Wifi / Password

Wifi name:
TDPK-WIFI

Username:
AWSOn-boardingworkshop1
AWSOn-boardingworkshop2
AWSOn-boardingworkshop3
Password
Welcome@2022

https://github.com/TIDC-PS-Inter/AWS-Workshop

Part 1

AWS Workshop Series

Day 5: Serverless

Taking Enterprise Beyond the Cloud by TrueIDC

Mr. Niran Sohinkong

Professional Service Manager

Presented by

- Niran Sohinkong (Nueng)
- Professional Service Manager, TrueIDC
- AWS DevOps
- AWS SysOps / Architect
- niran.soh@ascendcorp.com

trueIDC

Agenda

- Introduction to Serverless
- AWS Amplify
- AWS Lambda
- AWS API Gateway
- AWS Cognito
- AWS DynamoDB

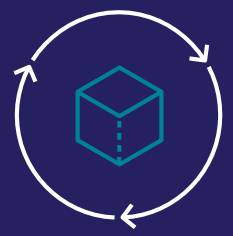
trueIDC

TODAY: 80% OF DEVELOPER TIME IS OPERATIONS AND MAINTENANCE*

In the future, the only code you write is business logic

* Source: Deloitte 2019

Serverless enables customer to focus on business value



Agility
Get to market faster, deliver features

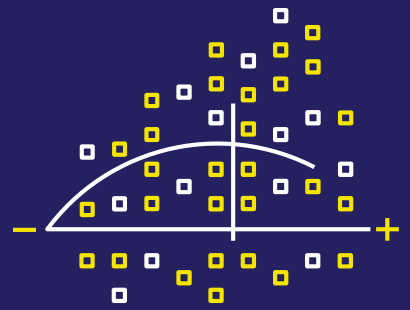
Performance
High performance and scalability

Cost
Pay for value, lower TCO

Security
Security and isolation by design

What are customers building with Serverless?

IT Automation



Data processing

Web applications

Machine Learning

Serverless is more than compute

COMPUTE

AWS Lambda

AWS Fargate

DATA STORES

Amazon S3

Amazon Aurora Serverless

Amazon DynamoDB

INTEGRATION

Amazon API Gateway

AWS AppSync

Amazon EventBridge

Amazon Simple Queue Service (Amazon SQS)

Amazon Simple Notification Service (Amazon SNS)

AWS Step Functions

AWS operational responsibility models

Compute	Virtual machine	Amazon EC2	AWS ElasticBeanstalk		Fargate	Lambda
Databases	MySQL	MySQL on Amazon EC2	Amazon RDS for MySQL	Amazon RDS	Amazon Aurora	DynamoDB
Storage	Storage					S3
Messaging	ESBs		Amazon MQ	Amazon Kinesis		EventBridge/SQS/SNS
Analytics	Hadoop	Hadoop on EC2	EMR	Amazon OpenSearch Service		Amazon Athena

AWS Amplify

Modern application requirements

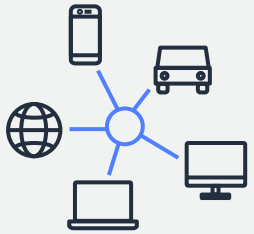
Performance	Millisecond latency
Scale	Traffic spikes and peak usage
Screens	100s of browsers, tablets, devices
Global	Worldwide access
Reliability	Zero downtime
Agility	Speed to market
Frameworks	React, Angular, Vue, NextJS
Operational overhead	Avoid managing physical servers
Economics	Pay for usage
User experience	Feature-rich applications
Security	Highest standards for privacy and data security

1. E-commerce sites
2. Media streaming
3. Social media
4. Online gaming
5. Shared economy

AWS Amplify provides tools for front-end developers to build, ship, and scale their app

Build

Configure application backend

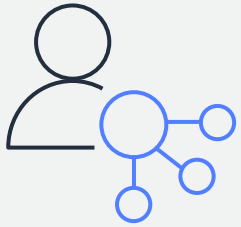


Build features and connect frontends to AWS Services

Ship

Deploy your app frontend and backend

Scale



Customize and integrate with CDK

AWS Amplify

The fastest way to build extensible, full-stack web and mobile apps

Get to market faster

Libraries

Data modeling

Built-in back-end code

Client-side data access code generation

Figma to React UI component code generation (New)

Build feature-rich apps

Sign-up and sign-in workflows

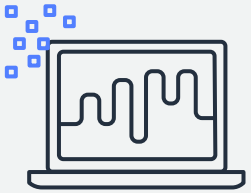
Location-aware

Real-time and offline

REST & GraphQL API calls

SQL, NoSQL and cloud storage

AI/ML, Voice, Text, Predictions



Scale

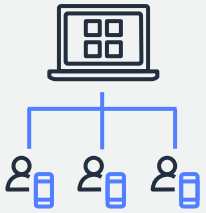
Global

Reliable

Secure

Scalable

Adapt



AWS Amplify

Tools and services for frontend developers to build web and mobile apps

Amplify Libraries

Use case-centric libraries to build cloud powered mobile and web apps. Amplify libraries are powered by AWS services and can be used with new backends created with the Amplify CLI and Amplify Studio, or your existing AWS backend.

Amplify Studio

Visual interface for setting up your app backend. New: visually build UI components and generate React code. Integrates with Figma to improve designer and developer collaboration.

Amplify CLI

Toolchain to configure and maintain your app backend from your local desktop. Configure cloud functionality using the CLI's interactive workflow and intuitive use cases such as auth, storage, API.

Amplify Hosting

Fully managed CI/CD and hosting service for fast, secure, and reliable static and server-side rendered apps that scale with your business. Supports modern web frameworks such as React, Angular, Vue, Next.js, Gatsby, Hugo, Jekyll, and more.

BUILD with AWS Amplify

Full-stack developer experience across use case feature categories

BUILD

Configure AWS backends fast

Seamlessly connect frontends

CLI

Studio

Libraries

Feature categories

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

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

Connect App Using Amplify Libraries

Frontend libraries for Web, iOS, Android, React Native, and Flutter

JavaScript	React	React Native	Angular
Vue	Next.js	Android	iOS
Ionic	Flutter		

AWS Amplify Hosting

Modern hosting requirements

Client-Side Rendered (single page application)

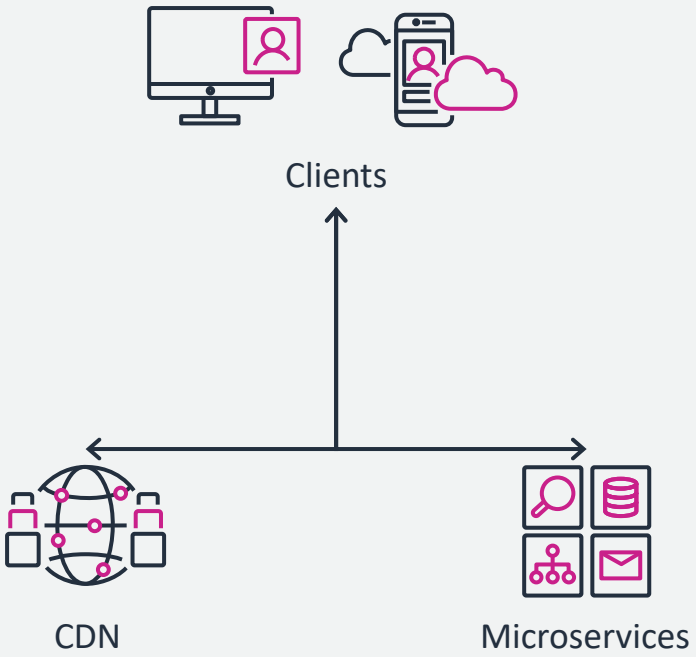
- Frontend application loaded as JavaScript and runs in the client browser.
- JavaScript files contain application logic, UI, and communication with backend.
- Popular frameworks include React, Angular, Vue

Server-Side Rendered (SSR)

- Rendering occurs on the server before sending page to browser.
- Data is fetched from a database or CMS.
- Ideal for applications that have content that is personalized for each user.
- Popular frameworks include NextJS, NuxtJS, GatsbyJS

Static Site Generators (SSG)

- Content is generated at the build time
- Ideal for sites content does not need to be highly personalized
- Typically used in concert with a headless CMS and CDN
- Popular solutions include Gatsby, Eleventy, Hugo, VuePress, and Jekyll



Amplify Hosting

Deploy and host globally using Amazon CloudFront

How it works

1

Connect your repository

GitHub

Bitbucket

GitLab

aws CodeCommit

2

Configure build settings

3

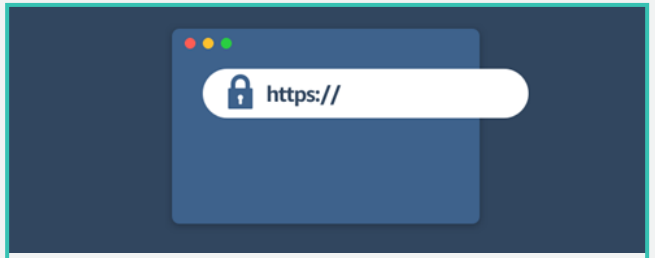
Deploy your app

```
02:33:00 Preparing Repository
02:33:05 Reticulating Splines
02:34:11 Launch Prep Initiated
02:34:57 Launch Prep Complete
02:35:03 We Have Lift-Off
```

Amplify Hosting

Features for hosting modern web applications

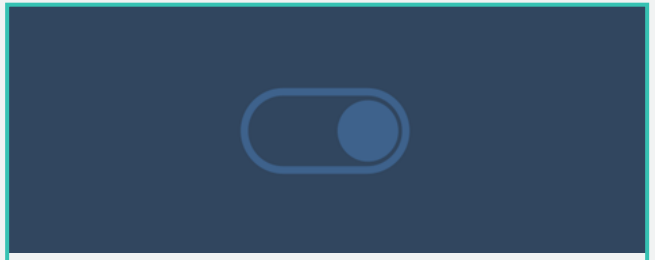
Globally available



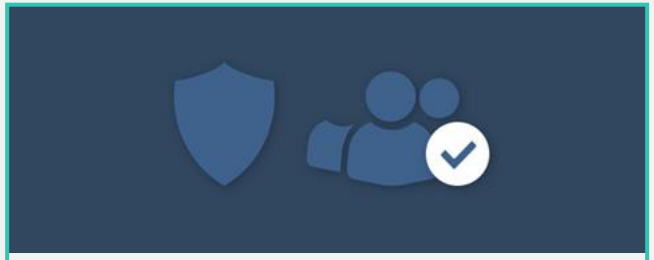
Easy custom domain setup

Simplified continuous workflows

Feature branch deployments



Atomic deployments



Password protection

Amplify Hosting benefits

Performance

Deliver lowest possible latency to users globally

Scale

Handle peak usage and traffic spikes

Flexibility

Build with a comprehensive offering of compute, integration services, databases and more

Lower TCO

Pay-per-usage

Agility

Fully-managed: innovate and get to market faster

Modern hosting

Supports the common frameworks used today

AWS Lambda

Event-driven function-as-a-service

© 2022, Amazon Web Services, Inc. or its affiliates.

