AWS re:Invent

DOP315-R1

Build using JavaScript with AWS Amplify, AWS Lambda, and AWS Fargate

Trivikram Kamat

Software Development Engineer,
AWS SDKs and Tools
Amazon Web Services

Vinod Dinakaran

Software Development Manager,
AWS SDKs and Tools
Amazon Web Services

AWS re:Invent

aws

What are our names again?

Tree + Weak + Rum

Tri + vik + ram

What are our names again?

V8 + Node.js

Vi + nod

Agenda

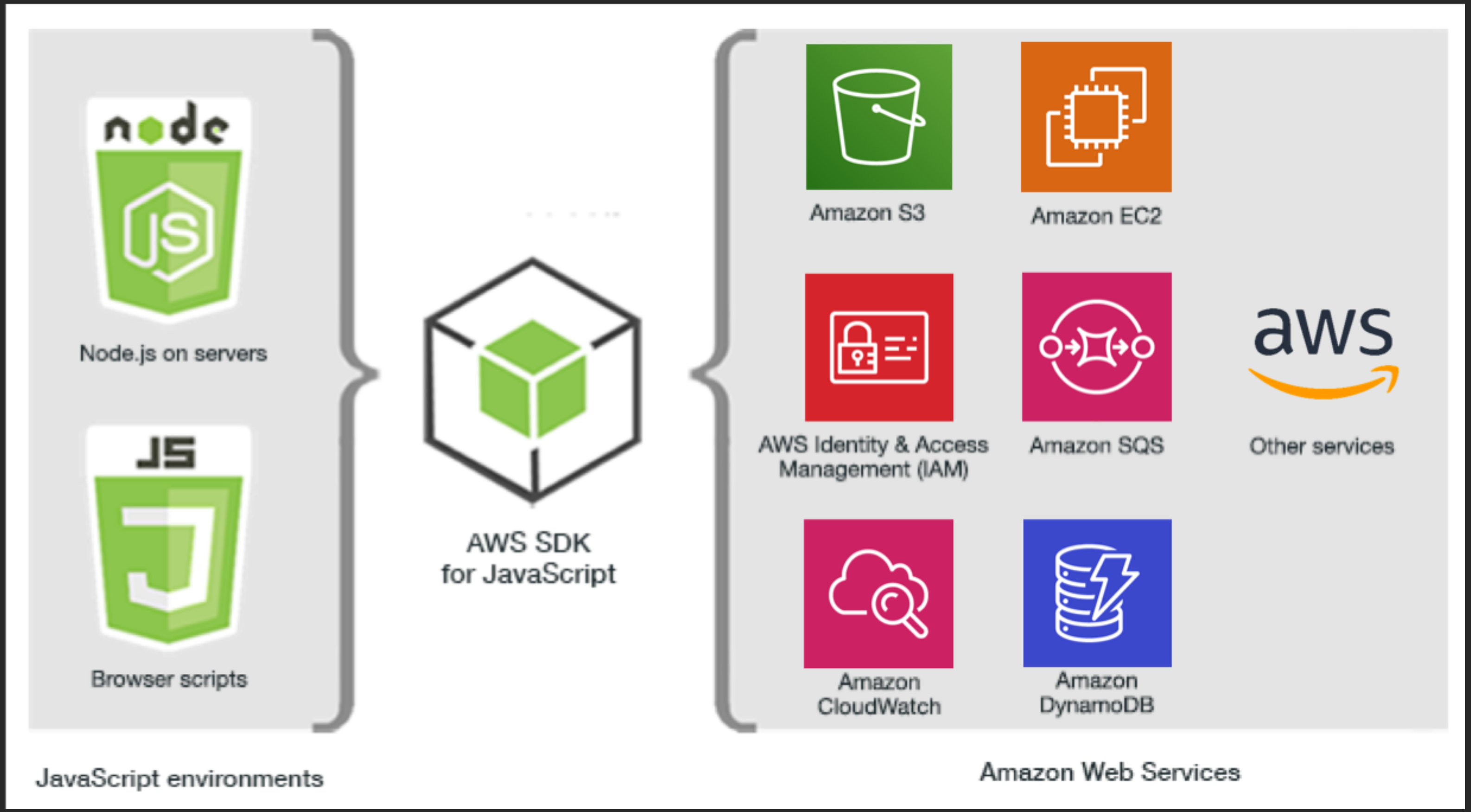
- What is AWS SDK for JavaScript?
- How to use AWS Amplify to build a modern frontend
- How to use JavaScript in AWS Lambda to build a clickstream ingestion pipeline
- How to use JavaScript SDK in containers (e.g., AWS Fargate) to build a service backend
- Compare AWS JS SDK v2 to the v3 dev-preview

What is AWS SDK for JavaScript?

