

API Gateway and Cognito

CS516 – Cloud Computing

Computer Science Department

Maharishi International University

Maharishi International University - Fairfield, Iowa



All rights reserved. No part of this slide presentation may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying or recording, or by any information storage and retrieval system, without permission in writing from Maharishi International University.

Content

- API Gateway
 - REST, HTTP, WebSocket APIs
 - CORS
 - HTTP proxy mode
 - Custom domain
 - Caching
 - Throttling
- Cognito
 - JWT token
 - OAuth, SAML
 - Amazon Cognito User pool
 - App client
 - Amazon Cognito Identity Pools (Federated Identities)

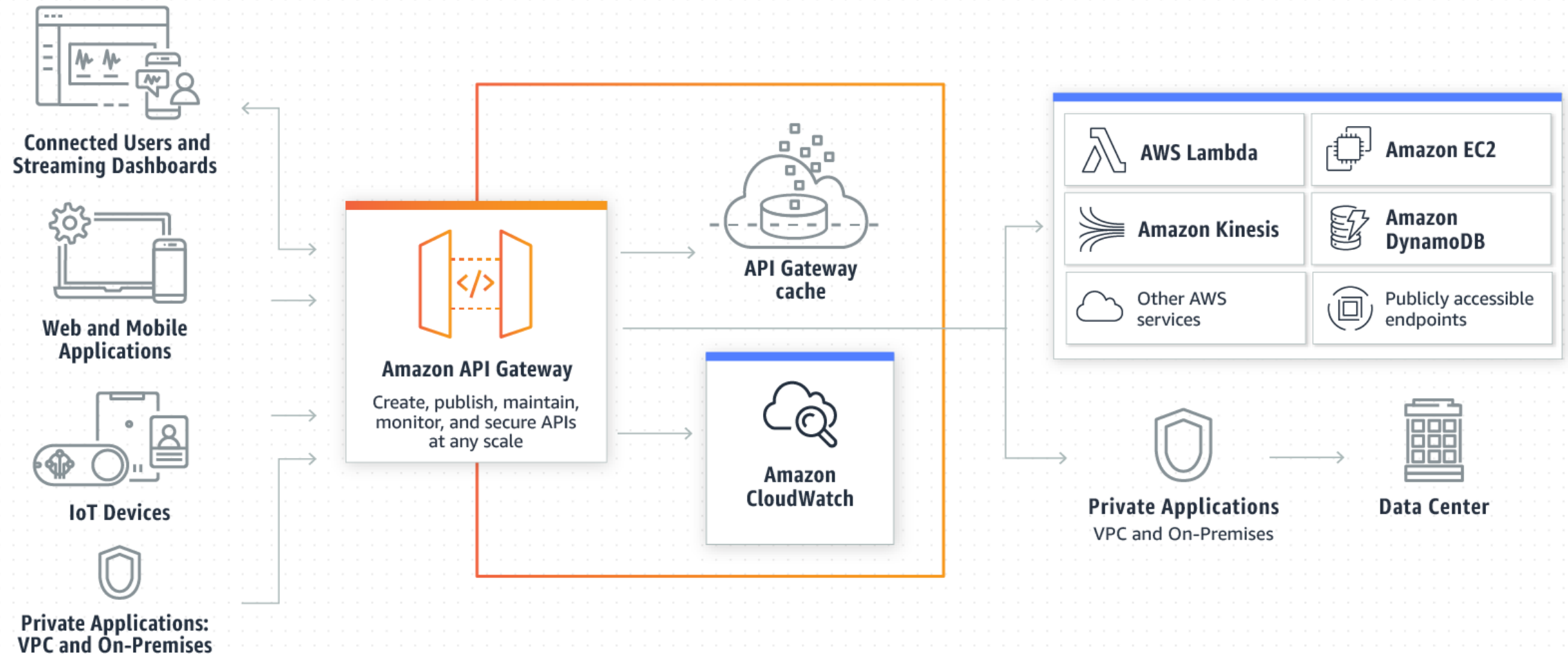
Amazon API Gateway

Amazon API Gateway is an AWS service for creating, publishing, maintaining, monitoring, and **securing** APIs at any scale. Depending on the region, it can handle 10,000 requests per second.

There are 3 types of APIs:

- REST – API uses HTTP verbs
- HTTP – Just REST API with some HTTP features such as [CORS](#).
- WebSocket – Real-time connection between client and server

How it works



Main benefits

- **AWS Authorization** – It helps you leverage signature version 4 authentication. You can use API Key, Lambda, JWT token.
- **API Keys for Third-Party Developers** – It helps you manage the ecosystem of third-party developers accessing your APIs. You can create API keys. You can also define plans that set **throttling** and request quota limits for each individual API key.
- **Resiliency** – It manages traffic to your backend systems by allowing you to set throttling rules. It handles all traffics so you can focus on business code rather than maintaining infrastructure. You can also setup cache in front of the API.

