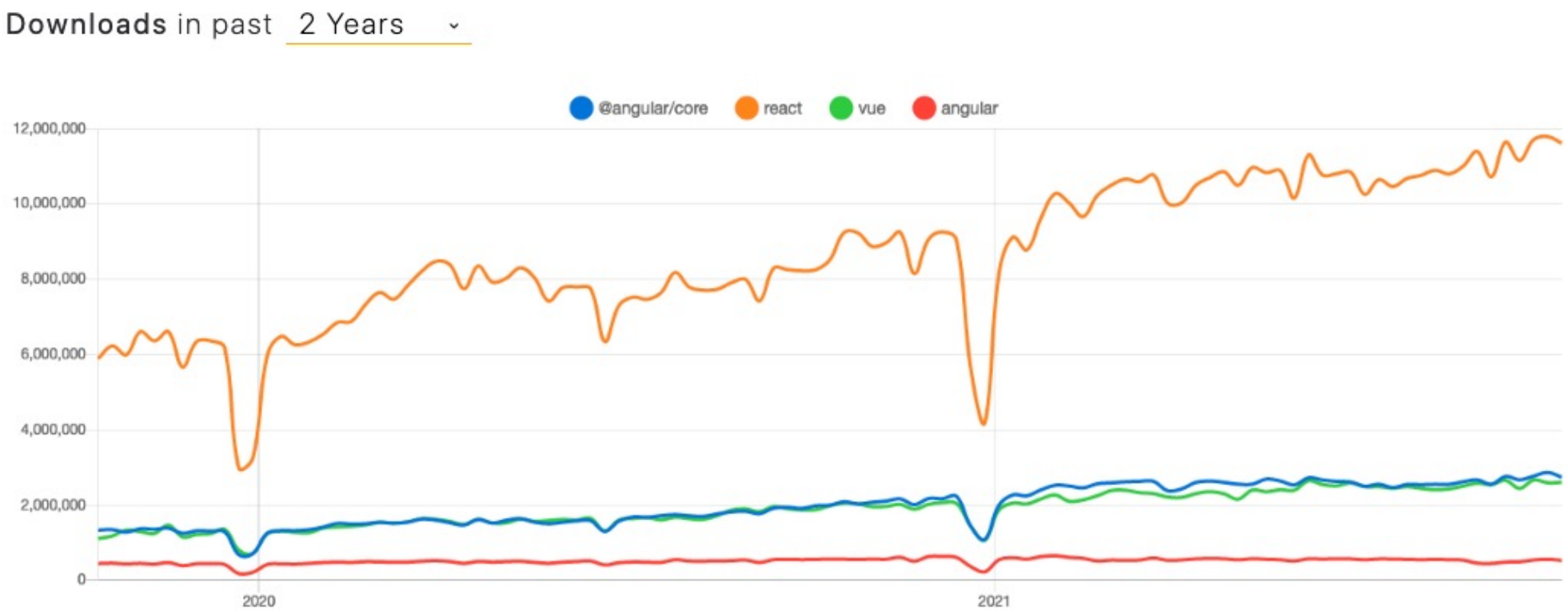
JAM Stack



https://www.freecodecamp.org/news/what-is-the-jamstack-and-how-do-i-host-my-website-on-it/

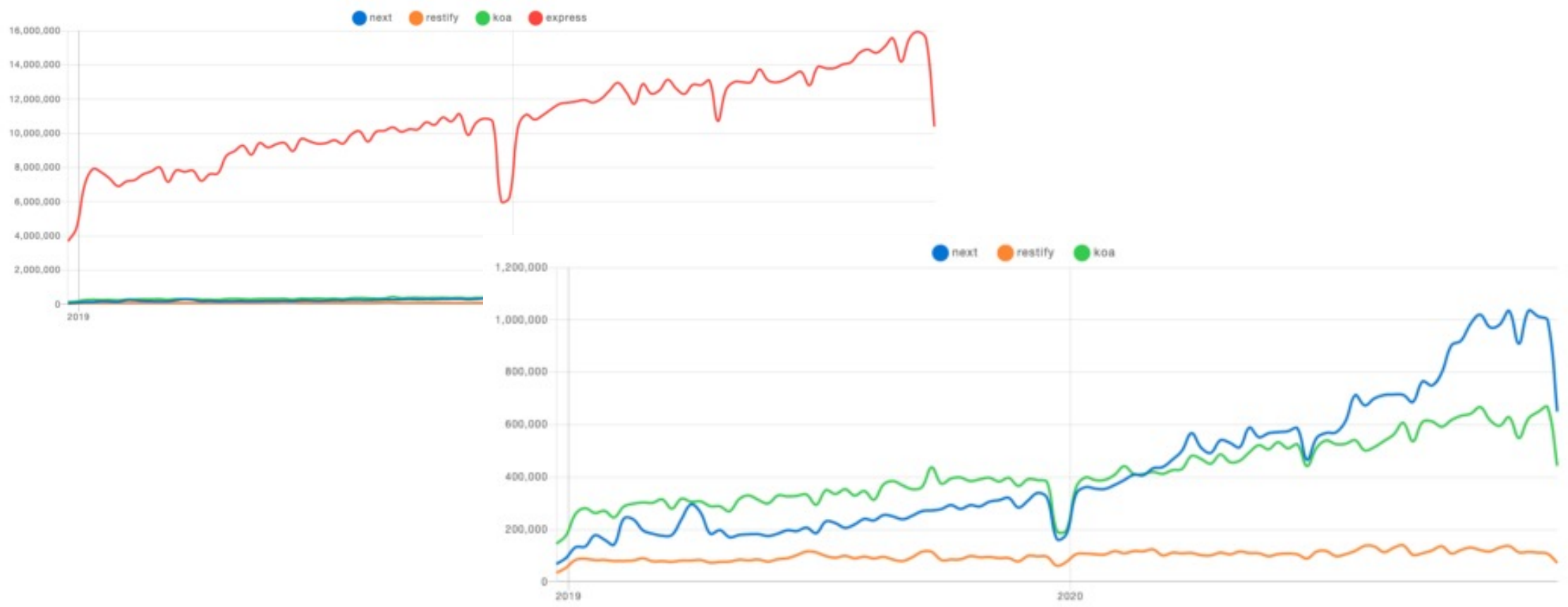
aws

JavaScript Frameworks and Trends 2021



https://www.npmtrends.com/@angular/core-vs-react-vs-vue-vs-angular

Server Side - JavaScript frameworks and Trends 2021

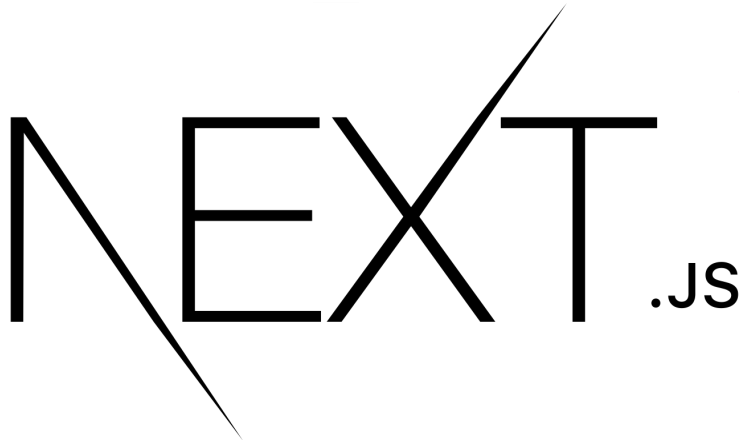


https://www.npmtrends.com/@angular/core-vs-react-vs-vue-vs-angular

NEXT.JS

"The React Framework for production"

- A framework created by Vercel
- Open source - MIT License (https://opensource.org/licenses/MIT)
- Based on **Node.js** and Babel
- NestJS is a framework that helps us to build efficient, scalable Node.JS server-side applications.
- NestJS uses progressive Javascript and uses a combination of OOP (Object Oriented Programming), FP (Functional Programming), and FRP (Functional React Programming).