It provides a JavaScript API for AWS services

It helps you build libraries or apps in:

- Modern browsers
- Node.js
- Electron (desktop)
- React Native (mobile)



Example applications

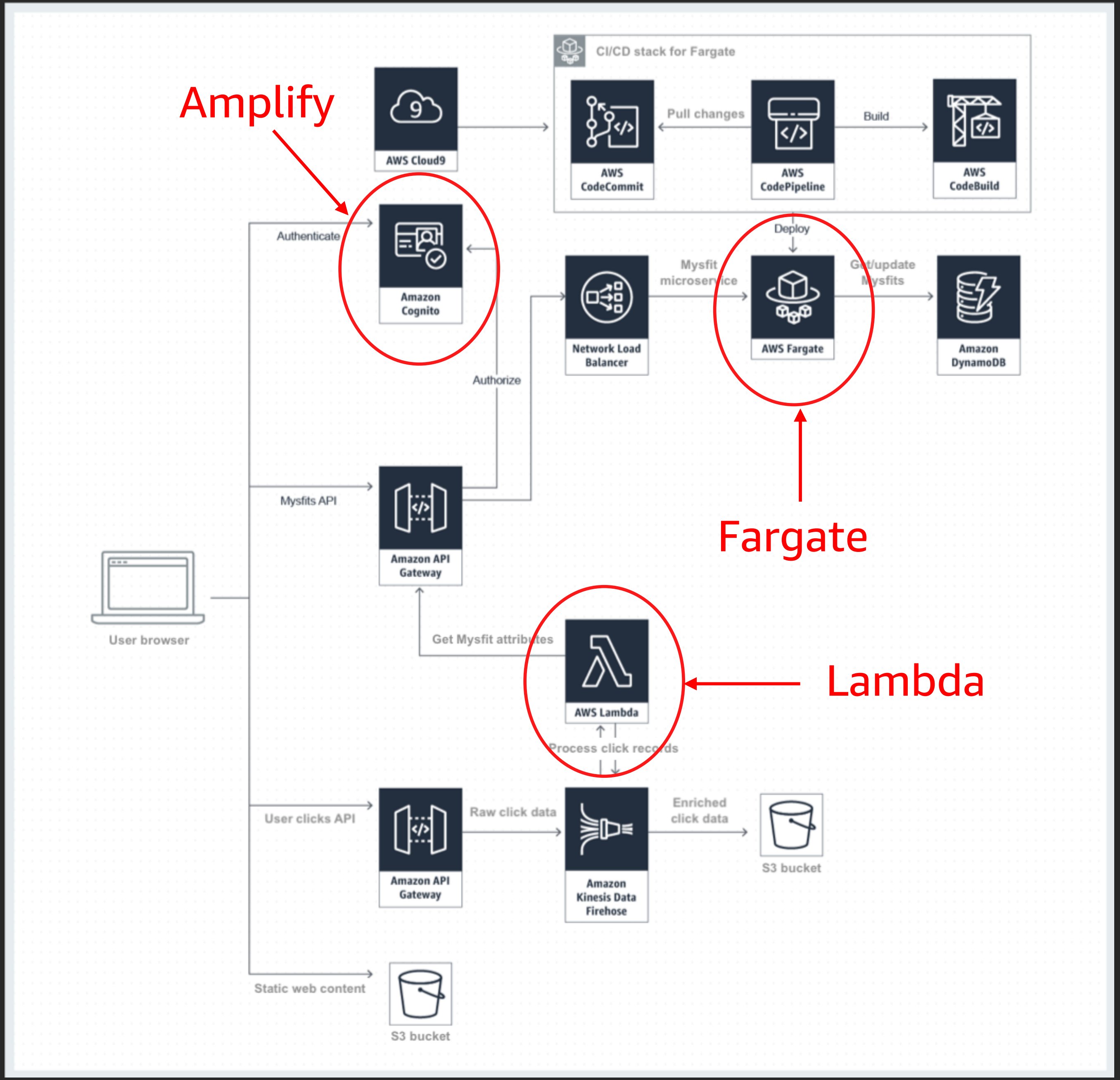
- Mythical Mysfits: An AWS sample application that mirrors common business use cases

https://www.mythicalmysfits.com

- AWS JS SDK v3 Workshop: Todomvc built using v2 and v3 of the SDK, helps compare usability and performance