Other benefits

- **SDK Generation** – It can generate client SDKs for a number of platforms which you can use to quickly test new APIs from your applications and distribute SDKs to third-party developers.
- **API Lifecycle Management** – API Gateway lets you run multiple versions of the same API simultaneously so that applications can continue to call previous API versions even after the latest versions are published.
- **Easy API Creation and Deployment** – It can execute AWS Lambda code in your account, start AWS Step Functions, or make calls to ECS, EC2, or other web services outside of AWS with publicly accessible HTTP endpoints.
- **API Operations Monitoring** – It provides a dashboard to visually monitor calls to the services such as API calls, latency, and error rates through CloudWatch.

Rest API

Representational state transfer (REST) is a software architectural style which uses a subset of HTTP.

A Web service that follows these guidelines is called RESTful. Such a Web service must provide its **web resources** in a textual representation and allow them to be read and modified with a **stateless** protocol and a **predefined set of operations**.

There is a contract between the service provider and consumer

Rest API

Resources are in red. Each resource can have multiple http methods.

The screenshot displays the AWS API Gateway console interface. On the left sidebar, the 'APIs' section is expanded, showing 'API: PetStore' with a red vertical bar next to the 'Resources' link. The main panel is divided into three tabs: 'Resources', 'Actions', and 'Methods'. The 'Resources' tab is active, showing a tree structure of resources. Three resources are highlighted with red boxes: the root resource '/', the '/pets' resource, and the '/{petId}' resource. Each resource has a dropdown arrow to its left. Below each resource, the available HTTP methods are listed: GET for '/', GET and OPTIONS for '/pets', and GET and OPTIONS for '/{petId}'. The 'Methods' tab is also visible, showing the 'GET' method selected for the '/{petId}' resource. The 'Mock Endpoint' section is expanded, showing 'Authorization: None' and 'API Key: Not required'.