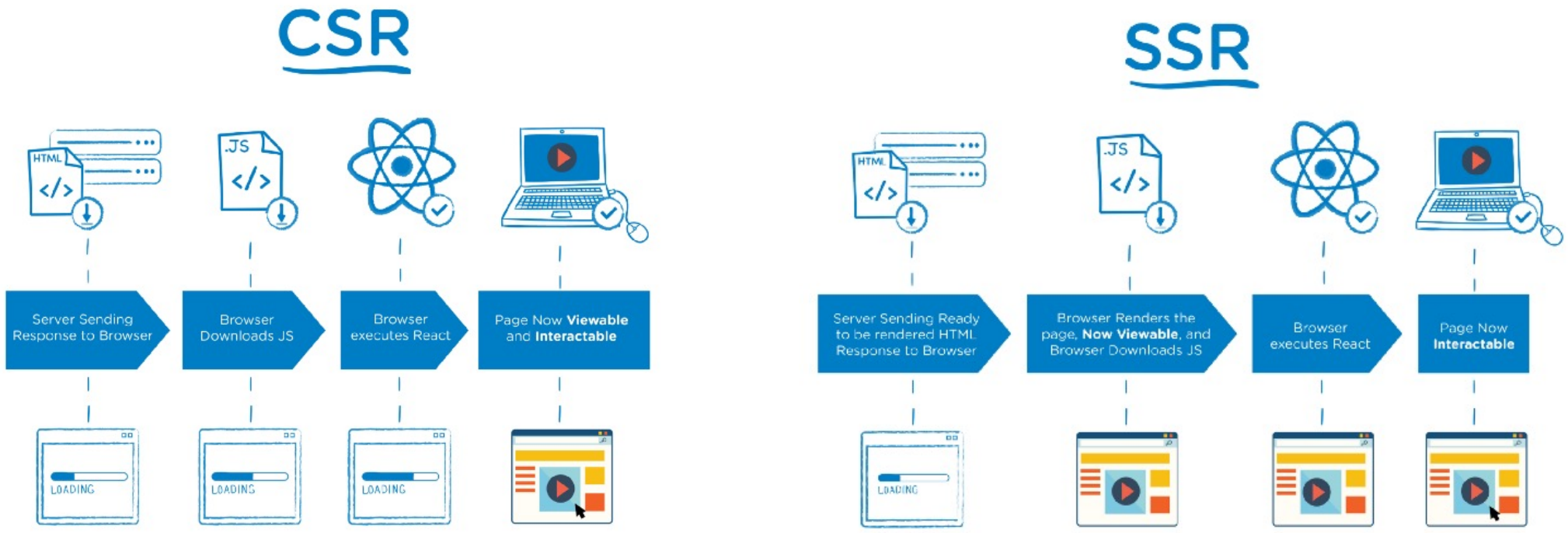


- https://nextjs.org/conf (Oct 26th)

https://nextjs.org/

aws

Client Side Rendering (CSR) and Server Side Rendering (SSR)

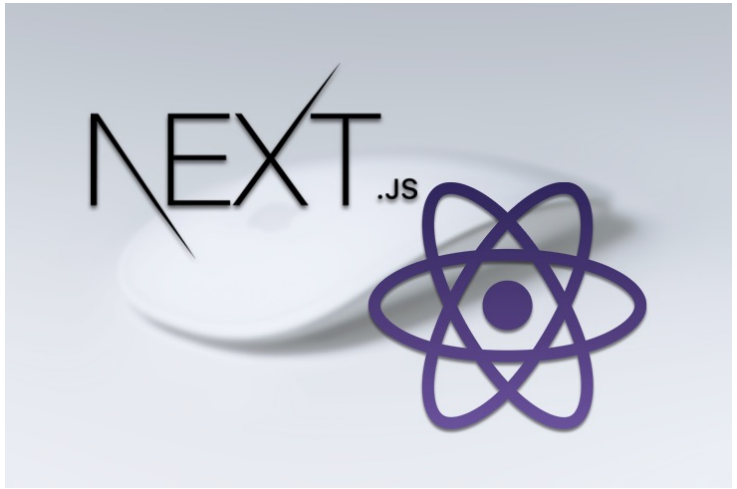


https://medium.com/walmartglobaltech/the-benefits-of-server-side-rendering-over-client-side-rendering-5d07ff2cefe8

aws

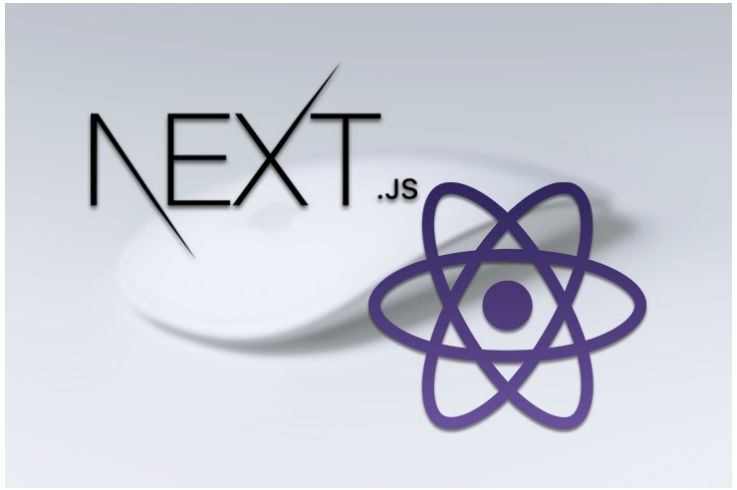
Next.JS vs React and Next.JS + React

- Both provide great opportunities for developers in their own ways
- Dependent on your requirements and needs
- Do your own research
- “Pick the right tool, for the right job"



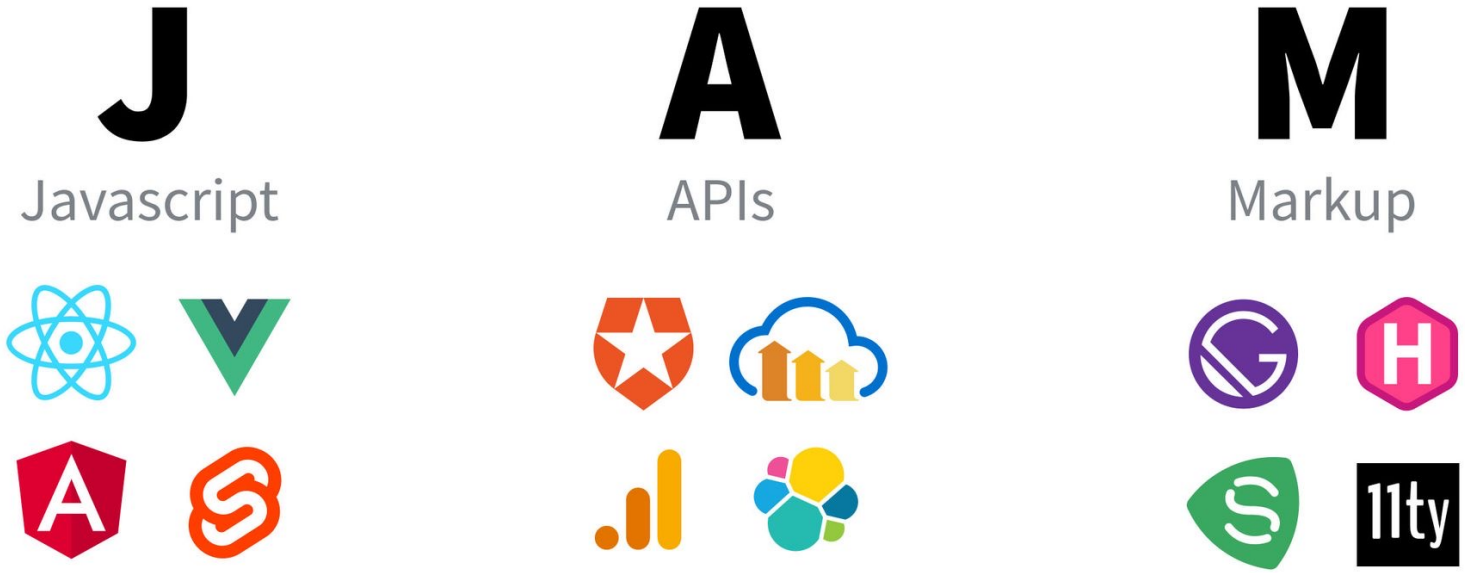
Next.JS and React

- https://blog.logrocket.com/next-js-vs-create-react-app/
- https://www.solutelabs.com/blog/nextjs-react-comparison
- https://www.byteant.com/blog/next-js-vs-gatsby-meeting-points-differences-and-the-best-ways-to-use/
- https://www.strivemindz.com/blog/why-should-you-move-to-next-js-from-react-js/
- https://medium.com/walmartglobaltech/the-benefits-of-server-side-rendering-over-client-side-rendering-5d07ff2cefe8