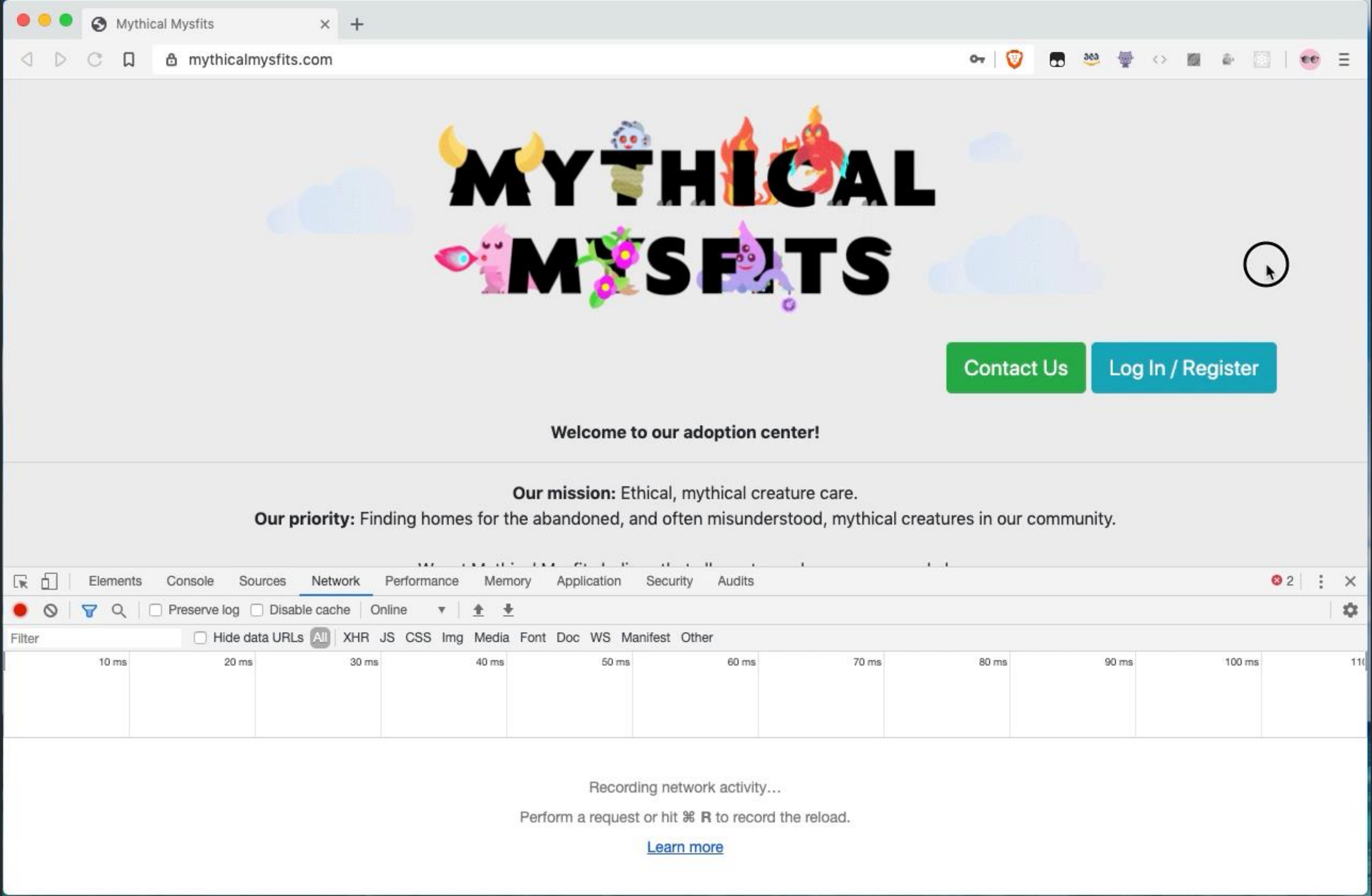
https://github.com/aws-samples/aws-sdk-js-v3-workshop

Mythical Mysfits architecture diagram



- Build a modern frontend
- Build a clickstream ingestion pipeline
- Build a search microservice

Build a modern frontend



Welcome to our adoption center!

Our mission: Ethical, mythical creature care.
Our priority: Finding homes for the abandoned, and often misunderstood, mythical creatures in our community.

