aws modern apps

Building an End-to-End Serverless Web Application

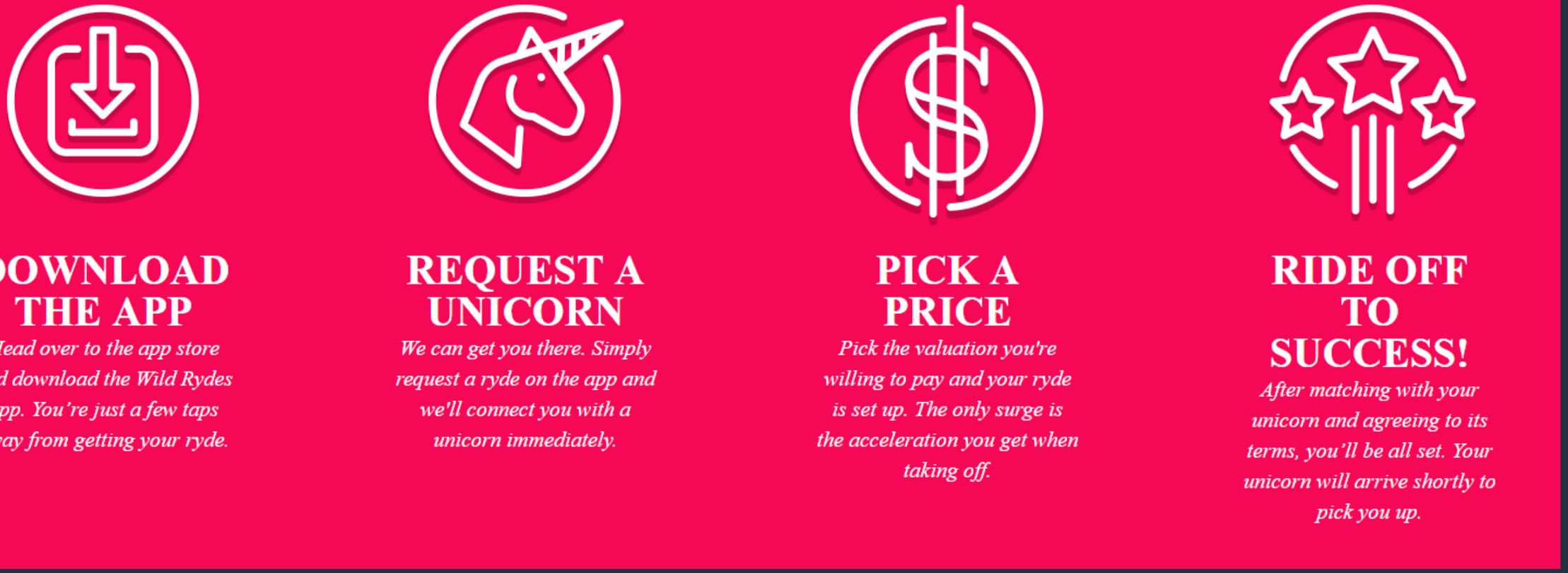
Chris McPeek, Sr. Solutions Architect
Uma Ramadoss, Serverless Solutions Architect
Veda Raman, Serverless Solutions Architect

Scenario: Wild Rydes (www.wildrydes.com)

aws modern apps

Help Wild Rydes disrupt transportation!

So how does this magic work?