aws

JAM Stack

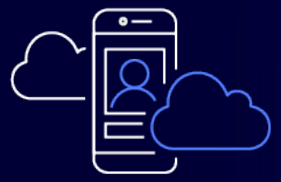


https://www.freecodecamp.org/news/what-is-the-jamstack-and-how-do-i-host-my-website-on-it/

aws

AWS Amplify

Users are accomplishing more virtually. How will your application stand out?



5B (+1.2B)

Mobile internet users by 2025

Source: GSMA

3.6 hours

Daily time on mobile per user

Source: Mary Meeker's Internet Trends 2019

24 GB (+17 GB)

Monthly mobile data usage per subscriber by 2024

Source: GSMA

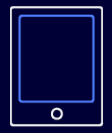
25%

Apps abandoned after first use

4M+

Apps published to global app stores you need to compete with for mindshare

To meet & exceed user expectations, the front-end developer assumes broad responsibility



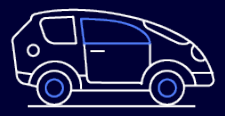
Platform(s)

Native mobile and web app?

Differentiation

Innovative & new ways to interact

How can your app stand out?



Screen and mobility

Responsive and progressive

Always available, even offline

Security

AuthN and AuthZ by default

Data layer

Instantaneous, with real-time feedback

Network friendly (fewer calls/more data/just right)

Time to market

Iterative, full-stack development

Developer productivity and team workflows

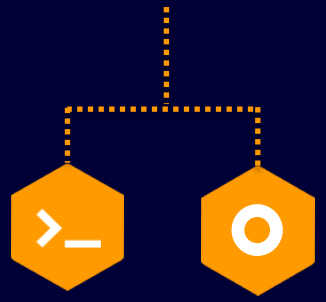
Ability for non-developers to manage the app

Amplify provides a full-stack developer experience across 10 feature categories

DEVELOP

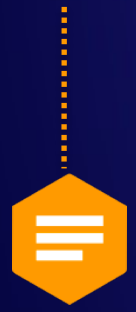
Configure AWS backends fast

Seamlessly connect front ends



CLI

Admin UI



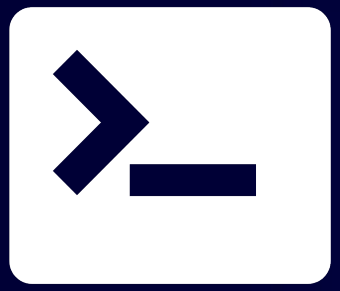
Libraries and UI components

FEATURE CATEGORIES

- Authentication
- DataStore
- Storage
- API (GraphQL & REST)
- Functions
- Analytics
- PubSub
- Predictions
- Interactions
- Notifications

Powered by AWS services, such as AWS AppSync (GraphQL API) and Amazon Cognito (authentication)

AWS Amplify CLI



CLI

Usage

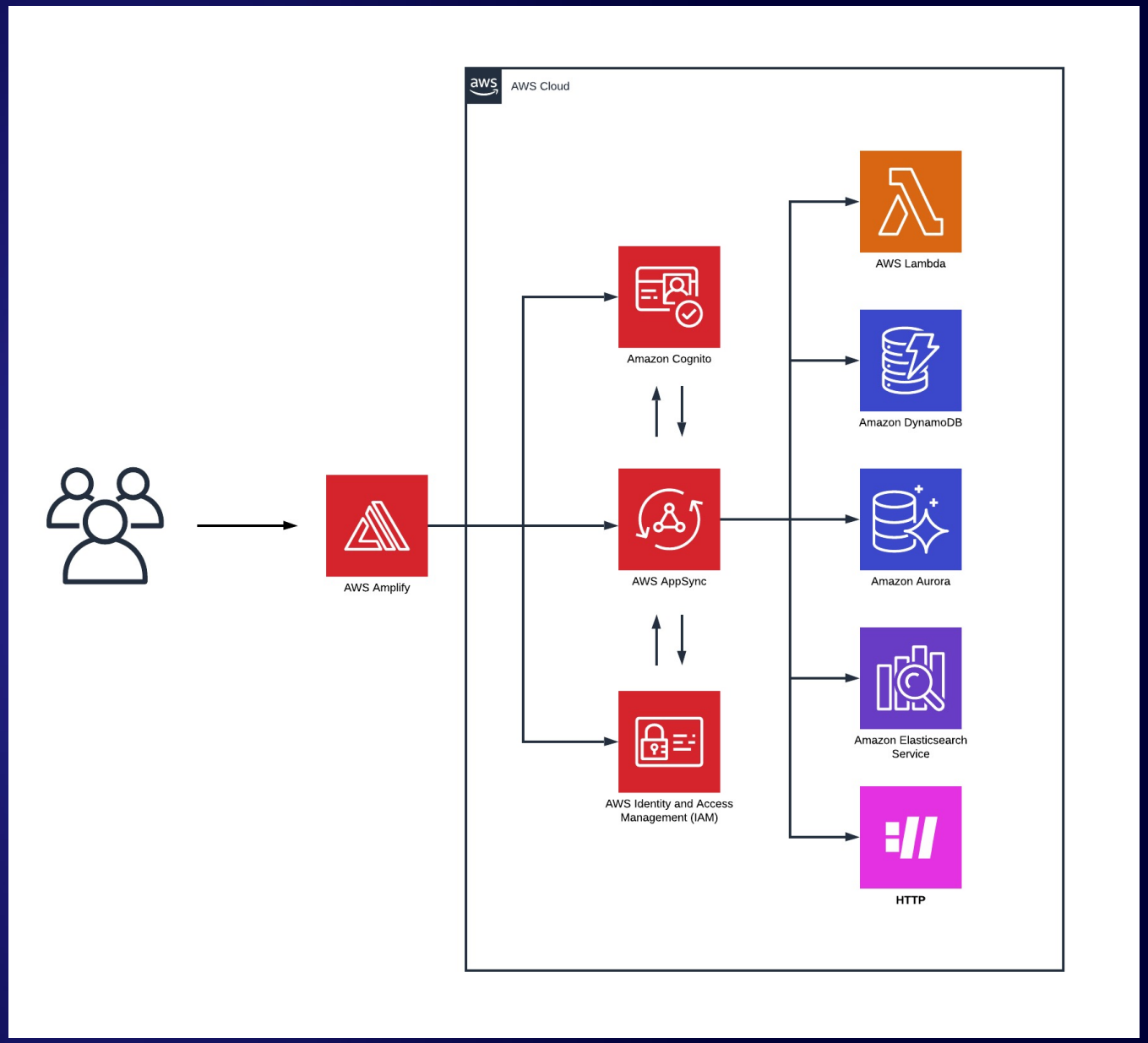
```
$ amplify init
$ amplify add api
$ amplify push
$ amplify update api
```

AWS Amplify CLI

AMPLIFY ADD API

API (GraphQL)