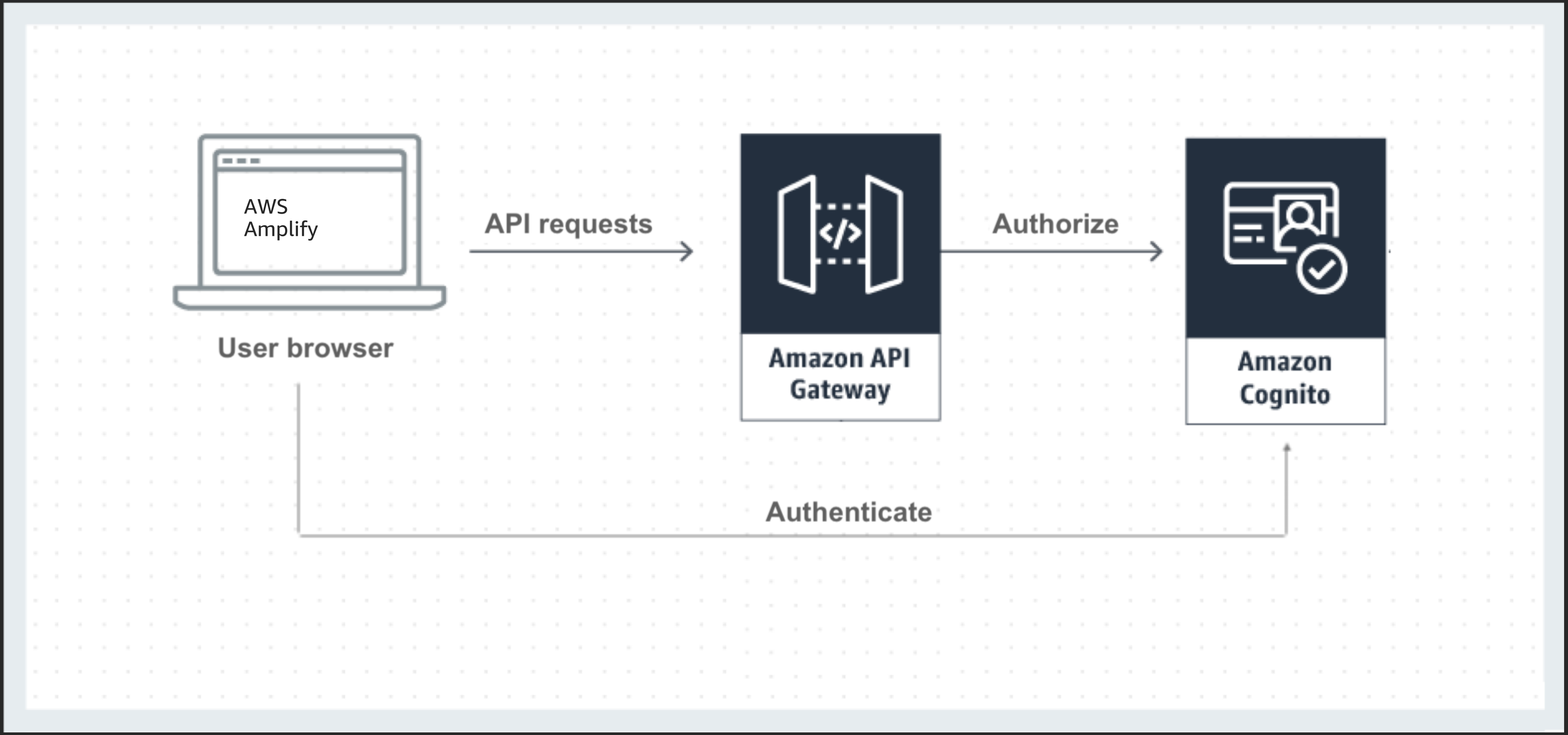
Recording network activity…

Perform a request or hit ⌘ R to record the reload.

AWS Amplify

- What is AWS Amplify?
  - AWS Amplify makes it easy to create, configure, and implement scalable mobile and web apps powered by AWS
- Where do we use Amplify?
  - Amplify is a framework that uses AWS SDK for JavaScript to help you quickly set up authentication, analytics, and offline data sync for your mobile applications
  - We use Amplify in the frontend for registrations and login