APIs

Custom Domain Names

VPC Links

API: **PetStore**

Resources

Stages

Resources **Actions** / Methods

/

GET

/pets

GET

OPTIONS

POST

/{petId}

GET

OPTIONS

GET

Mock Endpoint

Authorization **None**

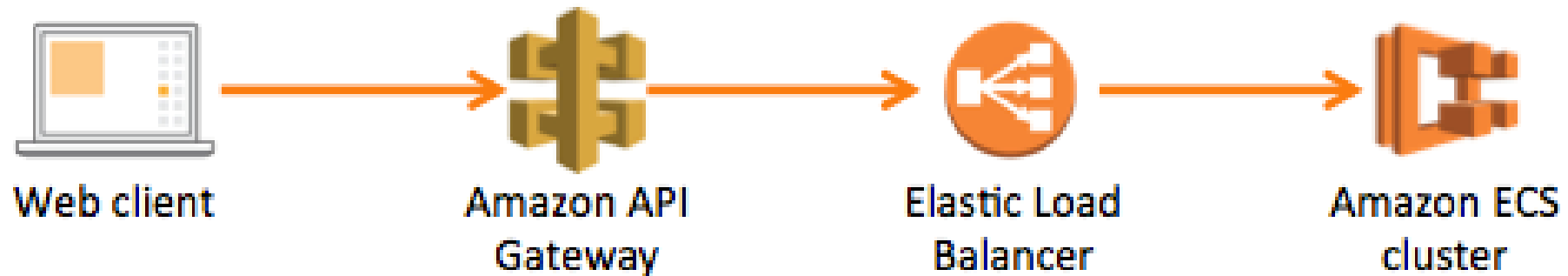
API Key Not required

You can run many other AWS services behind API Gateway

The screenshot shows the AWS API Gateway console interface. The breadcrumb navigation at the top indicates the path: **APIs** > **PetStore (y1i85rpt27)** > **Resources** > **/pets (3mdiwa)** > **GET**. The left-hand navigation pane includes sections for **APIs**, **Custom Domain Names**, **VPC Links**, and **API: PetStore**, with the **Resources** section currently selected. The main content area is titled **Method Execution /pets - GET - Integration Request** and contains the instruction: "Provide information about the target backend that this method will call and whether the incoming request data should be modified." The configuration fields visible are: **Integration type** (radio buttons for Lambda Function, HTTP, Mock, AWS Service, VPC Link, with **AWS Service** selected), **AWS Region** (dropdown menu set to us-east-1), **AWS Service** (dropdown menu with a list of services), **AWS Subdomain**, **HTTP method**, **Action Type**, **Path override (optional)**, **Execution role**, **Content Handling**, and **Use Default Timeout**. The **AWS Service** dropdown menu is open, displaying a list of services including Pinpoint, Polly, Redshift, Rekognition, Relational Database Service (RDS), Resource Groups, Route 53, Route 53 Domains, SageMaker, SageMaker Runtime, Secrets Manager, Security Token Service (STS), Serverless Application Repository, Service Catalog, Shield Advanced, Simple Email Service (SES), Simple Notification Service (SNS), Simple Queue Service (SQS), **Simple Storage Service (S3)** (highlighted in blue), and Simple Systems Management (SSM). Below the configuration fields, there are expandable sections for **URL Path Parameters**, **URL Query String Parameters**, and **HTTP Headers**.

HTTP proxy mode

Behind API gateway, you can run literally run any publicly accessible APIs on on-premise or in the cloud such as ALB, ECS, EC2.



Authorization

Authorizers enable you to control access to your APIs using Amazon Cognito User Pools or a Lambda function

Create Authorizer

Name *

Type * ⓘ

☐ Lambda

☒ Cognito

Cognito User Pool * ⓘ

us-east-1

Token Source * ⓘ

Token Validation ⓘ

Create

Cancel

Create Authorizer

Name *

Type * ⓘ

☒ Lambda

☐ Cognito

Lambda Function * ⓘ

us-east-1

Lambda Invoke Role ⓘ

Lambda Event Payload * ⓘ

☒ Token

☐ Request

Token Source* ⓘ

Token Validation ⓘ

Authorization Caching ⓘ

☒ Enabled

TTL (seconds)

300

Create

Cancel

Custom domain

Custom domain names are simpler and more intuitive URLs that you can provide to your API users.

Default: <https://api-id.execute-api.region.amazonaws.com/stage>

Custom: <https://api.example.com/myservice> or <https://myservice.api.example.com/>

After a custom domain name is created in API Gateway, you must create a record in AWS Route53 or your DNS provider.

Certificate you can issue with Amazon Certificate Manager (ACM).

Caching

You can enable API caching in Amazon API Gateway to cache your endpoint's responses. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of requests to your API.

When you enable caching, it caches responses from your endpoint for a specified time-to-live (TTL) period, in seconds.

- By default, TTL is 300 seconds (5 minutes)
- The maximum TTL is 3600 seconds (1 hour)
- The maximum size of a response that can be cached is 1 MB

Protect

To prevent your API from being overwhelmed by too many requests, Amazon API Gateway throttles requests to your API using the [token bucket algorithm](#).

It also limits the burst (the maximum number of concurrent requests) across all APIs within an AWS account, per Region. If it reaches the limit, clients get *429 Too Many Requests*.

Throttling

Route-level throttling limits for a specific stage or for individual routes in your API. It doesn't exceed account limit.

The rate limit is per second. If a client submits 10 requests (**rate limit**) **evenly in one second**, it will work fine.

But if it submits all of them **in the first millisecond**, it takes the first 5 requests (**burst limit**), and the rest gets 429 Too Many requests.

| Default route throttling | | Edit |
|--|------------|------|
| This throttling limit applies to each route in the stage except those defined for specific routes. | | |
| Burst limit | Rate limit | |
| 5 | 10 | |

| Account throttling | |
|--|------------|
| This throttling limit applies to the account and is shared by all APIs in this region. | |
| Burst limit | Rate limit |
| 5000 | 10000 |

Quotas

| Resource or operation | Default quota | Can be increased |
|---|---------------|------------------|
| Routes per API | 300 | Yes |
| Integrations per API | 300 | No |
| Maximum integration timeout | 30 seconds | No |
| Stages per API | 10 | Yes |
| Tags per stage | 50 | No |
| Total combined size of request line and header values | 10240 bytes | No |
| Payload size | 10 MB | No |
| Custom domains per account per Region | 120 | Yes |
| Access log template size | 3 KB | No |
| Amazon CloudWatch Logs log entry | 1 MB | No |
| Authorizers per API | 10 | Yes |
| Audiences per authorizer | 50 | No |
| Scopes per route | 10 | No |
| Timeout for JSON Web Key Set endpoint | 1500 ms | No |
| Response size from JSON Web Key Set endpoint | 150000 bytes | No |
| Timeout for OpenID Connect discovery endpoint | 1500 ms | No |
| VPC links per account per Region | 10 | Yes |
| Subnets per VPC link | 10 | Yes |
| Stage variables per stage | 100 | No |

Errors and solutions

- If you get the 403 (Missing Authentication Token), make sure you are hitting the right URL.
- You get the 500 error when the underlying service returns unexpected response. For example, the Lambda is not returning the statusCode and body in the JSON.
 - You can enable logging in CloudWatch in API Gateway Settings. That incurs an additional charge. Enable it only in the development. CloudWatch is a kinda expensive service. Your log can get too much when autoscaling keeps trying to start the app when the app has an error.