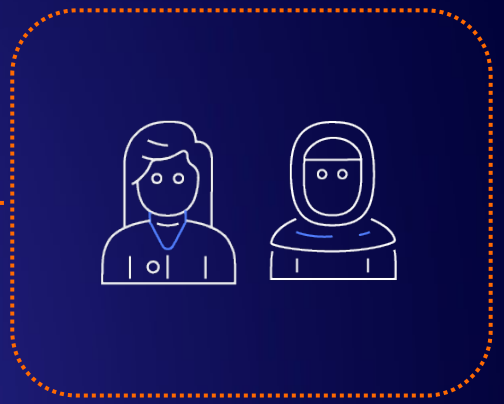
AWS AppSync
Amazon DynamoDB
AWS Lambda
Amazon Elasticsearch Service
Amazon Aurora Serverless



Create backend with the admin UI & CLI

Model data and relationships

Set up authentication and authorization



Team collaboration with or without AWS account

```
> curl -sL https://install.amplify.aws/cli
  | bash && $SHELL
> amplify add function
```

Extend with CLI toolchain to add functions, storage, and more

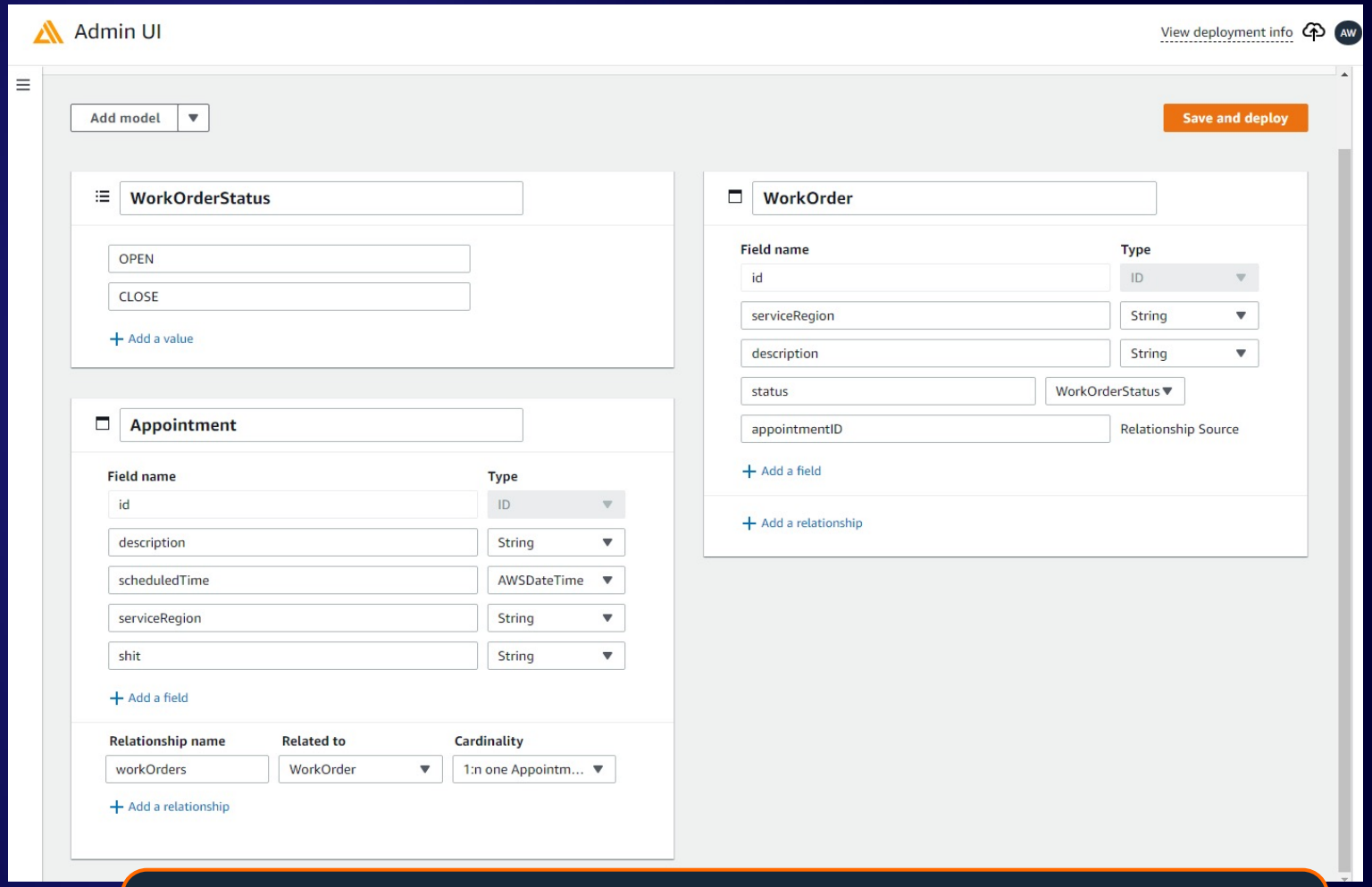
Example: Create backend for a field service app in the admin UI

Dispatcher

- Create appointments with many associated tasks
- Create and update data across all appointments

Agent

- Only see **my** appointments and tasks




```
> curl -sL https://install.amplify.aws/cli | bash && $SHELL
> amplify pull
```

Example: Create backend for a field service app in the CLI

Dispatcher

- Create appointments with many associated tasks
- Create and update data across all appointments

Agent

- Only see **my** appointments and tasks

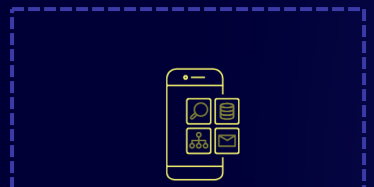
```
enum WorkOrderStatus {
  OPEN
  CLOSE
}

type WorkOrder
  @model
  @auth(rules: [{allow: public}])
  @key(name: "byAppointment", fields: ["appointmentID"]) {
  id: ID!
  serviceRegion: String!
  description: String!
  status: WorkOrderStatus!
  appointmentID: ID!
}

type Appointment @model @auth(rules: [{allow: public}]) {
  id: ID!
  description: String!
  scheduledTime: AWSDateTime
  serviceRegion: String!
  workOrders: [WorkOrder] @connection(keyName: "byAppointment", fields: ["id"])
}
```

```
> amplify add <functions | api | notifications | …>
```

Connect to your backend with UI components & libraries for popular mobile/web frameworks


Android


iOS


React Native

Ionic

NEW!


Flutter

JavaScript


React


Angular


Vue

Amplify client libraries

WEB

JS

Installation

```
$ npm install aws-amplify
```

JS

Usage

```
import { Storage } from 'aws-amplify';

const myImage = await Storage.get('my-image-key');
```

Example: Add sign-up/sign-in flows using the Amplify React UI component

Installation

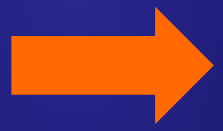
```
$ npm install @aws-amplify/ui-react
```