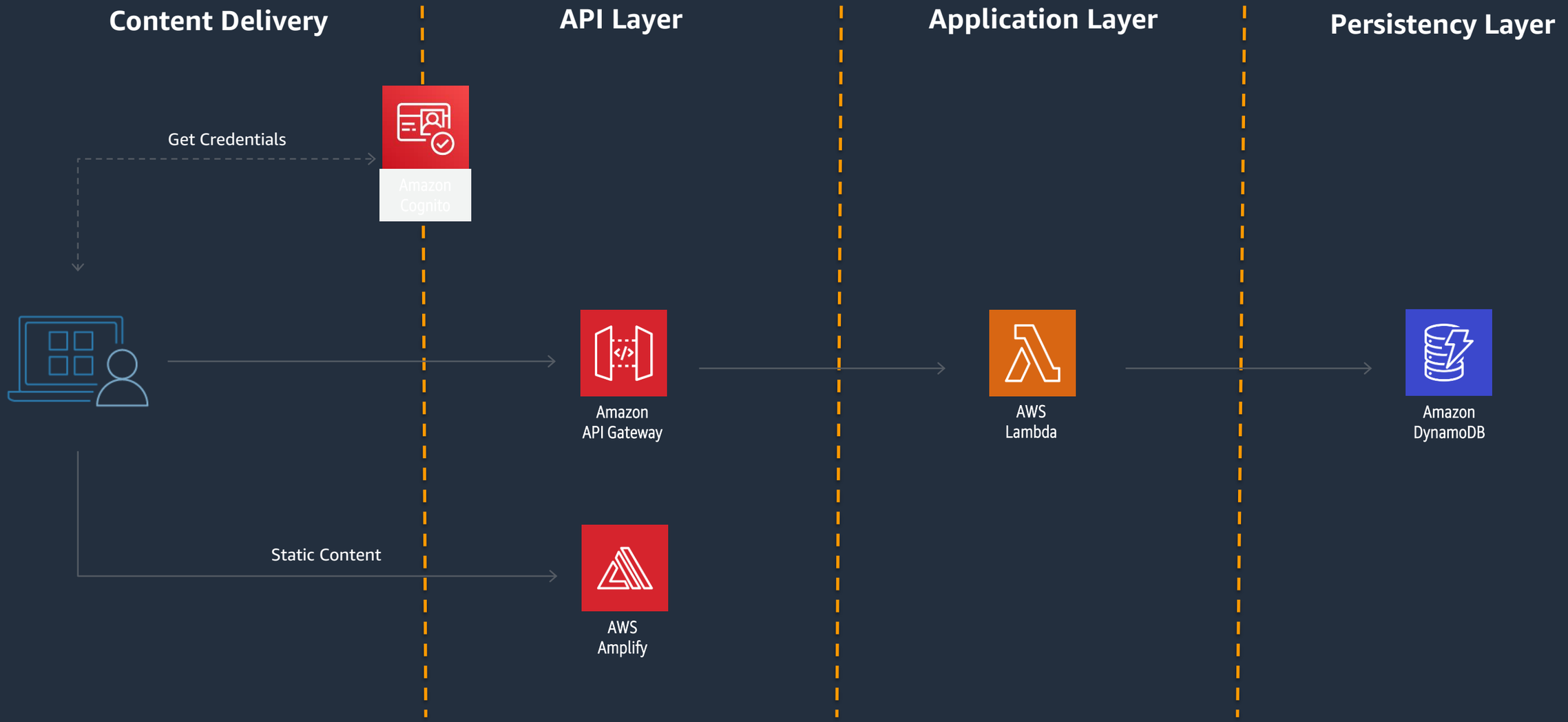
aws modern apps

Your Task: Build the Wild Rydes website

Welcome to Wild Rydes Inc.,
Employee #3!