Serverless Architecture

Event Source

Changes in data state

Requests to endpoints

Changes in resource state

Function

Node.js
Python
Java
C#
Go
Ruby
Bring Your Own

Services / Other

Anatomy of a Lambda Function

Handler function

- Function executed on invocation
- Processes incoming event

Event

- Invocation data sent to function
- Shape differs by event source

Context

- Additional information from Lambda service
- Examples: request ID, time remaining

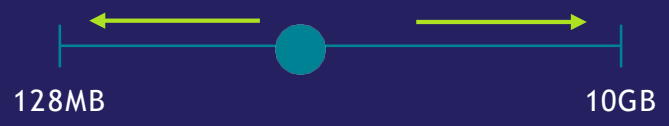
app.py

```
def handler(event, context):
  msg = 'Hello {}'.format(
                    event['name']
                   )
  return { 'message': msg }
```

Lambda Function Configuration

Power Rating

- Select between 128MB and 10GB
- CPU and network allocated proportionally
- Power tune to balance cost and speed



Permissions Model

- Execution Role grants function access to resources via IAM
- Function Policy controls invocation

Lambda Function Configuration

Timeout

- Up to 15 minutes

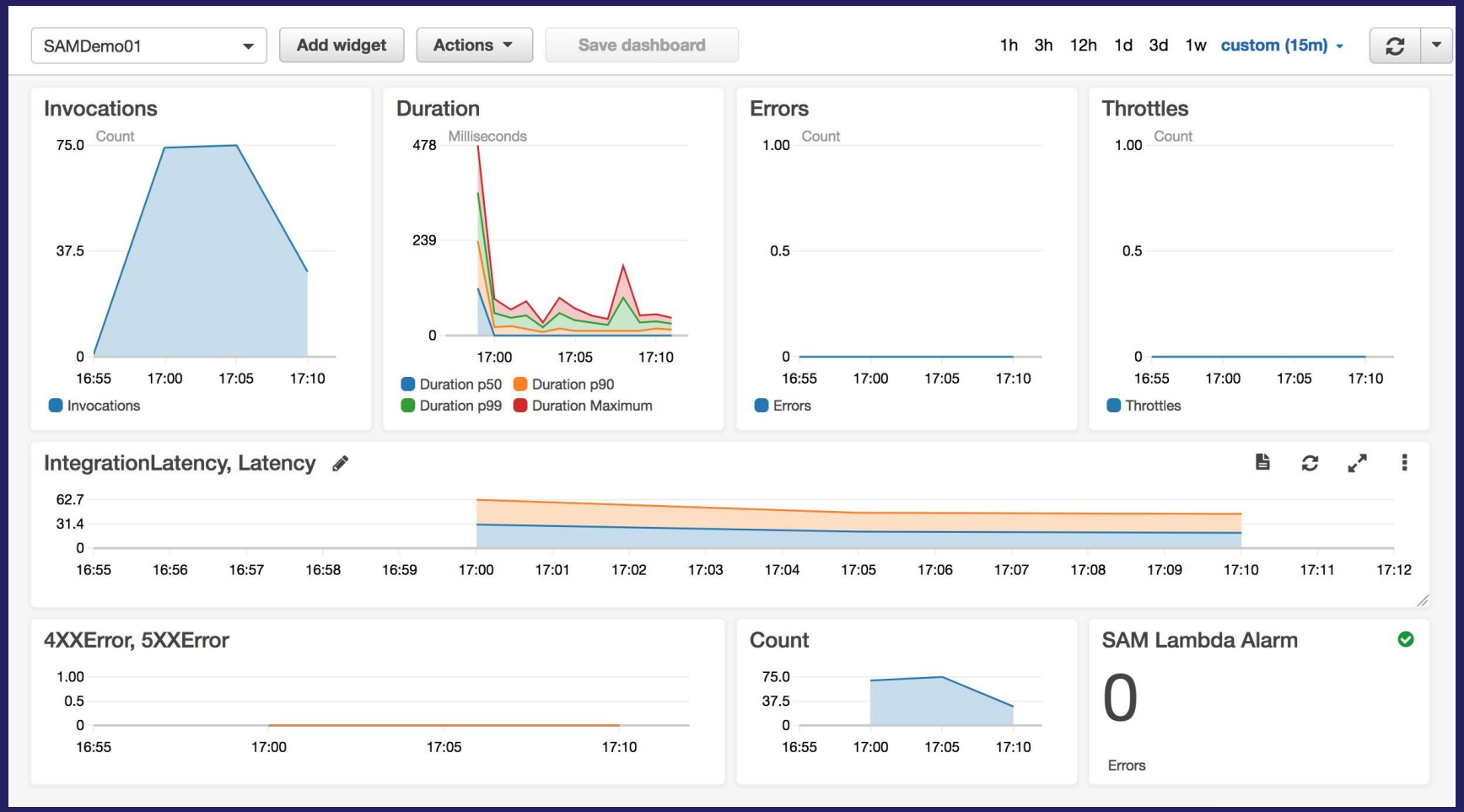
- Synchronous vs Asynchronous
- API Gateway timeout = 30 sec

Network Access

- Configure access to VPC
- Security Group rules apply

- VPC does not enhance security of function

Built in monitoring



MOST IMPORTANTLY:

Go Build Something!

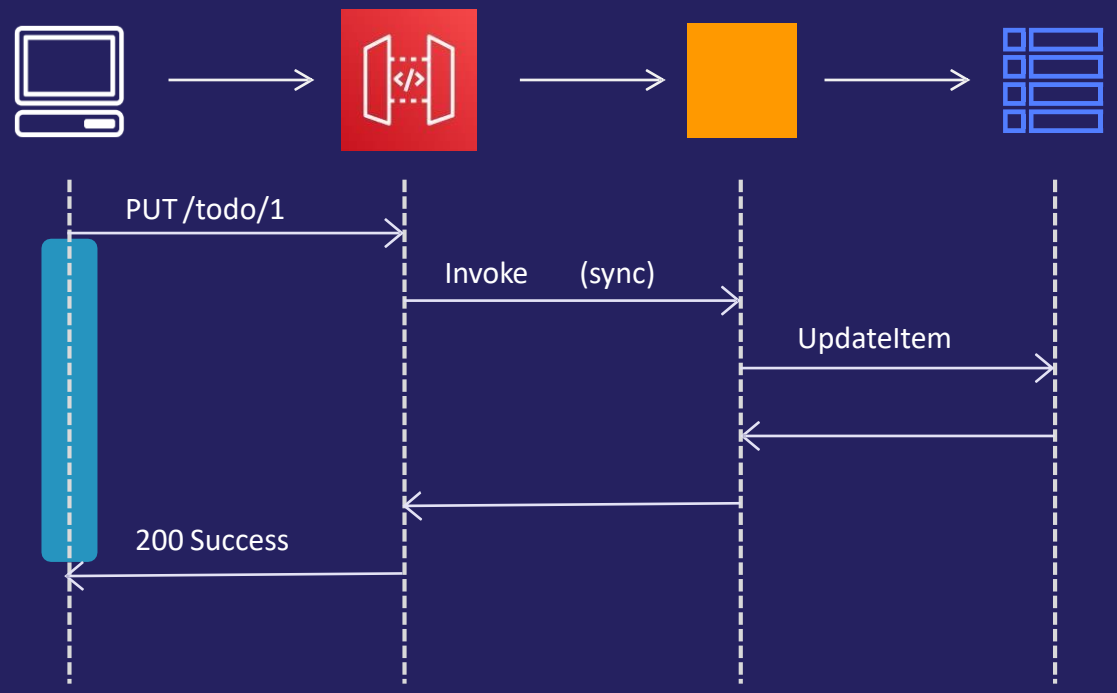
© 2022, Amazon Web Services, Inc. or its affiliates.

Serverless Architecture Patterns

© 2022, Amazon Web Services, Inc. or its affiliates.