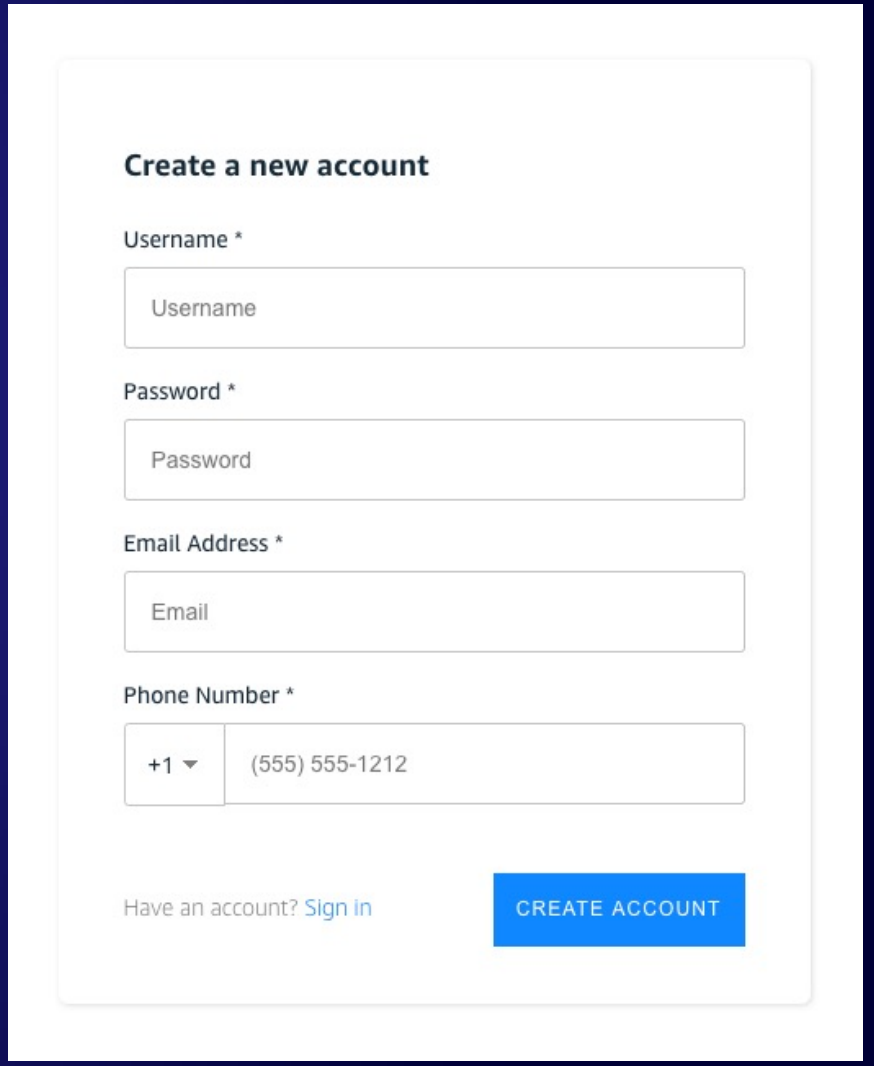
import

```
import {AmplifyAuthenticator, AmplifySignOut} from "@aws-amplify/ui-react";
```

Example: Add sign-up/sign-in flows using the Amplify React UI component

```
const App = () => (
  <AmplifyAuthenticator>
    <div>
      My App
      <AmplifySignOut />
    </div>
  </AmplifyAuthenticator>
);
```





AWS Amplify

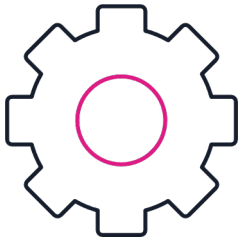
- https://aws.amazon.com/amplify/
- https://docs.amplify.aws/
- https://github.com/aws-amplify/amplify-js
- https://amplify.aws/community/resources

Workshop - Pathways

AWS Live Coding – Crash Course

Video with GitHub reference (30 minutes)

aws



Technical Depth

Level L200-L00+ (Intermediate)

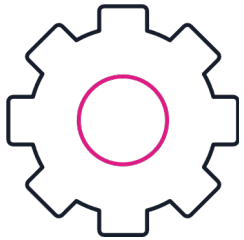
Delivery mode

Live coding (Video)

GitHub repository (Reference)

Time required

30 mins



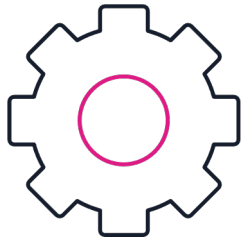
Suited to

Those who just want to watch and learn

Don’t want to code or code today

Don’t have developer permissions on laptop

Don’t have access to an AWS account



Environment and prerequisites

Internet access for Video (YouTube)

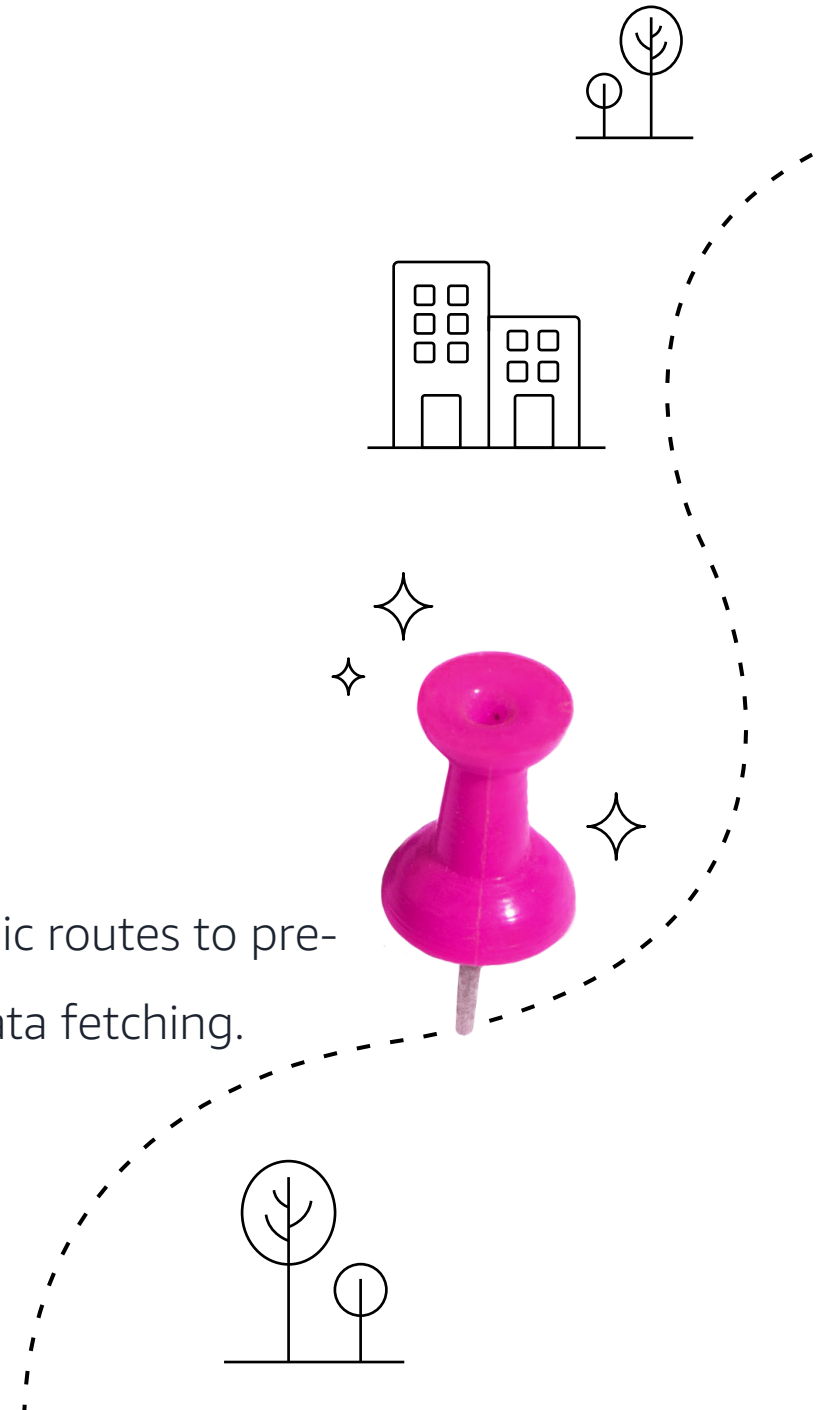
aws

Next.js with AWS Amplify Admin UI Crash Course

You'll learn how to

- Build a full stack Next.js app on AWS in less than 25 minutes.
    - Live Project : Blogging app

- You'll learn how to :
    - Deploy an API on AWS and connect a Next.js front end
    - Implementing markdown rendering, build time data fetching, dynamic routes to pre-render based on data, SSR, fallback routes, revalidation, and client data fetching.