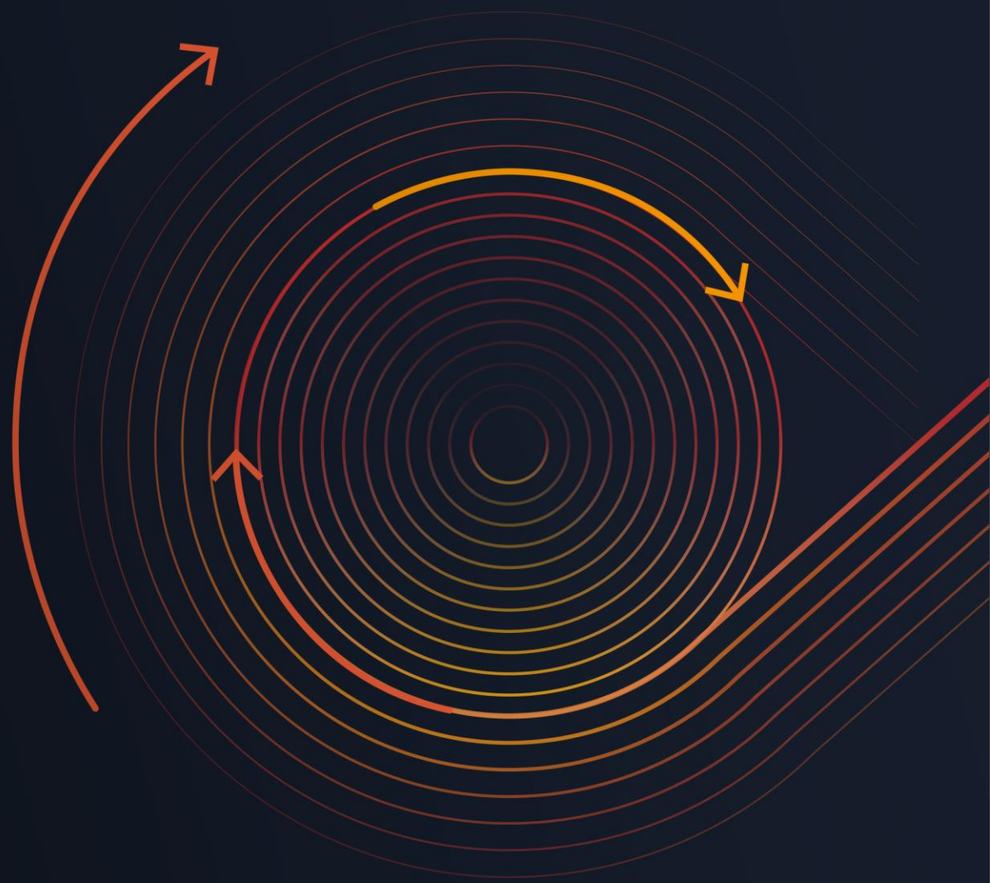
aws modern apps

What are we building?



aws modern apps

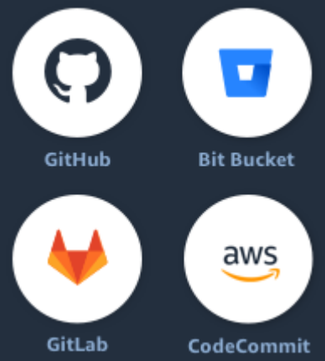
Services Used

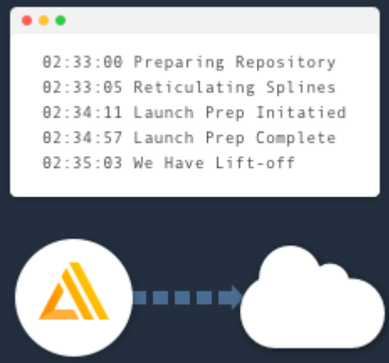


aws modern apps

AWS Amplify Console

Build, deploy, and host cloud-powered modern web apps





1. Connect your repository

2. Configure build settings

3. Deploy your app

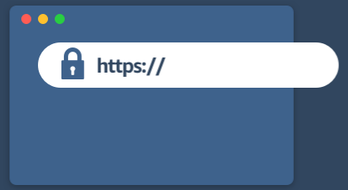
Optional deployment of backend resources + fully managed frontend hosting

aws modern apps

AWS Amplify Console

Globally available

Your app is served via Amazon's reliable content delivery network with 144 points of presence globally.



Easy custom domain setup

Set up custom domains managed in Amazon Route 53 with a single click plus get a free HTTPS certificate.

Simplified continuous workflows

Connect your repository to 'git push' changes to your frontend and backend in a single workflow.

Feature branch deployments

Work on new features without impacting production. Create branch deployments linked to each feature branch.

Atomic deployments

All deployments either rollout successfully or fail without requiring maintenance windows.

Password protection

Share yet-to-be released features with internal stakeholders by setting a username and password.

aws modern apps

AWS Lambda: Serverless computing

EVENT SOURCE

Changes in data state

Requests to endpoints

Changes in resource state

FUNCTION

Node.js
Python
Java
C#
Go
PowerShell
Ruby

SERVICES (ANYTHING)