API-Driven Architectures

- API defines the interface
  - e.g. REST, GraphQL
- Caller expects an immediate response
  - Response contains the result of the work
  - Generally, under 30 seconds
  - Synchronous processing
- Client must implement error handling, retry logic



Events enable interaction between services

Managed services provide routing, storage, and distribution of events

Amazon SQS

Messaging

Durable and scalable
Fully managed
Comprehensive security

Amazon SNS

Eventing

Performance at scale
Fully managed
Enterprise-ready

EventBridge

Choreography

Event filtering
Managed & scalable
SaaS integration

Step Functions

Orchestration

Sequencing
Parallel execution
State management

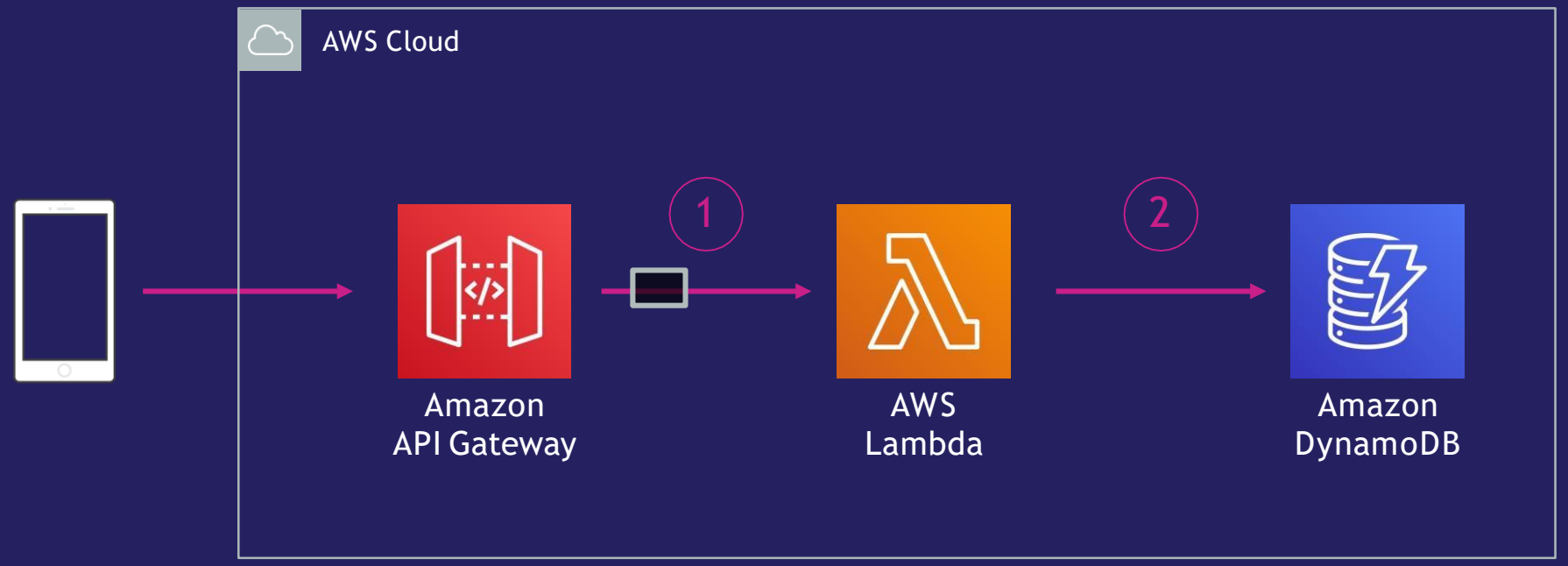
API-Driven Use Cases

Also event driven, synchronously processed

© 2022, Amazon Web Services, Inc. or its affiliates.

RESTful Microservices

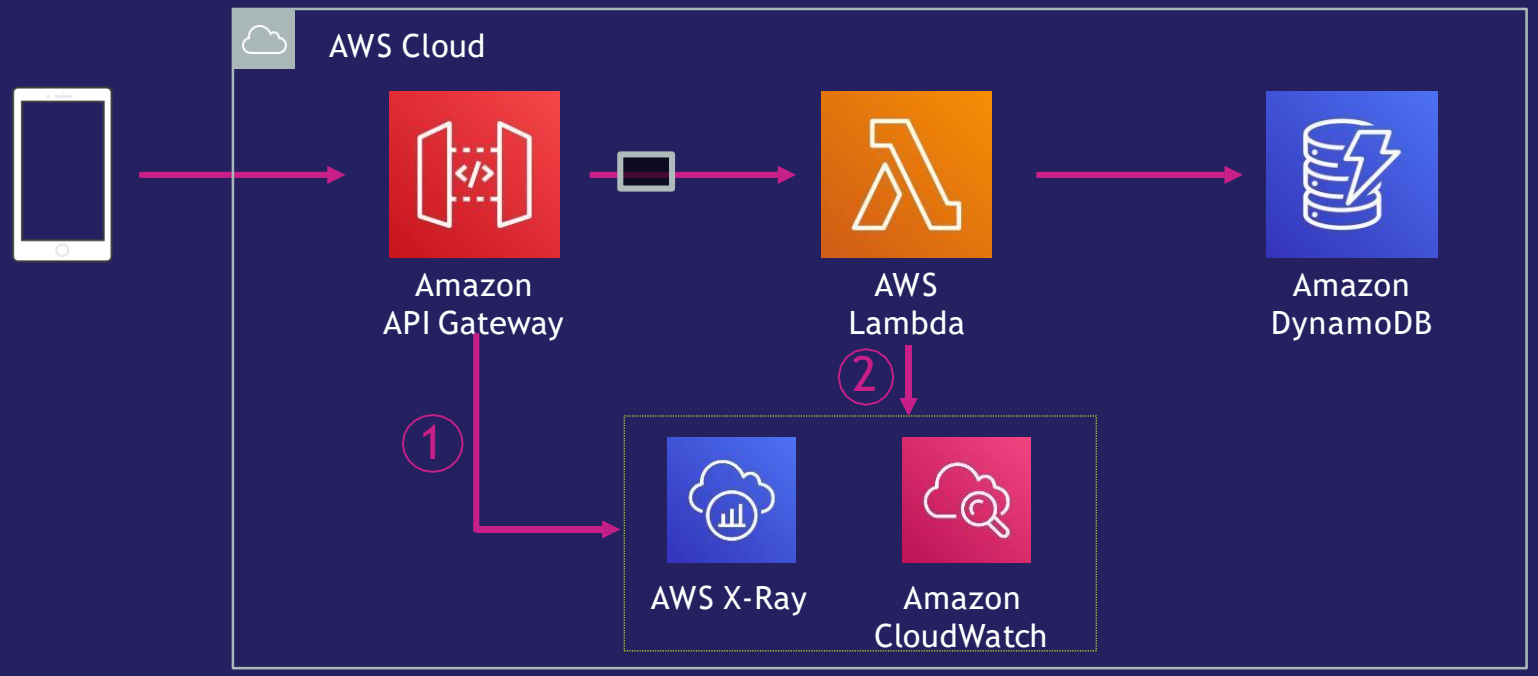
Highly-scalable microservices



1. API Gateway "translates" incoming HTTP request to event payload
2. Lambda reads / writes data from data store

RESTful Microservices with enhanced observability

Enable access logs, structured logging, and instrument code



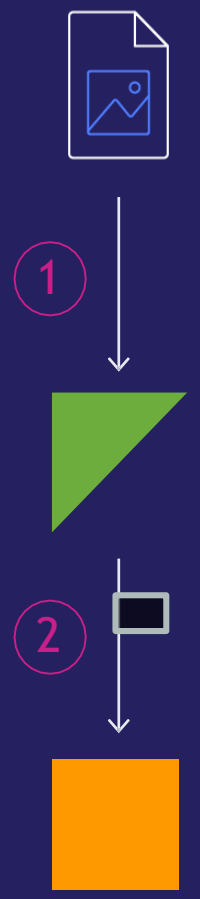
1. Enable access logs and tracing
2. Instrument code and create metrics asynchronously with CloudWatch Embedded Metric Format

Event-Driven Use Cases

© 2022, Amazon Web Services, Inc. or its affiliates.

Processing file uploads

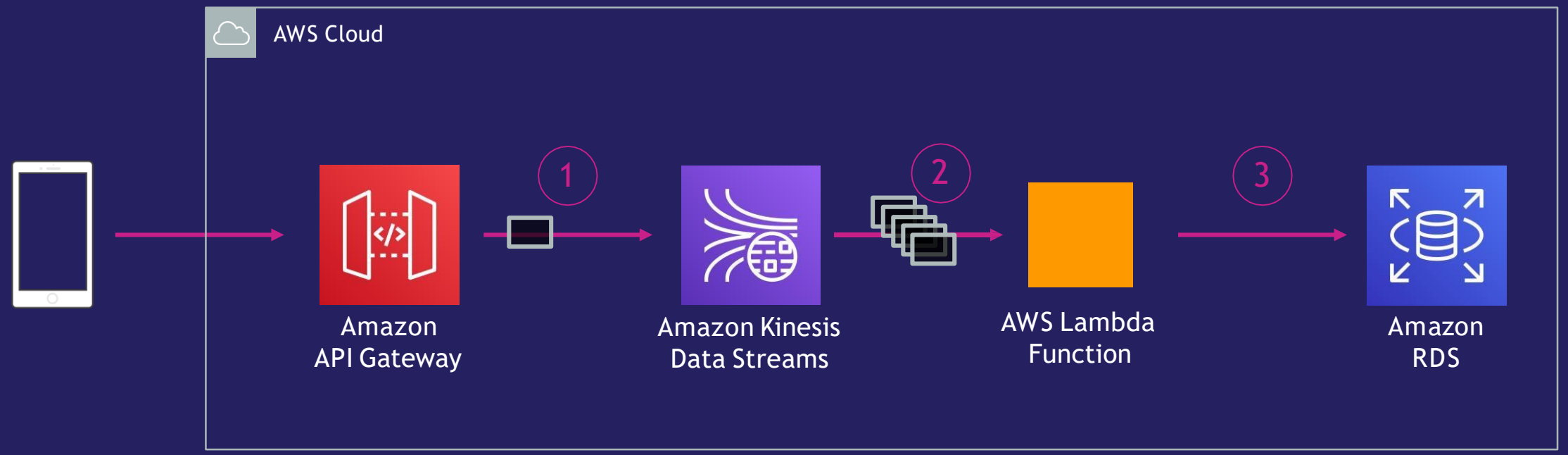
Resize photo, extract text, translate, etc.



