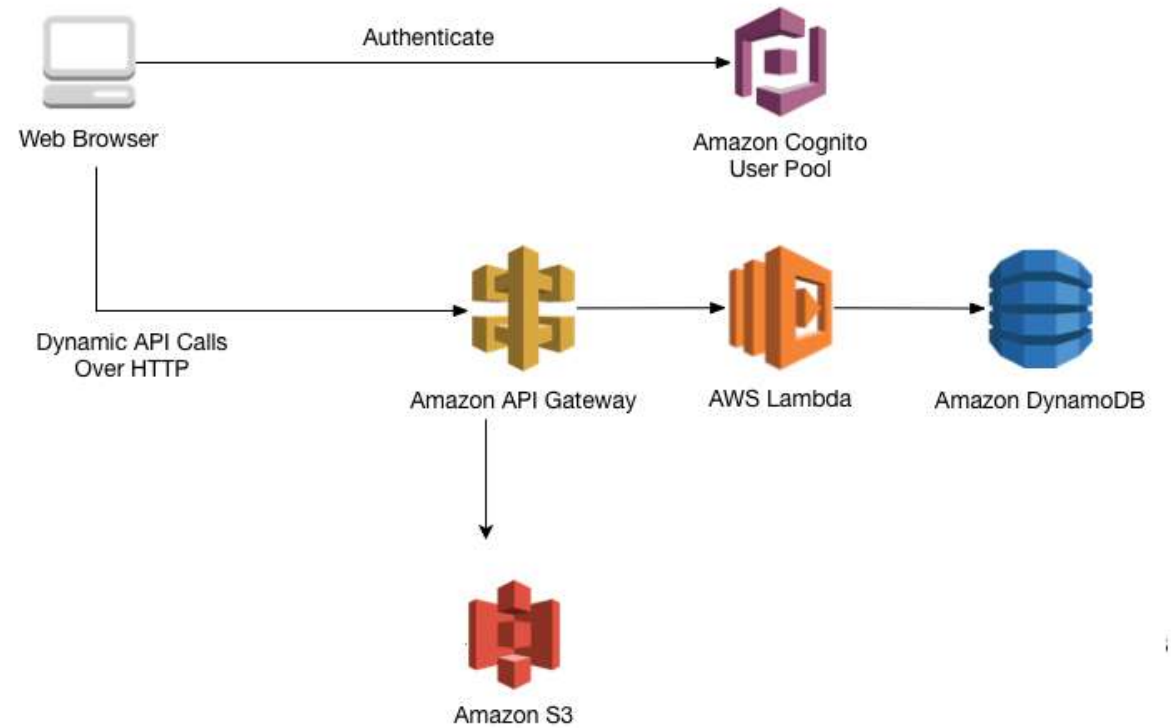


# Serverless

INTEGRATION OF MULTIPLE AWS SERVICES IN A WEB APPLICATION



# App Infrastructure



# DynamoDb

Fully  
managed  
NoSQL  
Database

High  
Availability  
& Durability

On-demand  
Backup and  
Restore

# DynamoDb Components



TABLES, ITEMS &  
ATTRIBUTES



PRIMARY KEY (SIMPLE  
AND COMPOSITE)



SECONDARY INDEXES



DYNAMODB STREAM

# Simple Storage Service (S3)



# S3 Components



Buckets



Objects



Keys



Regions

# S3 Features



## Storage classes

S3 STANDARD  
S3 STANDARD\_IA  
S3 Glacier



## Bucket policies



## Versioning

# Lambda



- ▶ Compute service for running code absent worry of managing servers or provisioning resources
- ▶ Integration with a multitude of AWS Services
  - ▶ S3
  - ▶ API Gateway
  - ▶ Step Functions
  - ▶ Events (EventBridge, DynamoDb Streams..)



# Lambda Concepts

- ▶ Function
- ▶ Qualifier
- ▶ Execution Environment
- ▶ Deployment Package
- ▶ Layer
- ▶ Runtime
- ▶ Extension
- ▶ Event
- ▶ Concurrency
- ▶ Trigger

# API Gateway



Creating, maintaining, publishing, monitoring and securing REST, HTTP and WebSocket APIs.

## RESTful APIs:

- ▶ HTTP-based
- ▶ Stateless
- ▶ Standard HTTP methods

## WebSocket APIs:

- ▶ Statefull
- ▶ Full-duplex
- ▶ Routing based on message content

# API Gateway Concepts

- ▶ Integration Request
- ▶ Integration Response
- ▶ Method Request
- ▶ Method Response
- ▶ Mapping Template
- ▶ Mock Integration
- ▶ Proxy Integration

# Step Functions



## Workflows:

- ▶ Standard
- ▶ Express

## Supports:

- ▶ Function orchestration (Linear)
- ▶ Branching
- ▶ Error handling (Retry, Catch)
- ▶ Parallel processing (can be dynamic)



# CloudWatch



Monitor AWS resources in real time.

- ▶ Metrics
- ▶ Execution Logs (Lambda, API Gateway, Step Functions...)
- ▶ Alarms
- ▶ Anomaly Detection

# Cognito



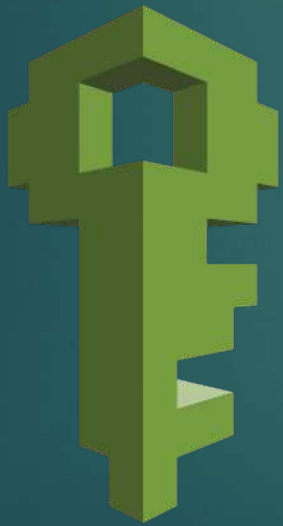
Provides:

- ▶ Authentication (local & federated)
- ▶ Authorization
- ▶ User management

Main Components:

- ▶ User pool
- ▶ Identity pool

# Identity and Access Management (IAM)



Securely control access to AWS resources using policies attached to IAM identities (users, groups of users or roles)

- ▶ Identity-based policies
- ▶ Resource-based policies