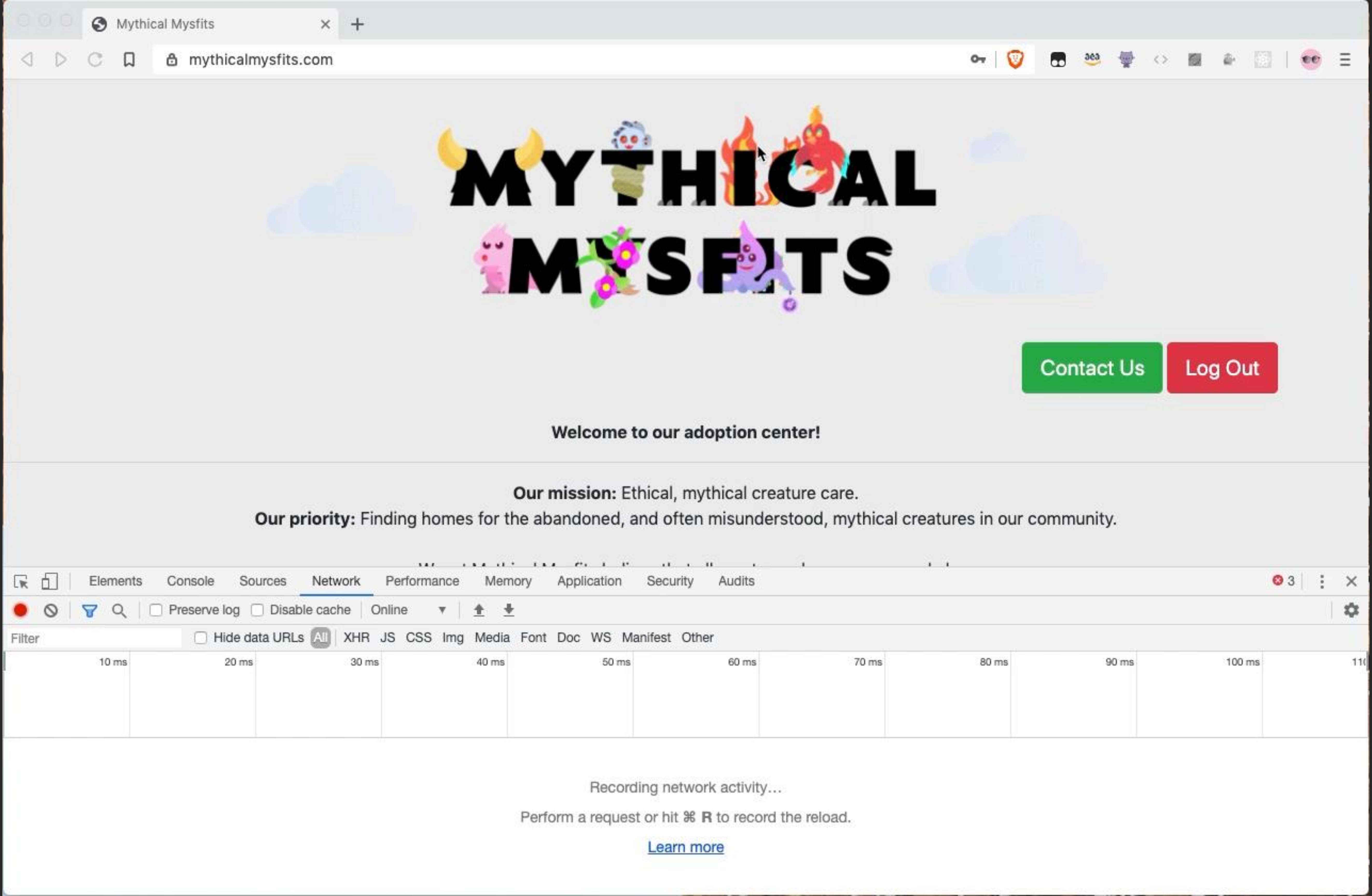
Architecture diagram



Sample code for signing in user

```
// Frontend code in React/Angular/Vue/other components

import { Auth, configure } from "aws-amplify";

configure({
  region: COGNITO_REGION,
  userPoolId: COGNITO_USER_POOL_ID,
  identityPoolId: COGNITO_IDENTITY_POOL_ID
});

const signInUser = (username, password) => {
  try {
    await Auth.signIn(username, password);
    // User successfully signed in
  } catch(e) {
    // Error during sign in
  }
}
```

Build a clickstream ingestion pipeline



Welcome to our adoption center!

Our mission: Ethical, mythical creature care.
Our priority: Finding homes for the abandoned, and often misunderstood, mythical creatures in our community.