1. Object uploaded to Amazon S3 Bucket
2. Asynchronous invoke of Lambda function, event payload includes:
   - Bucket name
   - Object key

“Semi-Serverless” Webhook

Scalable, resilient. Buffer downstream concurrency.



1. “Storage-first” pattern: API Gateway writes directly to Kinesis Data Stream
2. Kinesis as a buffer to limit concurrency downstream
3. Perform transaction(s) on batch

Amazon Cognito

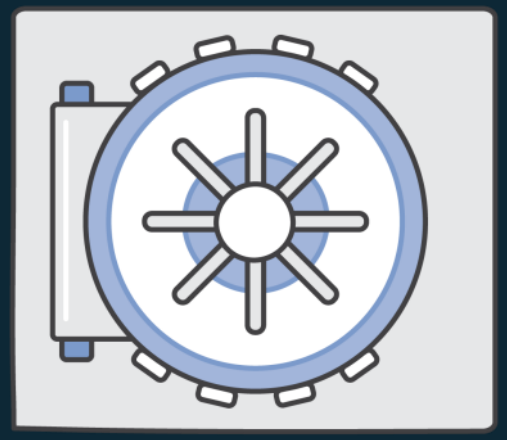
Identity for your web and mobile apps

What do customers want to do?

Focus on differentiating features in their applications

Provide flexible and modern options for authenticating

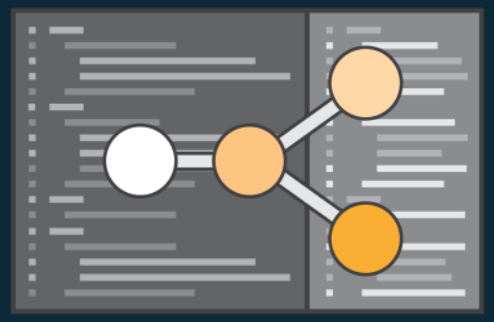
Build standards-based, API-driven interoperable platforms



Protect the security and privacy of their customers

What challenges are they facing?

Implementing identity from scratch is challenging to build and operate



Options result in more integrations to understand and code to build

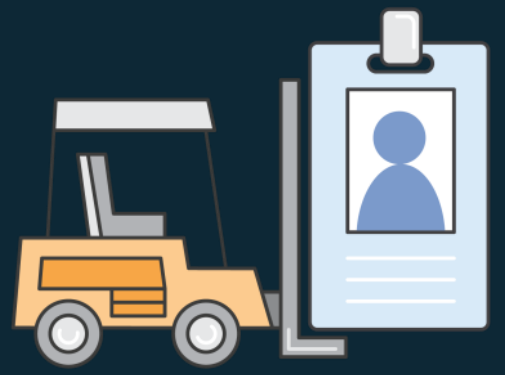
Standards such as Open ID Connect, OAuth2, and SAML are hard to master

Applications operate in an increasingly hostile landscape

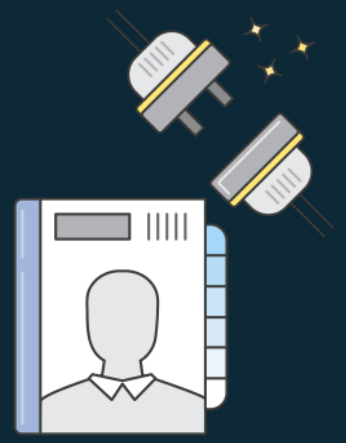
aws

Introducing Amazon Cognito

Simple and secure user sign-up, sign-in, and access control for web and mobile apps.

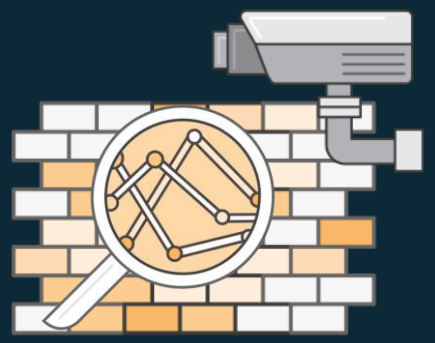


Offload undifferentiated identity heavy lifting



Use your choice of existing or cloud native identities.