aws

Next.js with AWS Amplify Admin UI Crash Course

- Live coding video (25 mins) : https://youtu.be/bQ1Giqn5G38
- GitHub : https://github.com/dabit3/next.js-amplify-datastore
- Check out the Amplify Admin UI sandbox here: https://sandbox.amplifyapp.com/

Quick link:

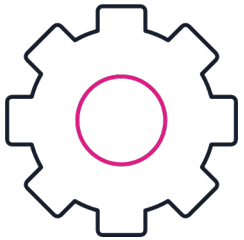
https://rebrand.ly/b85ub8s

aws

Advanced ~ Expert Level - GitHub Project

In this workshop you will learn how to build a full stack cloud application with Next.js, Tailwind, & AWS Amplify.

aws



Technical Depth

Level L300/400+
(Advanced/Expert)

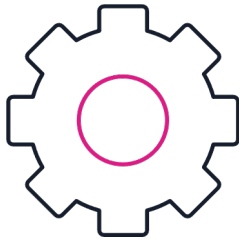
Delivery mode

Self-paced

GitHub repository

Time required

2-4 hours



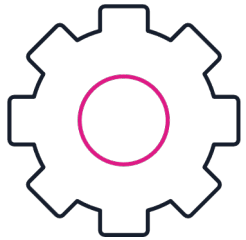
Suited to

Developers, who have advanced knowledge of

Git experience

Knowledge of common dev tooling e.g. npm, yarn

Good with debugging



Environment and prerequisites

Workstation permissions (developer)

Ability to install dependencies
- Node.js v10.x or later
- npm v5.x or later
- git v2.14.1 or later

(Existing) Access to an AWS Account

aws

Workshop / App overview

In this workshop you will learn how to build a full stack cloud application with Next.js, Tailwind, & AWS Amplify.

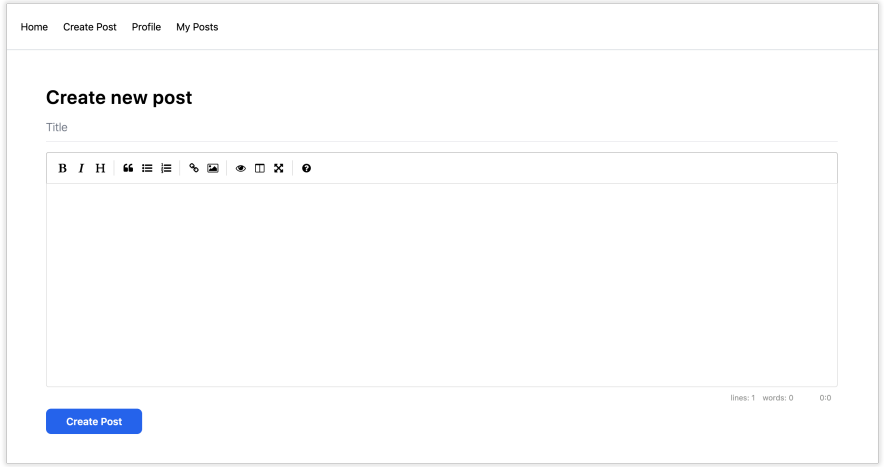
We'll start from scratch, creating a new Next.js app. We'll then, step by step, use the Amplify CLI to build out and configure our cloud infrastructure and then use the Amplify JS Libraries to connect the Next.js app to the APIs we create using the CLI.

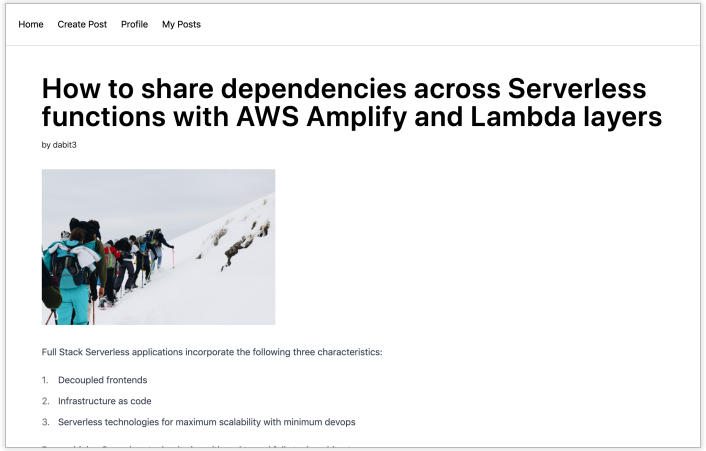
- The app will be a multi-user blogging platform with a markdown editor.

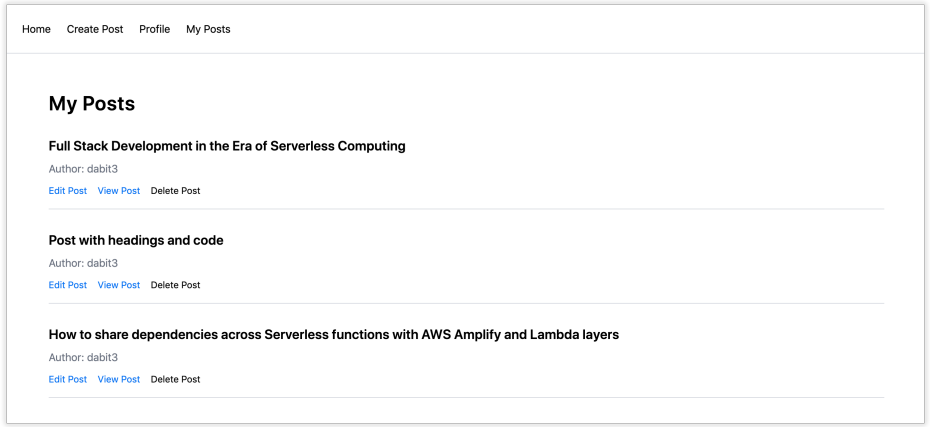
When you think of many types of applications like Instagram, Twitter, or Facebook, they consist of a list of items and often the ability to drill down into a single item view. The app we will be building will be very similar to this, displaying a list of posts with data like the title, content, and author of the post.

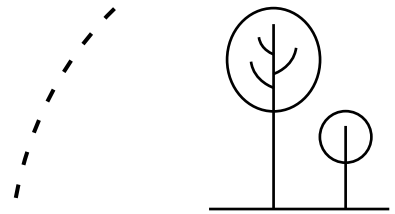
aws

Outcome of workshop (Live app)