aws modern apps

Anatomy of a Lambda function

Handler() function

Function to be executed upon invocation

Event object

Data sent during Lambda Function Invocation

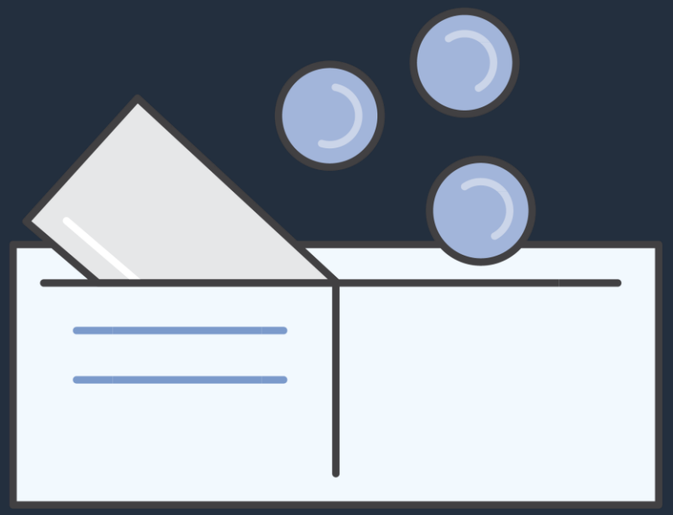
Context object

Methods available to interact with runtime information (request ID, log group, etc.)

```
s3 = boto3.resource('s3')
app = App()

def lambda_handler(event, context):
        # do something
...
```


aws modern apps

Fine-grained pricing



- Buy compute time in 100ms increments
- Low request charge
- No hourly, daily, or monthly minimums
- No per-device fees

Never pay for idle

Free Tier

1M requests and 400,000 GB-s of compute.

Every month, every customer.

aws modern apps

Using AWS Lambda

Bring your own code

- Node.js, Java, Python, C#, Go, Powershell, Ruby
- Bring your own libraries (even native ones)

Flexible use

- Synchronous or asynchronous
- Integrated with other AWS services

Simple resource model

- Select power rating from 128 MB to 3 GB
- CPU and network allocated proportionately

Flexible authorization

- Securely grant access to resources and VPCs
- Fine-grained control for invoking your functions

aws modern apps

Lambda permissions model

Fine grained security controls for both execution and invocation:

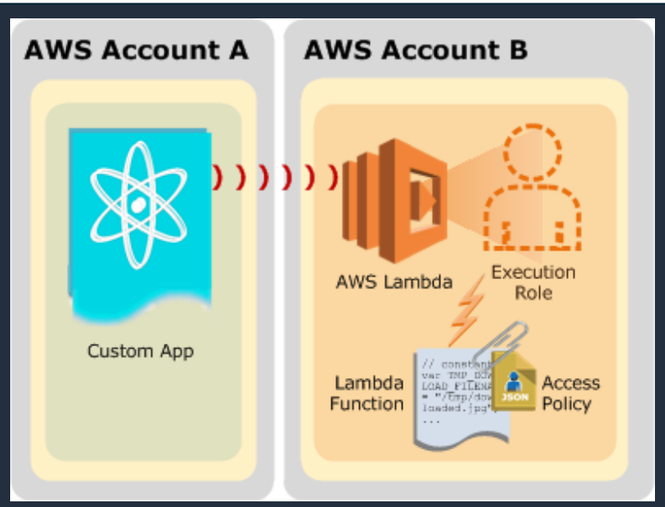
Execution policies:

- Define what AWS resources/API calls can this function access via IAM
- Used in streaming invocations
- E.g. “Lambda function A can read from DynamoDB table users”

Function policies:

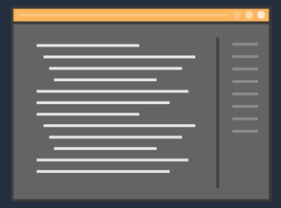
- Used for sync and async invocations
- E.g. “Actions on bucket X can invoke Lambda function Z"
- Resource policies allow for cross account access

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```



aws modern apps

Using AWS Lambda



Authoring functions

- Cloud9
- WYSIWYG editor or upload packaged .zip
- Third-party plugins (Eclipse, Visual Studio)

Programming model

- Use processes, threads, /tmp, sockets normally
- AWS SDK built in (Python and Node.js)

Monitoring and logging

- Metrics for requests, errors, and throttles
- Built-in logs to Amazon CloudWatch Logs
- X-Ray integration

Stateless

- Persist data using external storage
- No affinity or access to underlying infrastructure