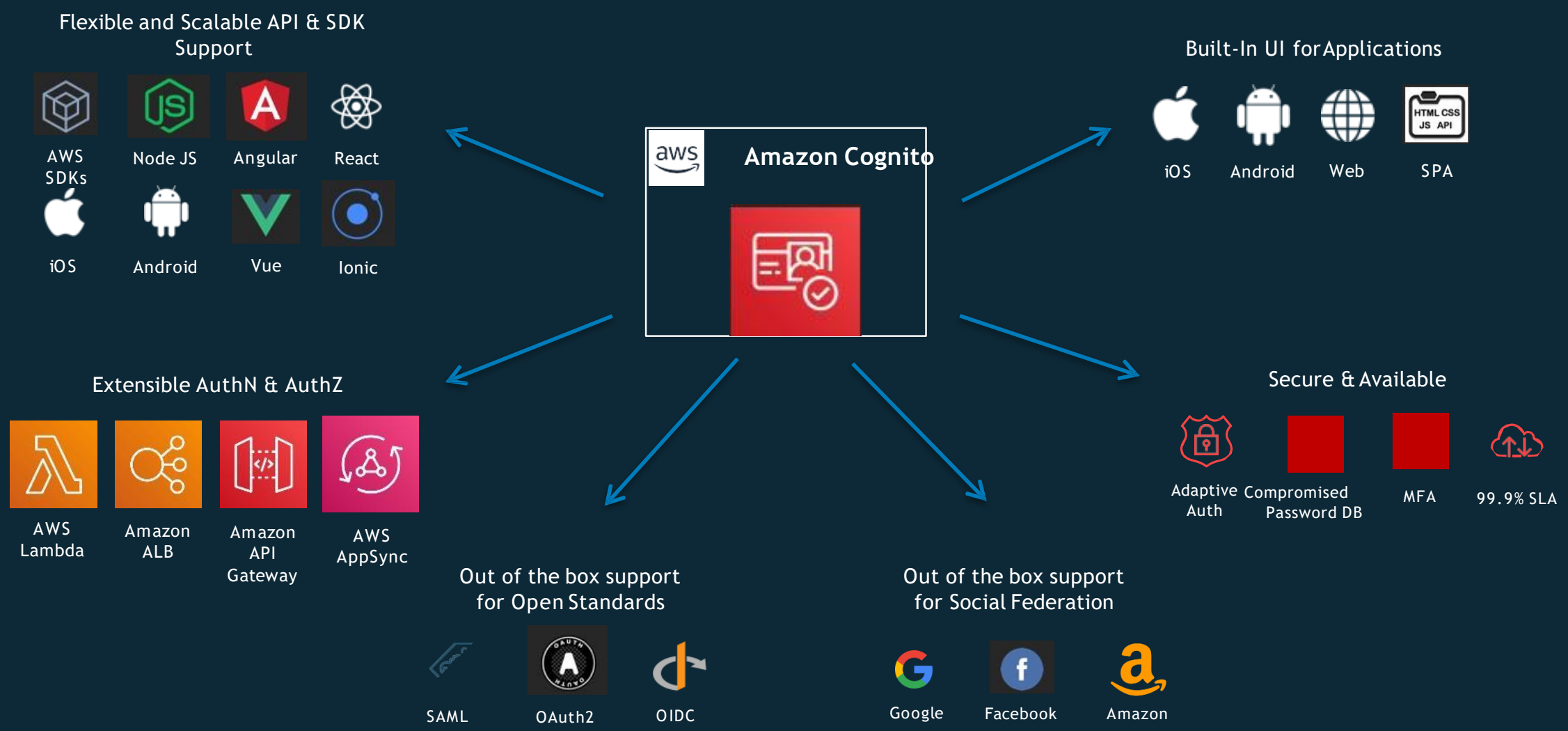
Use standards-based authentication



Provide advanced security for your apps and users

aws

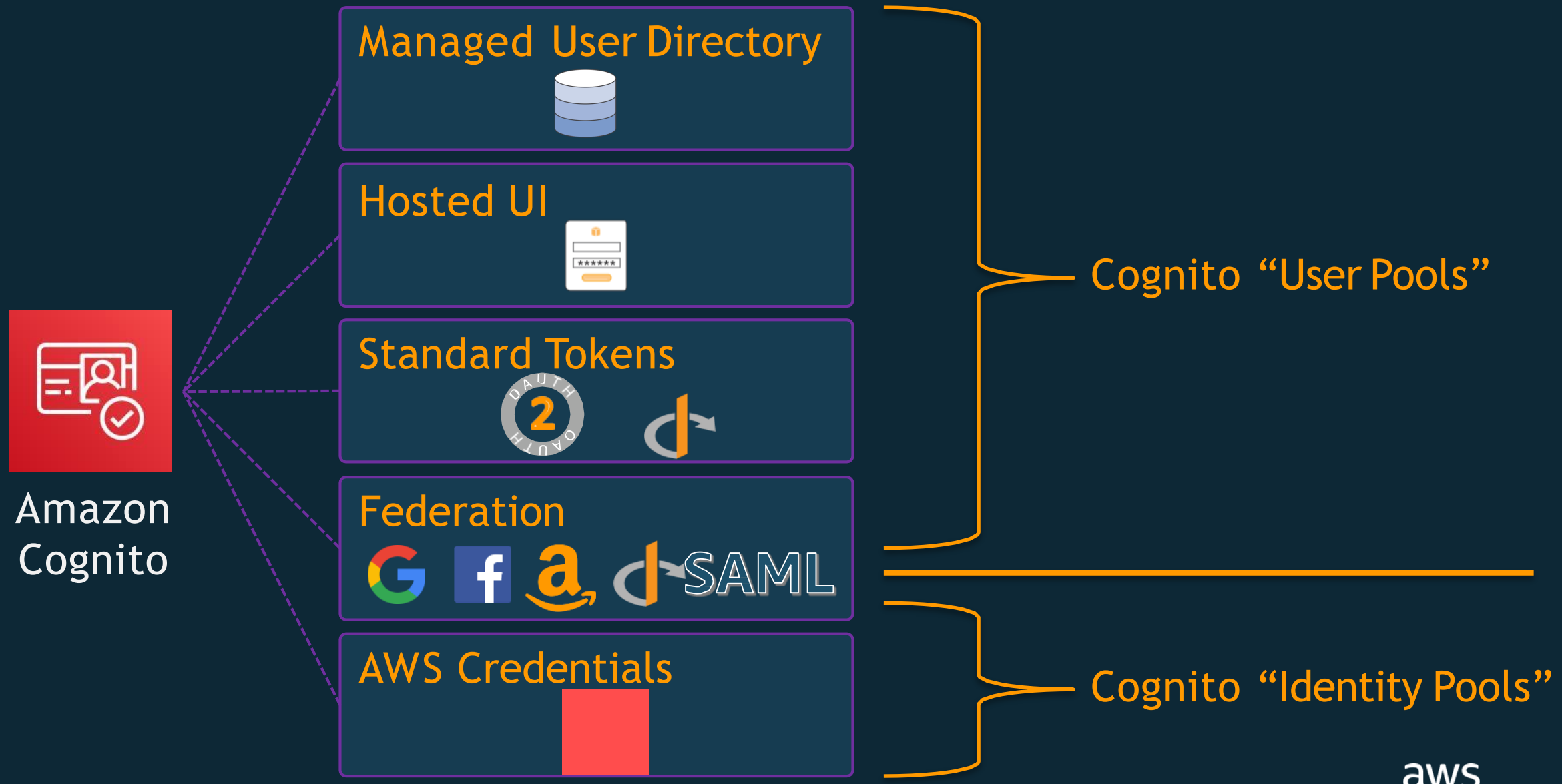
Cognito: Flexible and Fully Managed Application Identity



Amazon Cognito

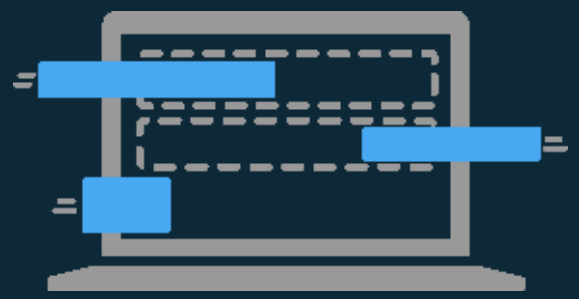
Key features

Amazon Cognito Overview



Key Features in Amazon Cognito User Pools

Secure, low-cost, and fully managed user directory that scales to millions of users



Built-in, Customizable User Interface for Sign up / Sign in

OAuth 2.0 Support

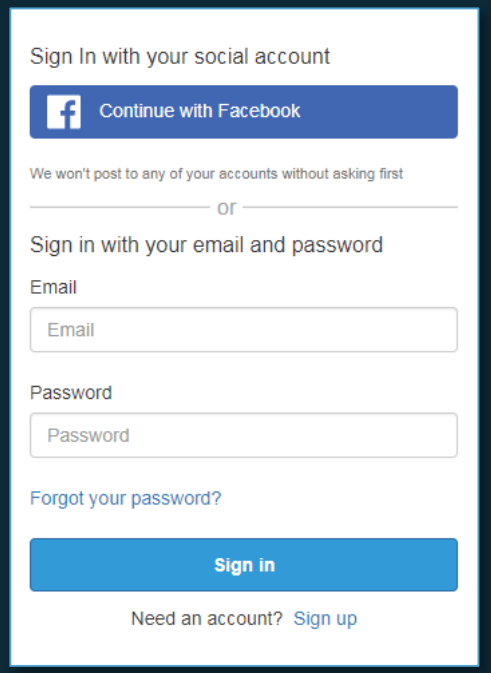
Federation with Facebook, Login with Amazon, Google, and custom OIDC / SAML providers

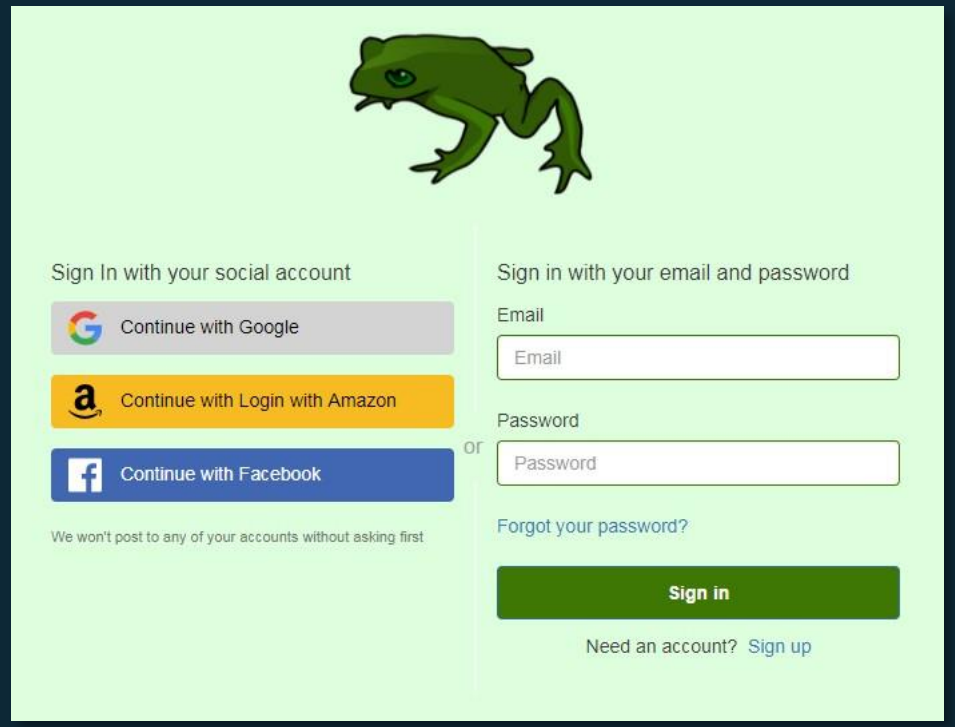
Managed user directory

- Serverless directory
  - Nothing to manage
  - API driven
  - Multi-AZ redundancy
- User & group storage
  - Profile information (name, email, etc)
  - Credential & device information (SRP verifier, MFA, etc)
  - Extensible with custom attributes

Hosted user interface

- Facilitates user flows (sign up/in, forgot password, etc)
- Customizable logo, style and branding
- Use your own domain





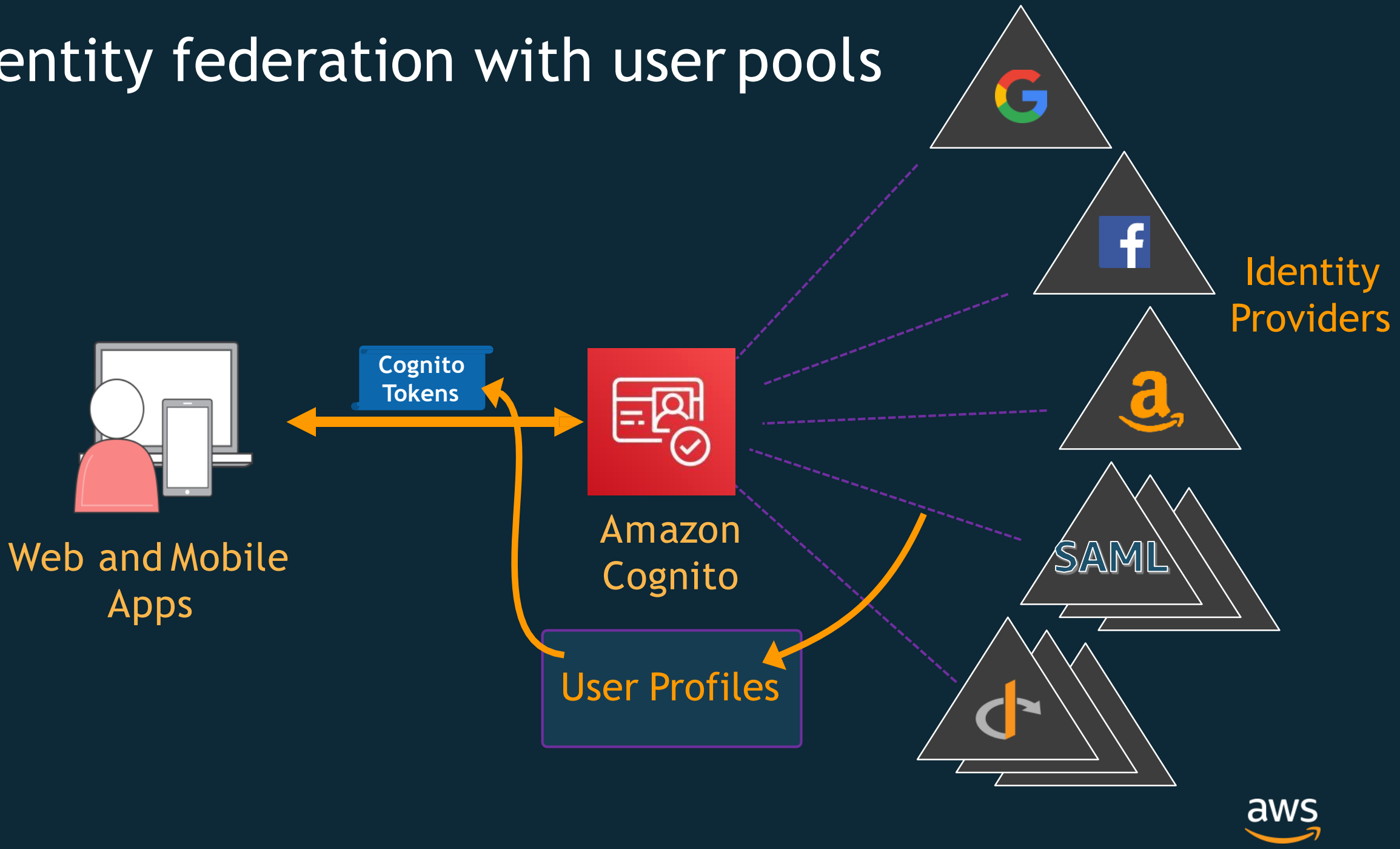
Support for OAuth 2.0 in Cognito User Pools