Recording network activity…

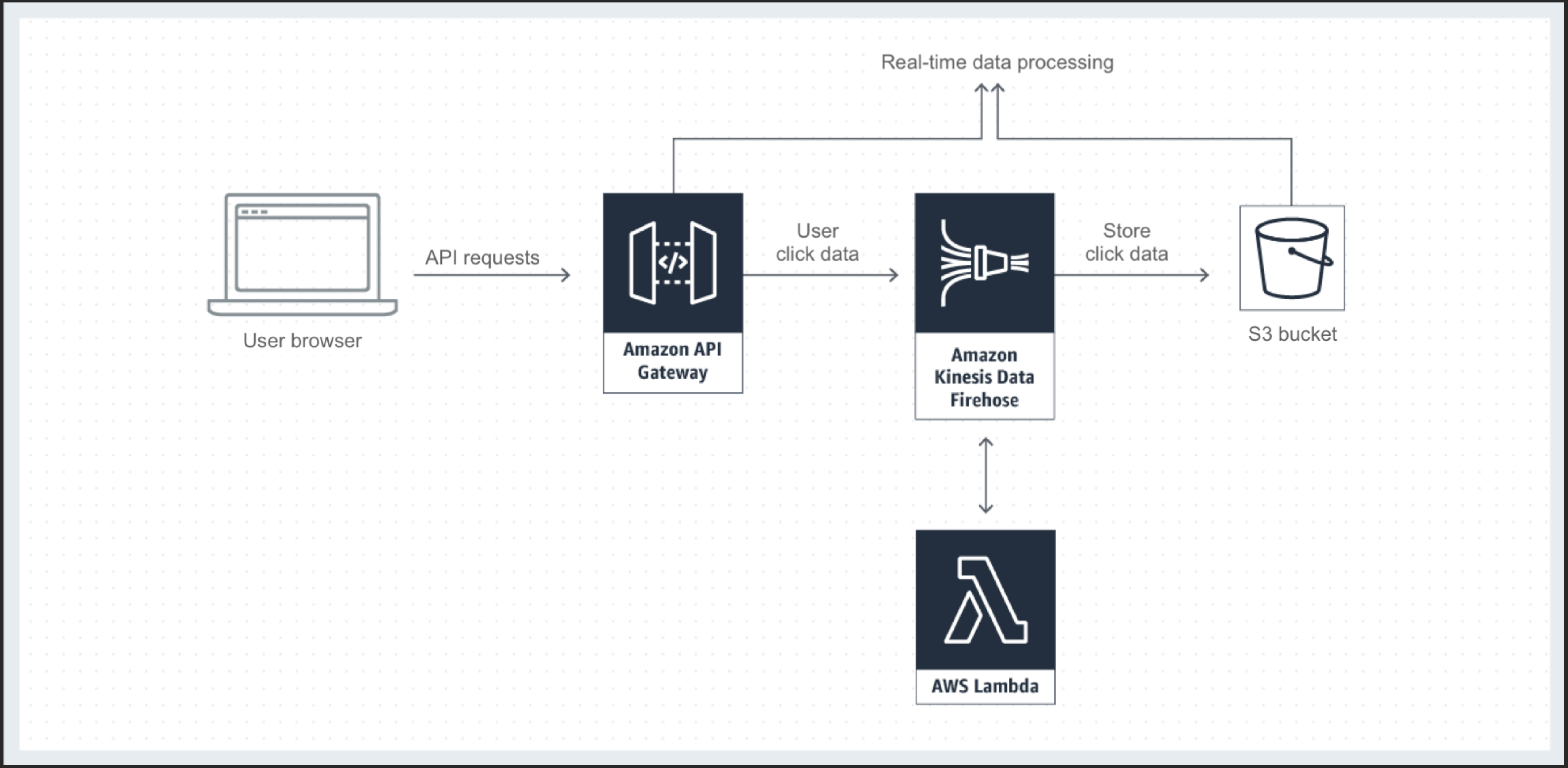
Perform a request or hit ⌘ **R** to record the reload.

Learn more

AWS Lambda

- What is AWS Lambda?
  - AWS Lambda lets you run code without provisioning or managing servers
  - You pay only for the compute time you consume; there is no charge when your code is not running
  - Just upload your code, and Lambda takes care of everything required to run and scale your code with high availability
- Where do we use Lambda?
  - Lambda is great for event-driven applications that need to respond in real time to changes in data, shifts in system state, or actions by users
  - We use Lambda for processing user clicks on Mysfits