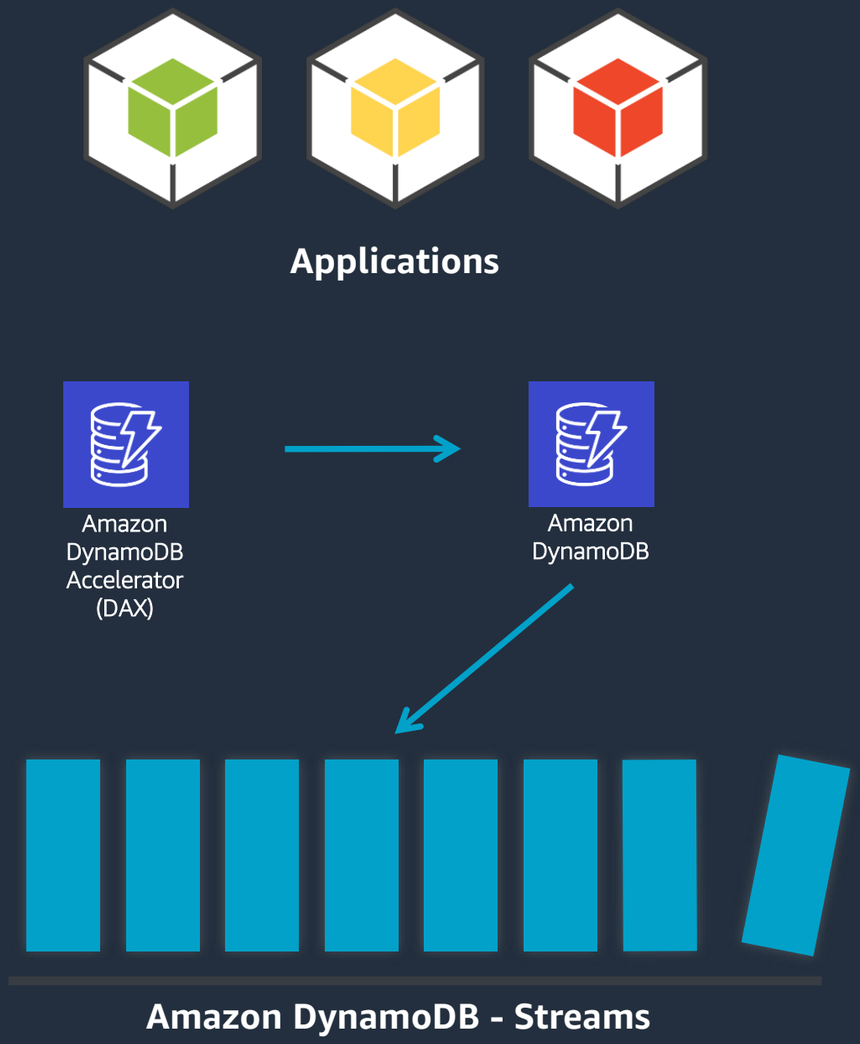
aws modern apps

Accessing stored data in Amazon DynamoDB

Simple

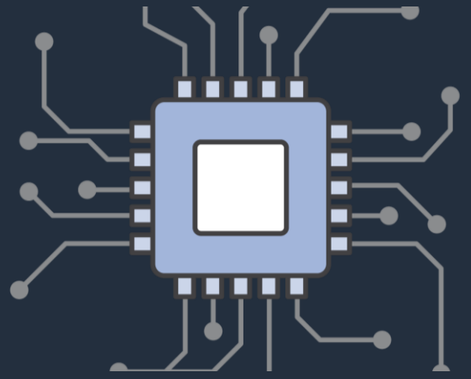
- GetItem(primaryKey)
- PutItem(item)

```
const doc = require('dynamodb-doc');
const dynamo = new doc.DynamoDB();

exports.handler = (event, context, callback) => {

    const id = event.payload.id;
    dynamo.getItem(id, callback);

};
```