- OAuth 2.0 flows:
  - Authorization code
  - Implicit
  - Client credentials
  - Resource owner password credentials

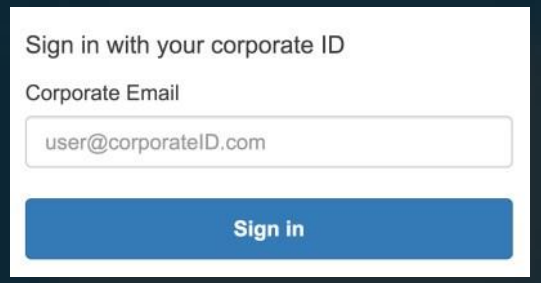
- Custom scopes defined for resource servers

aws

Identity federation with user pools



Cognito manages federation process for you



SAML Service Provider
(Your App/Cognito)

1. Provide e-mail or user data

2. Discover relevant IdP

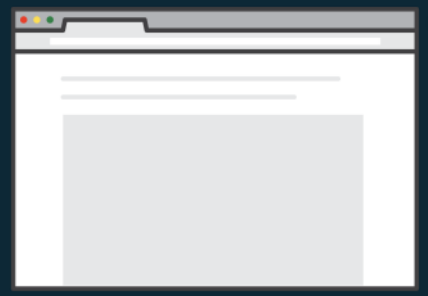
3. Generate SAML Auth request

4. Redirect to SAML IdP and authenticate

5. POST SAML response

6. User in app (authenticated)

Your Browser



SAML Endpoint
e.g. ADFS
or Shibboleth

Corporate Directory
e.g. Active Directory
or OpenLDAP

Supports just-in-time user provisioning

aws

Comprehensive user flows

User Sign-Up and Sign-In	Allow users to sign up and sign in using an email, phone number, or username (and password) for your application.
User Profile Data	Enable users to view and update their profile data – including custom attributes
Forgot Password	Provide users the ability to change their password when they forget it with a one-time password challenge
Token Based Authentication	Use JSON Web Tokens (JWTs) based on OpenID Connect (OIDC) and OAuth 2.0 standards for user authentication in your backend
Email or Phone Number Verification	Require users to verify their email address or phone number prior to activating their account with a one-time password challenge
SMS Multifactor Authentication	Require users to complete a second factor of authentication by inputting a security code received via SMS as part of the sign-in flow

aws

Amazon Cognito

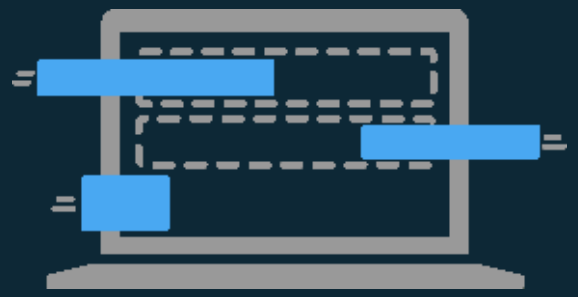
Integrating with your application

Option 1: AWS Amplify

- Amplify: Comprehensive library for building sophisticated cloud-powered apps.
- A powerful toolchain built for developers.
- A minimalist styled UI component library.
- Your choice:
  - Cognito hosted UI via Amplify
  - Native Amplify UI
- Best for: Native mobile apps and Javascript-based web apps.

https://aws-amplify.github.io/

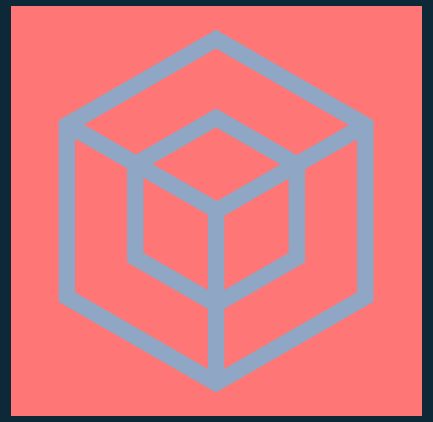
Option 2: Hosted UI using OAuth flows



- Choose the O uth flow that's appropriate for you application type (e.g. authorization code grant).
- Construct necessary requests, headers & query parameters to execute flow.
- Language agnostic, light-weight integration.
- Best for: Existing apps or new web apps where mplify isn't a fit (e.g. server side rendered)

Option 3: Direct API integration

- Integrate Cognito APIs directly into your application using the AWS SDK for your chosen language.
- Provides complete control over user experience and flows.
- Best for: Server side integrations, other advanced use cases.



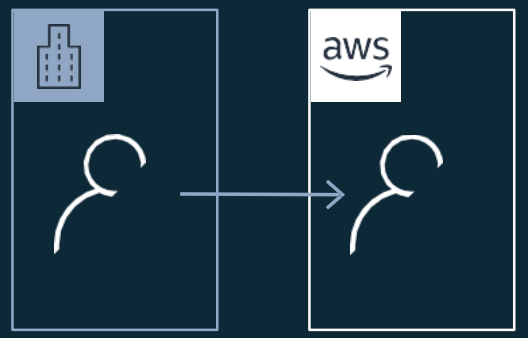
Amazon Cognito

Additional features

Additional notable features

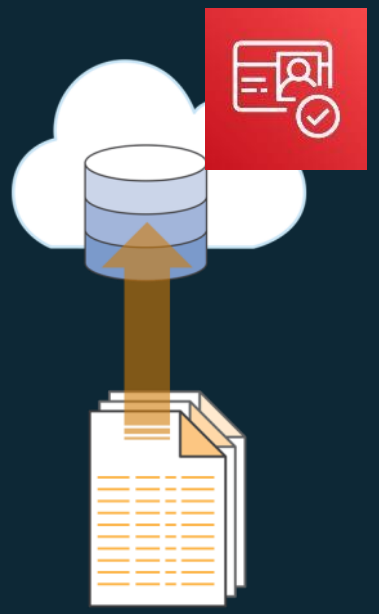
MFA options:
SMS & TOTP

Advanced Security Features:
Compromised Credentials,
Adaptive Authentication,
Security Reporting



Migration options:
Batch & just-in-time

Importing Existing Users



- Batch Imports
  - Import users by uploading .csv files
  - Users will create a new password when they first sign-in
- One-at-a-Time Migration
  - Lambda trigger integrates migration into the sign in workflow and retains existing passwords

Amazon Cognito Compliance

Australian Government
PROTECTED

General Data Protection
Regulation (GDPR)

aws

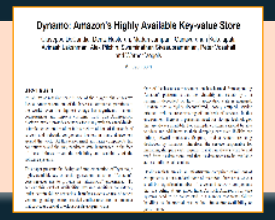
DynamoDB Immersion Day

History & How it works

The Amazon NoSQL journey

Dec 2004:

Database scalability challenges



Oct 2007:

Dynamo paper published

Jan 2012:

DynamoDB general availability

Q3 2016:

DynamoDB leader in Gartner MQ, Forrester Wave

Today:

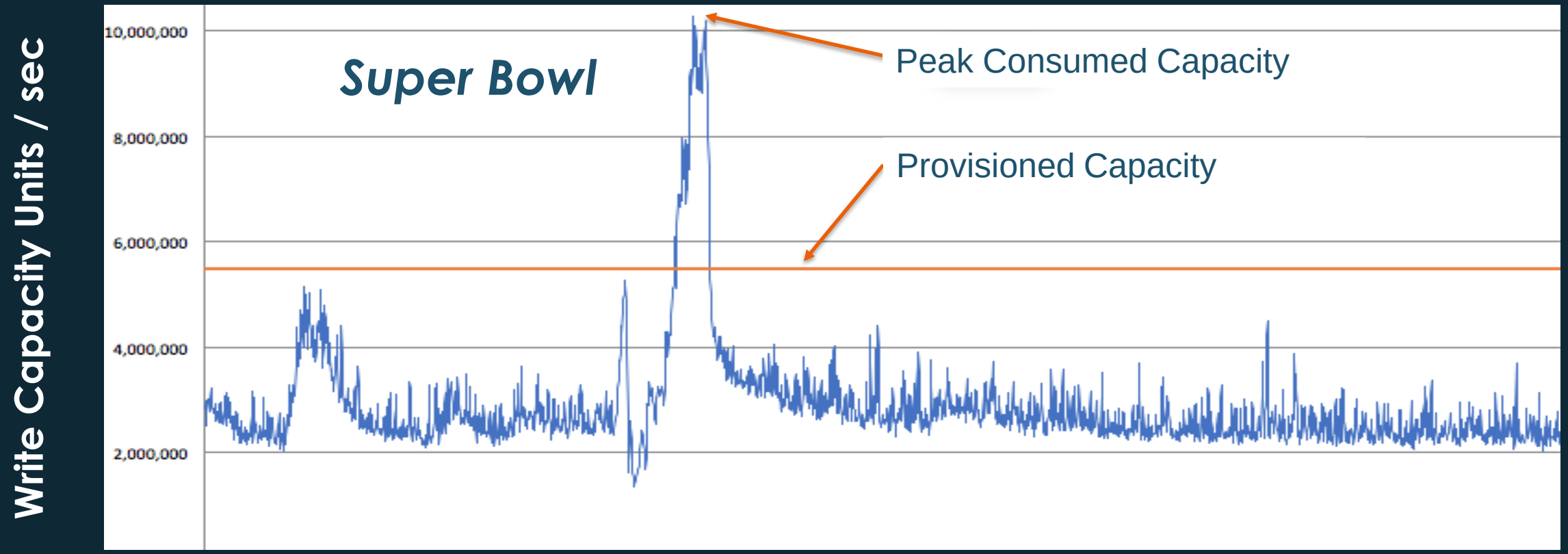
Tier 0 service powering most of Amazon

Characteristics of internet-scale apps

E-commerce | Media streaming | Social media | Online gaming | Shared economy

Users	1 million+
Data volume	TB, PB, EB
Locality	Global
Performance	Microsecond latency
Request rate	Millions per second
Access	Mobile, IoT, devices
Scale	Up and down
Economics	Pay as you go
Developer access	Instant API access

Global-Scale Events: Elastic is the New Normal



Global tables provide apps with multi-Region replication

Performance at scale

Build high-performance, globally distributed applications

Low-latency reads and writes to locally available tables

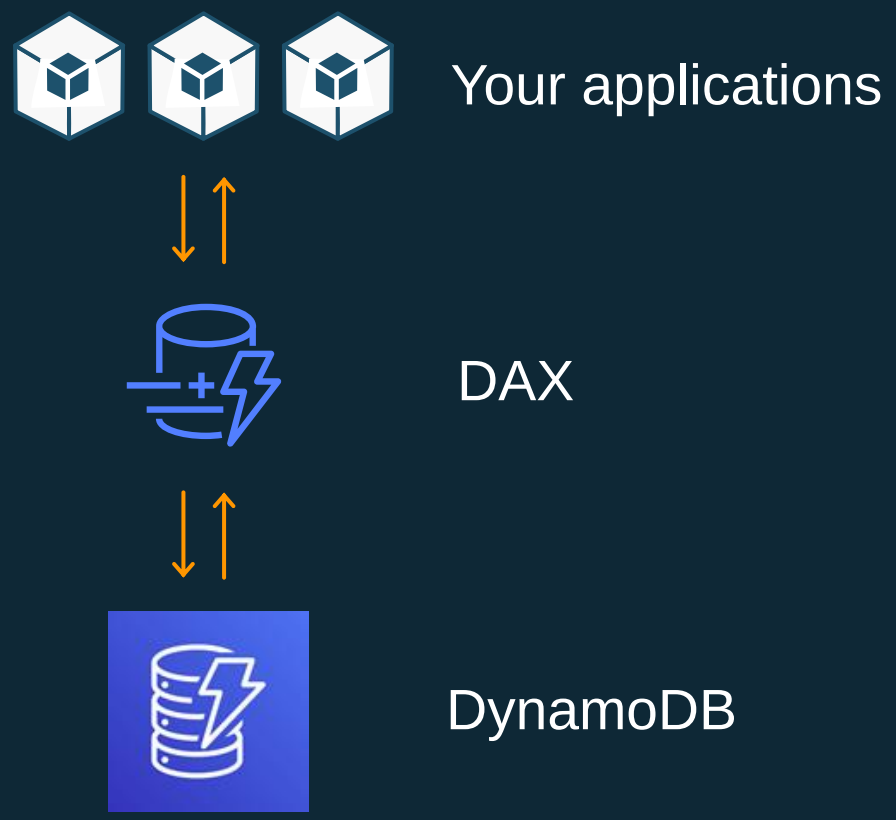
Multi-Region redundancy and resiliency and 99.999% availability

Multi-active writes from any Region

Easy to set up and no application rewrites required

DynamoDB Accelerator (DAX) adds read cache

Performance at scale



Fully managed,
highly available cache
for DynamoDB

Even faster—microsecond latency

Scales to millions of read requests
per second

API compatible

DynamoDB

- Fully managed, cloud-native NoSQL database service
- Designed for mission-critical OLTP use cases
    - Where you know access patterns
- Operational database that provides:
    - Extreme scale with horizontal scaling
    - Consistent performance at any scale
    - High availability and reliability with zero downtime
    - Global availability and cross-region replication
    - Full serverless experience
    - Integration with AWS Lambda and other AWS services

Key Concepts

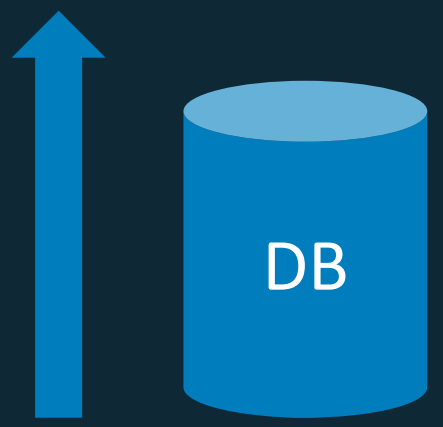
SQL and NoSQL side by side

SQL	NoSQL
Optimized for storage	**Optimized for compute**
Normalized/relational	Denormalized/hierarchical
Ad hoc queries	Instantiated views
Scale vertically	Scale horizontally
Good for OLAP	Built for OLTP* at scale

(*) DynamoDB is. Some NoSQL databases are built for analytical workloads.

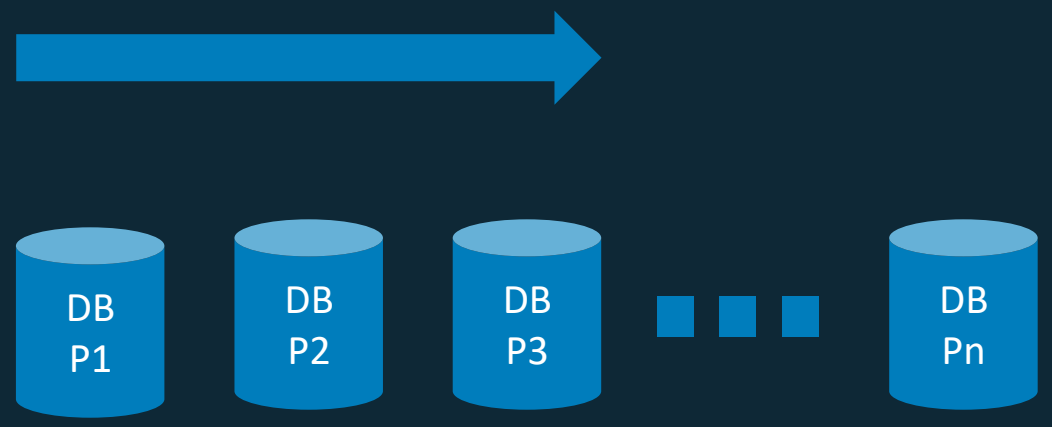
Scaling databases

Traditional SQL



Scale up

NoSQL



Scale out to many shards

Basic premise: There is a way to design data that's horizontally scalable.

Managing Throughput

Scaling

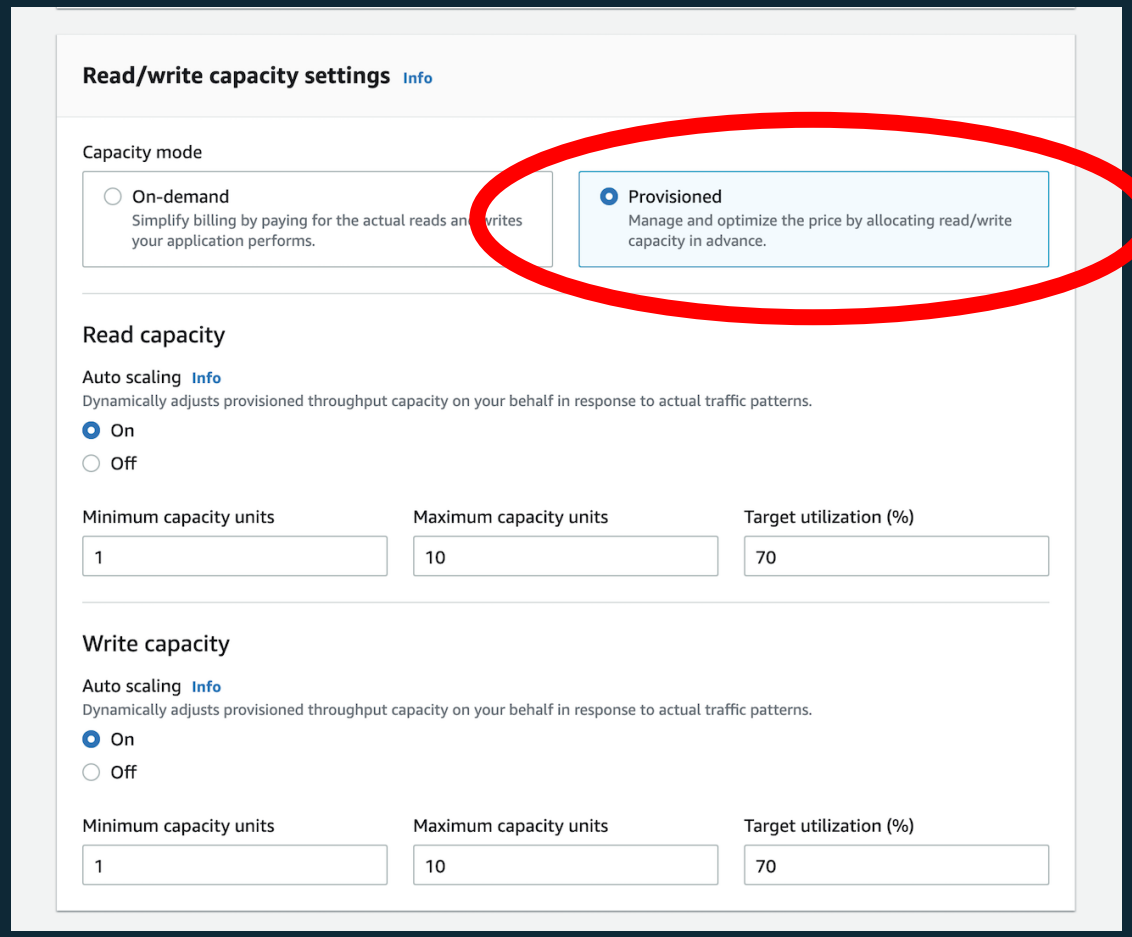
- Throughput
  - Provision any amount of throughput to a table
  - Write Capacity Unit (WCU) – 1 KB per second
  - Read Capacity Unit (RCU) – 4 KB per second
  - Independent of each other