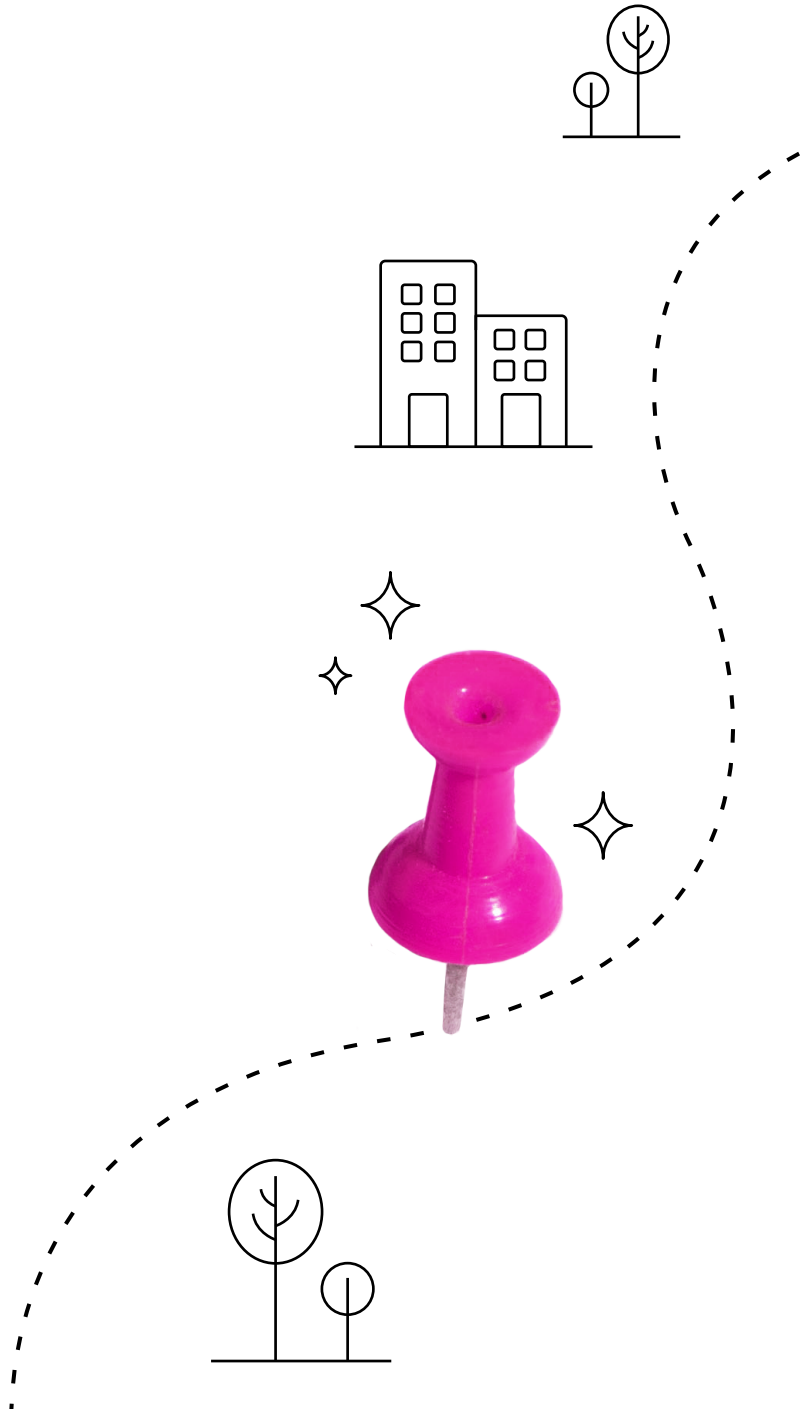
Advanced ~ Expert GitHub project - Next.js, Tailwind, & AWS Amplify

- GitHub : https://github.com/dabit3/next.js-amplify-workshop

- Live coding video (1hr 20 mins) : https://www.youtube.com/watch?v=13nYLmjZ0Ys&t=1274s

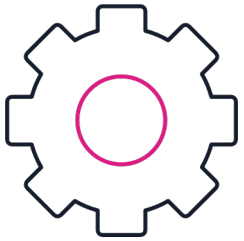
Quick link:

https://rebrand.ly/ppihxsh



AWS Workshop (Amplify)

This hands-on lab will build a full-stack serverless application on AWS using Amplify, Next.js, GraphQL



Technical Depth

Level 200-300+ (Intermediate)

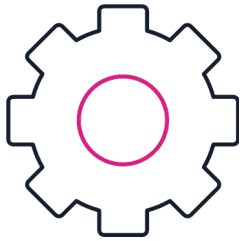
Delivery mode

Self-paced

Lab instructions

Time required

2-4 hours

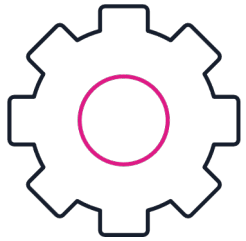


Suited to

Developers, who have advanced knowledge of :

Git experience

Awareness of common developer tooling



Environment and prerequisites

Workstation permissions (developer)

Ability to install dependencies
- Node.js v10.x or later
- npm v5.x or later
- git v2.14.1 or later