aws modern apps

Amazon API Gateway



Create a unified API frontend for multiple micro-services

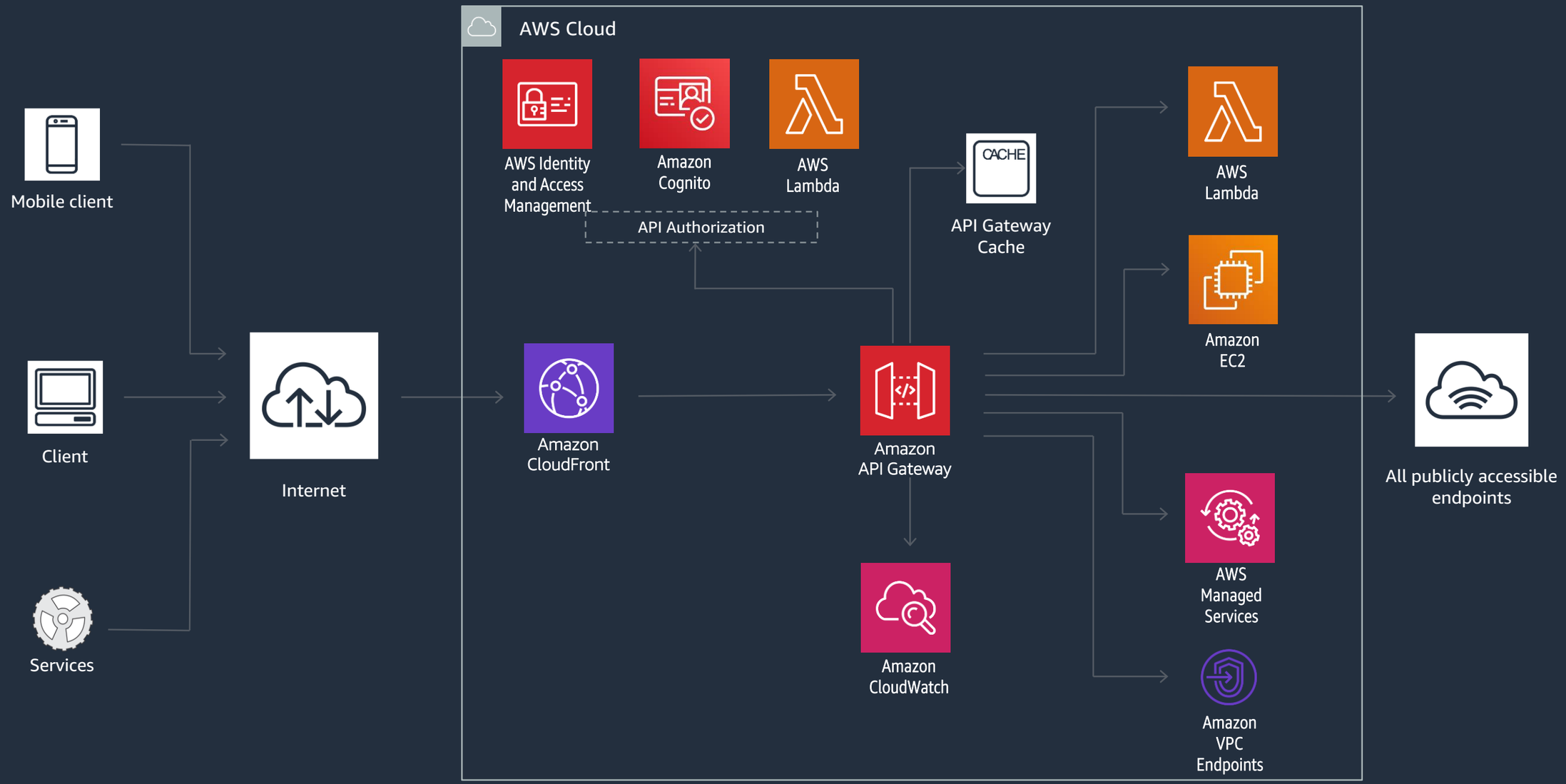
DDoS protection and throttling for your backend

Authenticate and authorize requests to a backend

Throttle, meter, and monetize API usage by 3rd party developers

aws modern apps

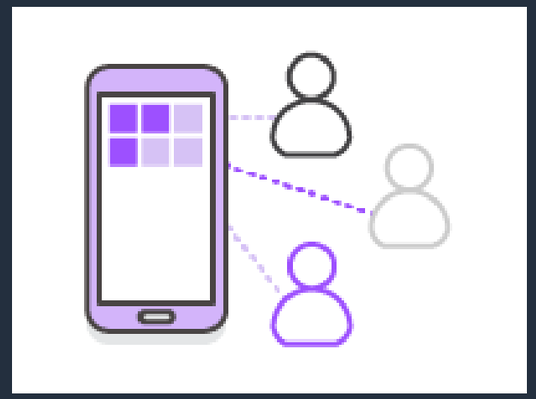
API Gateway - Serving dynamic content



aws modern apps

Amazon Cognito

Add user sign-up, sign-in, and data synchronization to your apps



Add user sign-up and sign-in to your mobile and web apps

Federate identities and provide secure access to AWS resources

Store and sync across devices

aws modern apps

aws modern apps

Let’s Build!

https://webapp.serverlessworkshops.io/