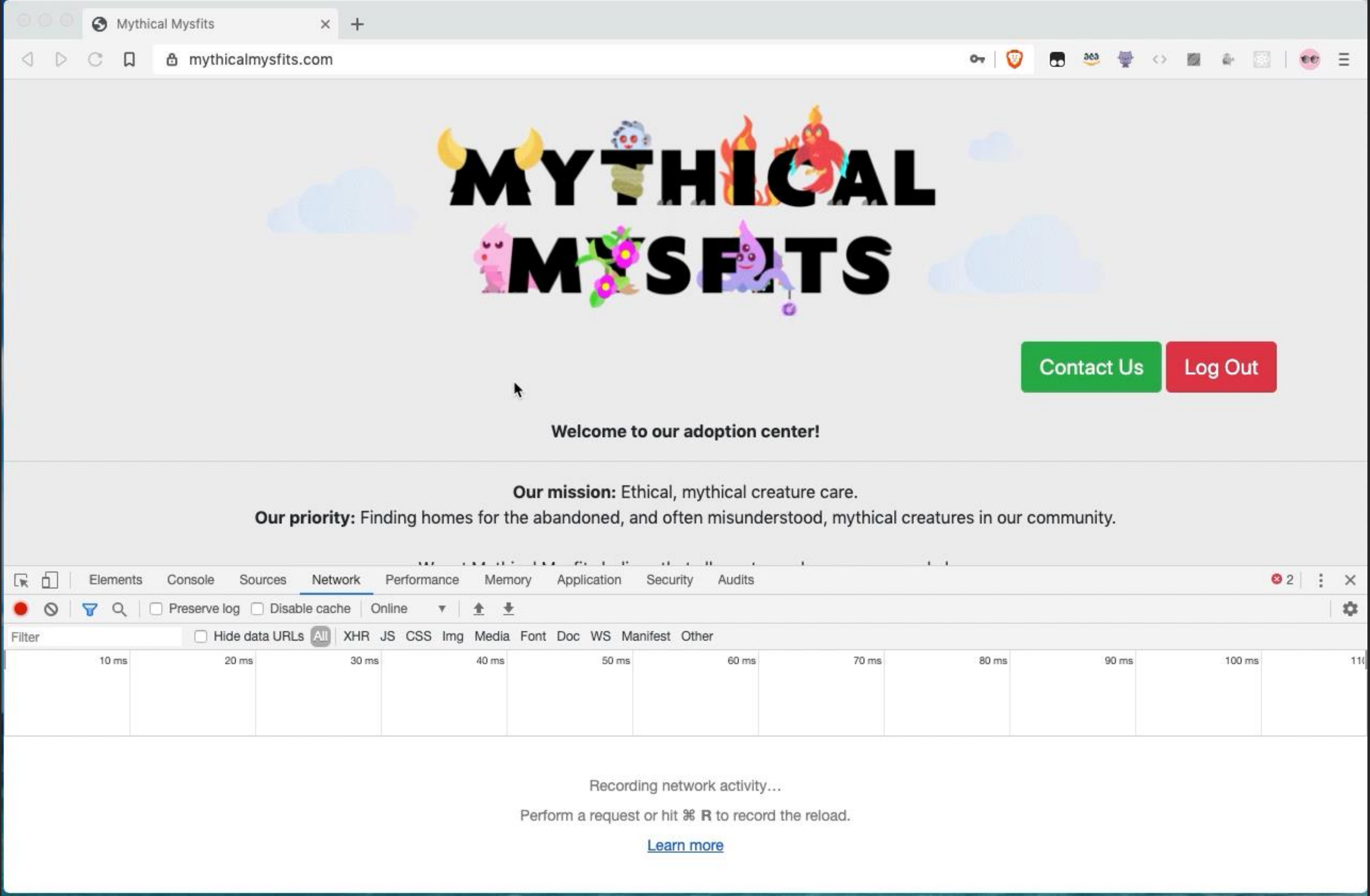
Architecture diagram



Sample code for processing clicks

```
const processRecord = async (event) => {
  let output = [];

  // retrieve the list of records and loop through them
  for (let record in event.records) {
    const enrichedClick = {
              'userId': click['userId'],
              'mysfitId': mysfitId,
              //other data from click here
        }

    output.push(enrichedClick);
  }

  return output;
};

export processRecord;
```

Build a search microservice



Welcome to our adoption center!

Our mission: Ethical, mythical creature care.
Our priority: Finding homes for the abandoned, and often misunderstood, mythical creatures in our community.

Recording network activity…

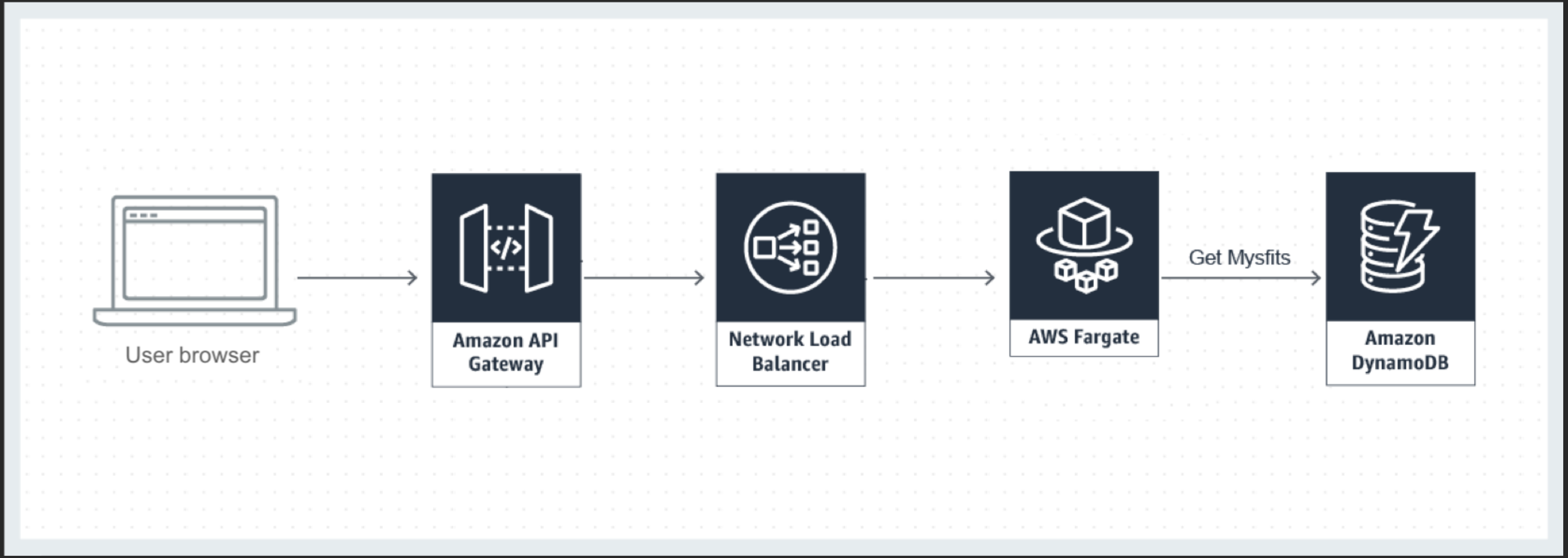
Perform a request or hit ⌘ R to record the reload.