- Size
  - Add any number of items to a table
    - Max item size is 400 KB

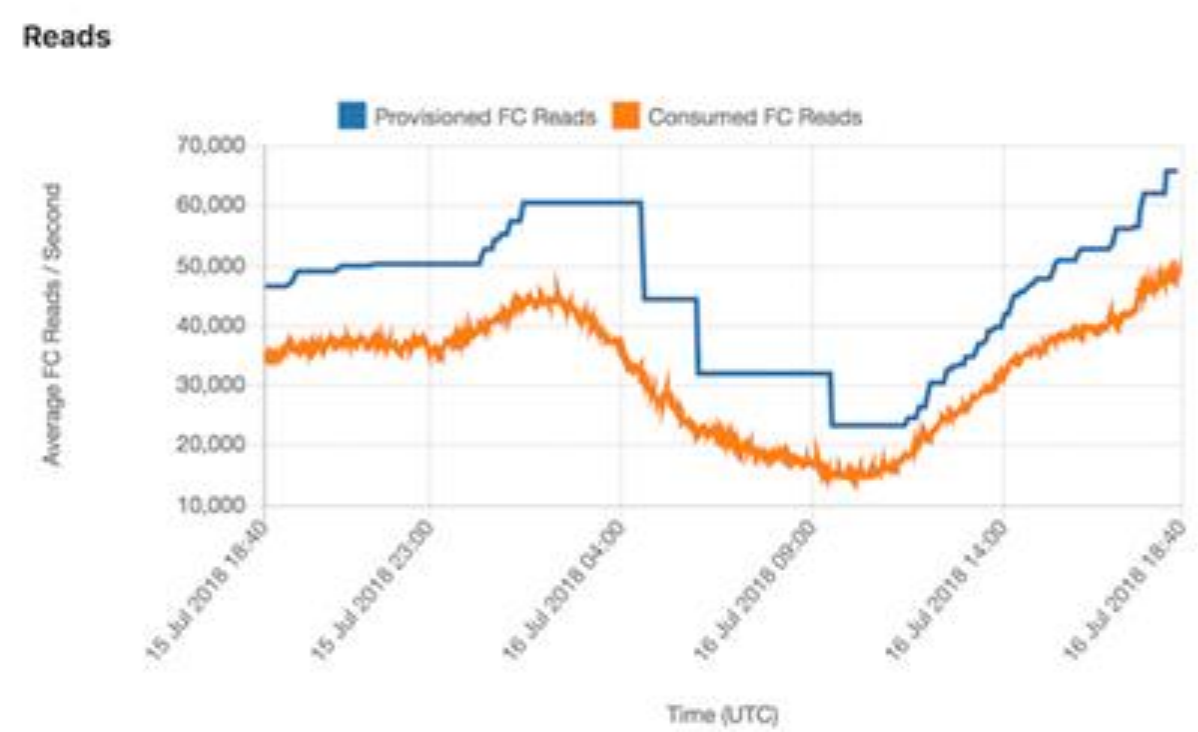
- Scaling is achieved through partitioning

    - Each virtual partition delivers 1000 writes/second AND 3000 reads/second
    - By Capacity = (Total RCU / 3000) + (Total WCU/1000)
    - By Size = Total Size / 10 GB

Provisioning Table Capacity



Auto Scaling



DynamoDB on-demand capacity mode

Features

- No capacity planning, provisioning, or reservations—simply make API calls
- Pay only for the reads and writes you perform

Key benefits

- Eliminates tradeoffs of overprovisioning or underprovisioning
- Instantly accommodates your workload as traffic ramps up or down

On-demand scaling properties

Base throughput

- Up to 4,000 write request units: 4,000 writes per second
- Up to 12,000 read request units: 24,000 eventually consistent reads per second
- Any linear combination of the two

Maximum throughput

Unlimited!

Pay per request: Use nothing, pay nothing

Use provisioned mode

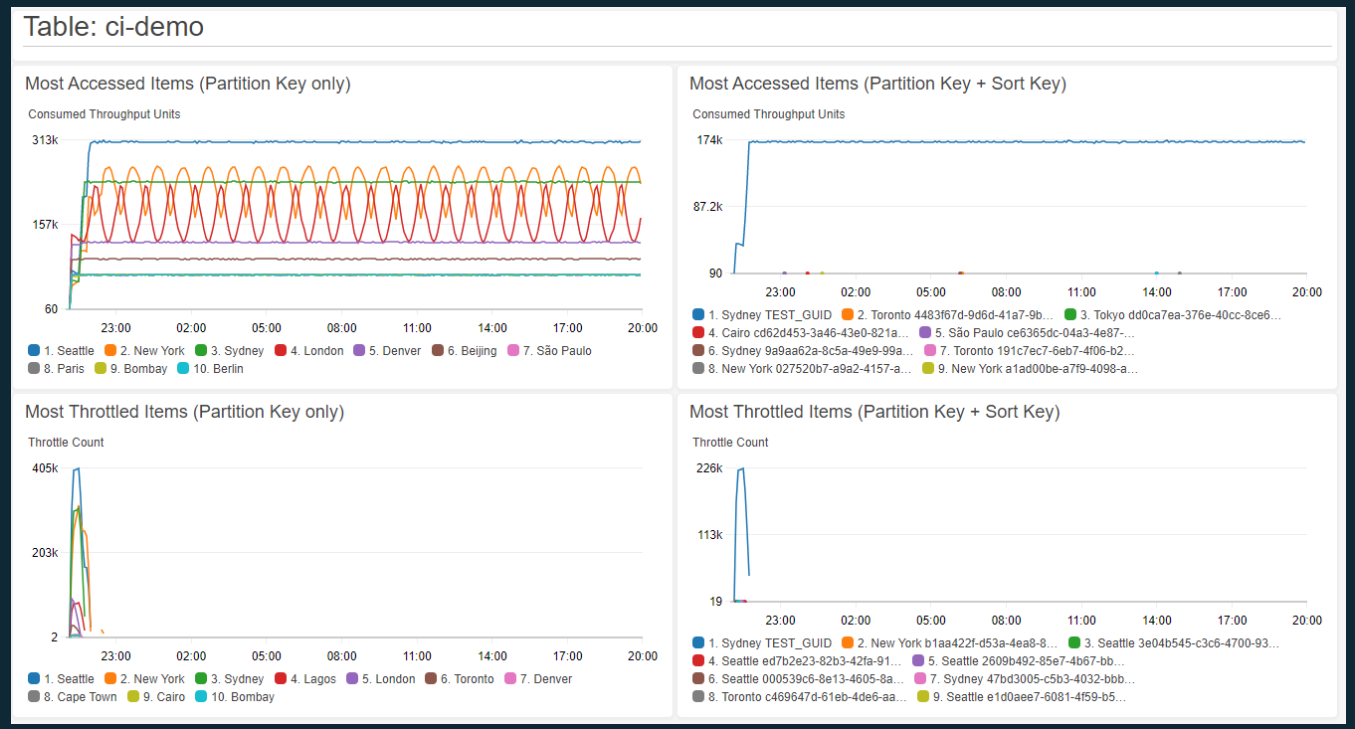
- Steady workloads
- Gradual ramps
- Events with known traffic
- Ongoing monitoring

Use on-demand mode

- Unpredictable workloads
- Frequently idle workloads
- Events with unknown traffic
- “Set it and forget it”

Consider your tolerance for operational overhead and overprovisioning

Amazon CloudWatch Contributor Insights for DynamoDB



Features

- Key-level activity graphs
- 1-click integration between DynamoDB and CloudWatch

Key benefits

- Identify frequently accessed keys and traffic trends at a glance
- Respond appropriately to unsuccessful requests

Feature Highlights

DynamoDB feature highlights

- 99.999% SLA
- DynamoDB Accelerator (DAX)
- Global tables
- DynamoDB Streams and Kinesis Data Streams support
- Auto scaling
- Adaptive capacity
- Time To Live (TTL)
- NoSQL Workbench
- Transactions
- Encryption at rest
- Point-in-time Recovery (PITR)
- On-demand backup and restore
- Export to Amazon S3
- Amazon CloudWatch Contributor Insights for DynamoDB
- Audit logging with AWS CloudTrail

Takeaways

trueIDC

REGIONAL
DATA CENTER &
CLOUD SERVICE
PROVIDER