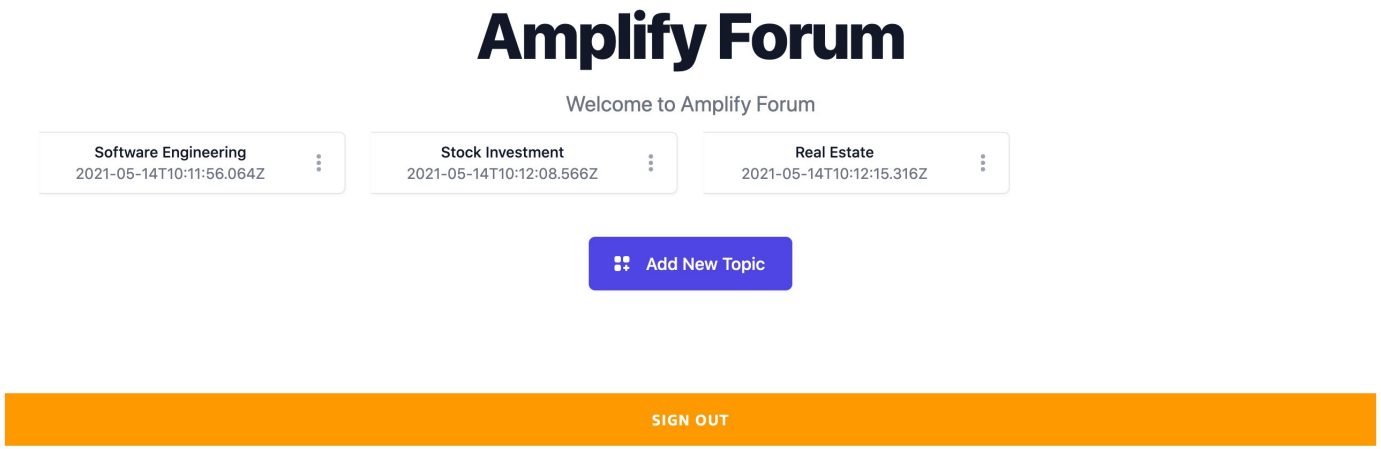
(Existing) Access to an AWS Account

OR : AWS Event / AWS Lab Environment

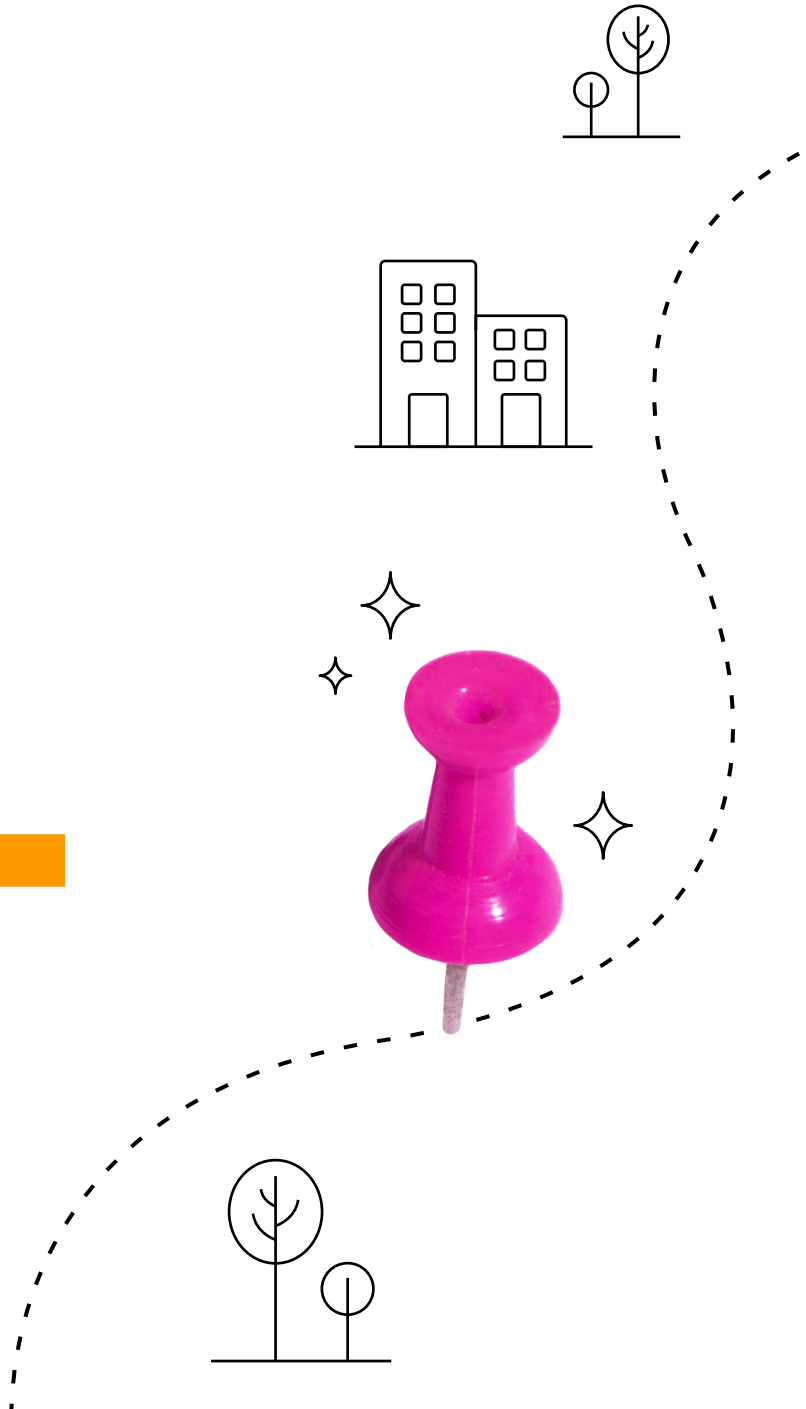
Workshop / App overview

- This hands-on lab will build a full-stack serverless application on AWS using Amplify, Next.js, GraphQL
- Application we will build is going to be a Reddit-like message forum.
- Once users follow this guide, they will have a working application running on AWS.

Outcome of workshop (Live app)



Working demo: https://dev.d2j3zmrl7b2iga.amplifyapp.com



Next.js with AWS Amplify Intermediate (workshop)

Workshop :

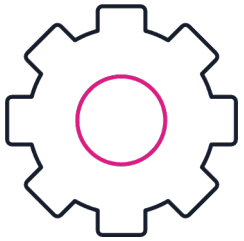
https://build-message-forum-with-amplify.workshop.aws/en/

Quick link:

https://rebrand.ly/zuyxyuz

aws

Future event



Technical Depth

Level 200-300+ (Intermediate)

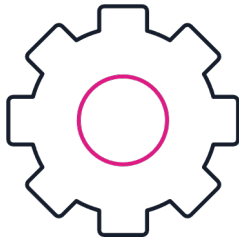
Delivery mode

Self-paced

Lab instructions

Time required

2-4 hours

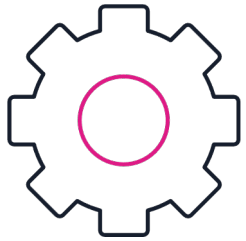


Suited to

Developers, who have advanced knowledge of :

Git experience

Awareness of common developer tooling



Environment and prerequisites

Workstation permissions (developer)

Ability to install dependencies
- Node.js v10.x or later
- npm v5.x or later
- git v2.14.1 or later

(Existing) Access to an AWS Account

OR : AWS Event / AWS Lab Environment

I would like to complete a detailed hands on workshop

- Based on demand and interest. Will run a future 0.5 day event (3-4 hours)
- Will include AWS Lab Accounts for registered attendees
- Date TBD : *Approx. targeted 2-4 weeks*
- Register interest :

  https://rebrand.ly/kft8or2

Future event

Learning pathways

Option A: Watch and learn

- 30 mins; YouTube video
- https://rebrand.ly/7mperhq

Option B: Don't have an existing AWS account and/or developer workstation

- Interested in a future ½ day event
- https://rebrand.ly/kft8or2

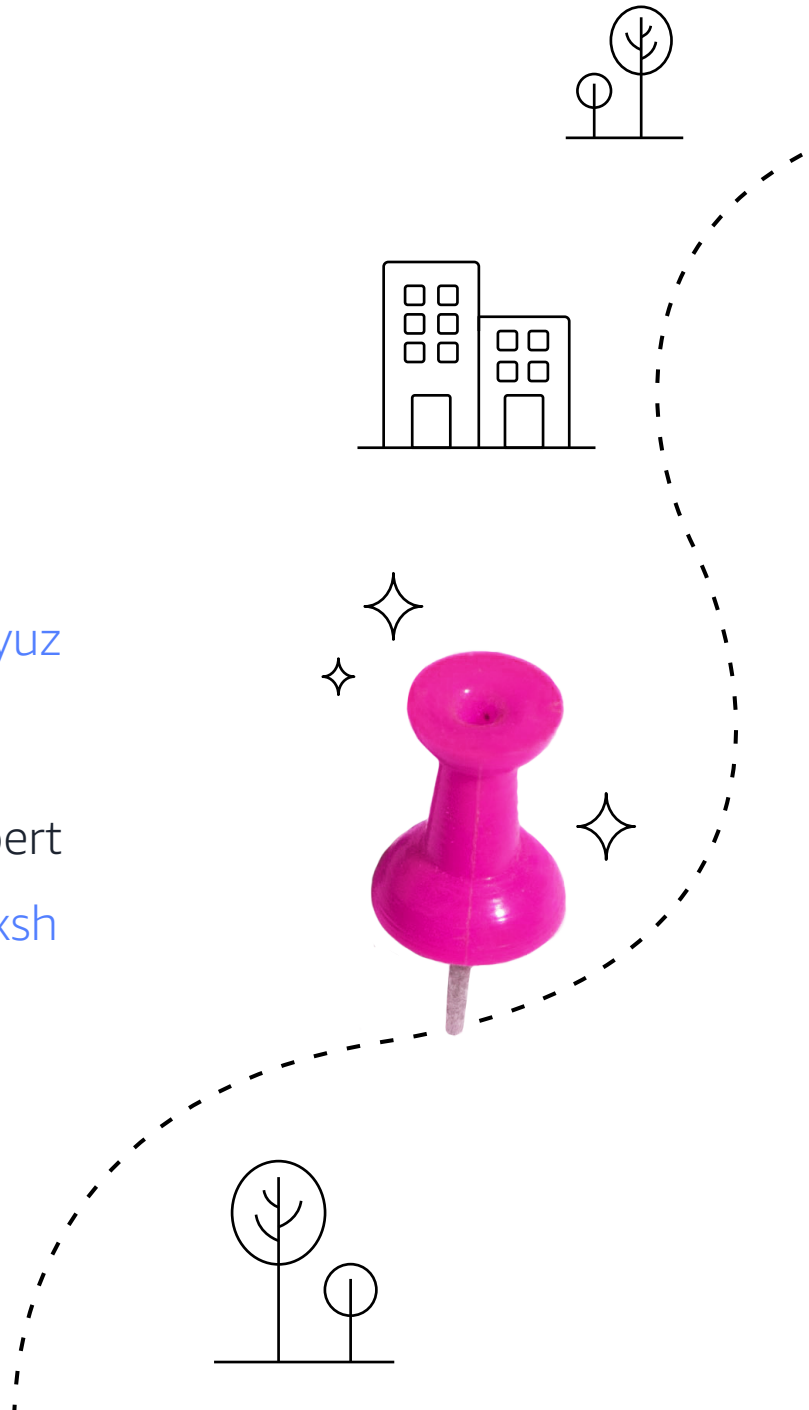
Have an existing AWS account

- 2-4 hours. Self paced
  - Option C: Intermediate

    https://rebrand.ly/zuyxyuz

- Option D: Advanced/Expert

  https://rebrand.ly/ppihxsh



Thank you

Chris Modica
Principal SA, ANZ

@chris_modica_

ln/christophermodica

cmmodica@amazon.com