Learn more

AWS Fargate

- What is AWS Fargate?
  - AWS Fargate is a compute engine for deploying and managing containers, which frees you from having to manage any of the underlying infrastructure
  - With AWS Fargate, you no longer have to provision, configure, and scale clusters of virtual machines to run containers
- Where do we use Fargate?
  - It’s a great choice for building long-running processes, such as microservices backends for web, mobile, and PaaS platforms
  - With Fargate, you get the control of containers and the flexibility to choose when they run without worrying about provisioning or scaling servers
  - We deploy our backend returns information about Mysfits on AWS Fargate

Architecture diagram



/mysfits?filter=GoodEvil&value=Good

$\downarrow$

queryMysfitItems(“GoodEvil”, “Good”)

Sample code to query Mysfits (v2)

```
import AWS from "aws-sdk";
import express from "express";

const app = express();
const client = new AWS.DynamoDB();

app.get("/mysfits?filter=:filter&value=:value", (req, res) => {
  const { filter, value } = req.params;
  const params = getQueryParams(filter, value);
  const result = await client.query(params).promise();
  return res.send(result.Items);
});
```

Sample code to query Mysfits (v3 dev-preview)

```
import { DynamoDBClient, QueryCommand } from "@aws-sdk/client-dynamodb";
import express from "express";

const app = express();
const client = new DynamoDBClient();

app.get("/mysfits?filter=:filter&value=:value", (req, res) => {
  const { filter, value } = req.params;
  const params = getQueryParams(filter, value);
  const result = await client.send(new QueryCommand(params));
  return res.send(result.Items);
});
```

AWS JavaScript SDK v2 vs v3

The v3 SDK:

- Is modular
- Has cold/warm start improvements
- Has customizable middleware
- Is typescript-based

	Bundle size	AWS Lambda cold start (90th)	AWS Lambda warm start (90th)
AWS JS SDK v2	~470 KB	1.2 s	139 ms
AWS JS SDK v3 - DynamoDB Client	~76 KB		
AWS JS SDK v3 - DynamoDB Command	~26 KB	776 ms	136 ms

Recap

- Overview of AWS SDK for JavaScript
- AWS Amplify to build a modern frontend
- JavaScript in AWS Lambda to build a clickstream ingestion pipeline
- JavaScript SDK in containers (e.g., AWS Fargate) to build a service backend
- Compare AWS JS SDK v2 to the v3 dev-preview

Demo

Q&A

Starter questions/topics

1. Got a JavaScript problem?
2. When to use Serverless vs containers?
3. AWS JS SDK v2 vs v3

Resources

- AWS JS SDK v3 https://github.com/aws/aws-sdk-js-v3
- AWS JS SDK v3 workshop https://github.com/aws-samples/aws-sdk-js-v3-workshop
- Amplify https://aws.amazon.com/amplify/
- AWS Lambda https://aws.amazon.com/lambda/
- AWS Fargate https://aws.amazon.com/fargate/
- Mythical Mysfits https://mythicalmysfits.com/

Related talks

- Amplify
  - [MOB303] Build and ship full-stack serverless apps with AWS Amplify
  - [DOP334] Set up a serverless app using React and AWS Amplify
- Lambda
  - [SVS343] Building microservices with AWS Lambda
  - [DAT306] Implement microservice architectures with Amazon DynamoDB & AWS Lambda
  - [DAT335] Build serverless applications with Amazon DynamoDB and AWS Lambda
  - [SVS322] Best practices for CI/CD with AWS Lambda and Amazon API Gateway
- Fargate
  - [CON208] Build your microservices application on AWS Fargate

Learn DevOps with AWS Training and Certification

Resources created by the experts at AWS to propel your organization and career forward

Take free digital training to learn best practices for developing, deploying, and maintaining applications

Classroom offerings, like DevOps Engineering on AWS, feature AWS expert instructors and hands-on activities

Validate expertise with the **AWS Certified DevOps Engineer - Professional** or **AWS Certified Developer - Associate** exams

Visit aws.amazon.com/training/path-developing/

aws training and certification

Thank you!

Trivikram Kamat
trivikr@amazon.com

Vinod Dinakaran
vinoddin@amazon.com

Please complete the session survey in the mobile app.