Know-how

- If you want to handle binary files, add it (“image/*”) in the API’s settings. Then in the req or res, select “Convert to binary if needed”. 10 MB max.
- Integration requests and responses are very important. With them, you can transform payload.
- Stage variable is an environment variable for the API
- In order to capture query strings or path params, you must define that in method req and res. Otherwise, it will omit them. API gateway doesn’t forward all headers. Even if your Lambda is sending the CORS headers, those headers are omitted. You must specify allowed header in the “Method Response”. Then write the actual values in the “Integration Response”.

Amazon Cognito

Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily.

Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and enterprise identity providers, such as Microsoft Active Directory.

With Amazon Cognito user pools groups, you can manage your users and their access to resources by mapping IAM roles to groups.

Amazon Benefits

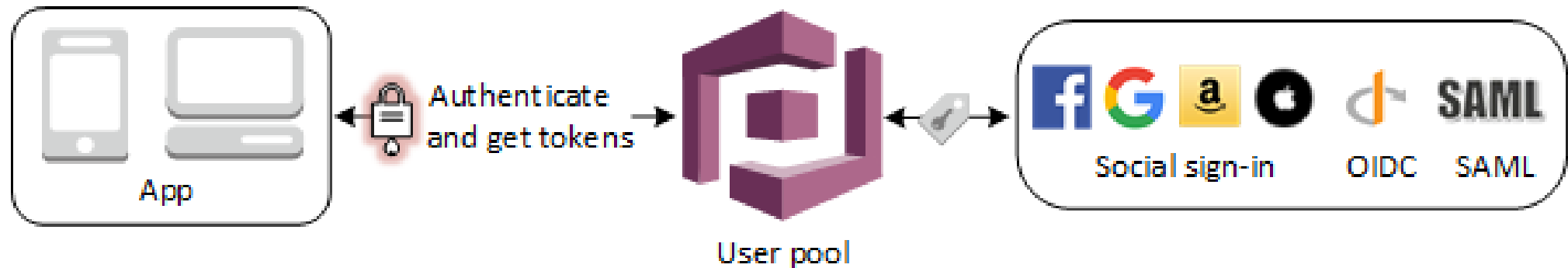
- **Scalable user directory** – Scales to hundreds of millions of users. No server or infrastructure to manage.
- **Social and enterprise identity federation** – Supports identity and access management standards, such as OAuth 2.0, SAML 2.0.
- **Security for your apps and users** – Multi-factor authentication and encryption of data-at-rest and in-transit.
- **Access control for AWS resources** – You can define roles and map users to different roles so your app can access only the resources that are authorized for each user.
- **Easy integration with your app** – With a built-in UI and easy configuration. You can add your branding. That built-in UI, you can use as a SSO to your internal applications. There is also API Amplify library, with that you can easily integrate your front-end app with Cognito.

Cognito searchable attributes

- username (case-sensitive)
- email
- phone_number
- name
- given_name
- family_name
- preferred_username
- cognito:user_status (called Status in the Console) (case-insensitive)

JWT token

After successfully authenticating a user, Amazon Cognito issues JSON web tokens (JWT) that you can use to secure and authorize access to your own APIs, or exchange for AWS credentials.



JWT token

A JWT is a structured security token format used to encode JSON data.

Using a JWT allows the token to be validated locally, without making an HTTP request back to the IdP, thereby increasing your application's performance.

Applications can make use of data inside the token, further reducing expensive HTTP calls and database lookups.

JWT can be stored in a shared caching server so applications can scale out easily as servers don't need to store user session.

Even if the hackers compromised the token, it is **temporary**.

JWT token stores user data

```
{
  "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "device_key": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "cognito:groups": [
    "admin"
  ],
  "token_use": "access",
  "scope": "aws.cognito.signin.user.admin",
  "auth_time": 1562190524,
  "iss": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_example",
  "exp": 1562194124,
  "iat": 1562190524,
  "jti": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "client_id": "57cbishk4j24pabc1234567890",
  "username": "janedoe@example.com"
}
```

OAuth 2.0

OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.

This mechanism is used by companies such as Amazon, Google, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites.

SAML 2.0

Security Assertion Markup Language (SAML) is an open standard for exchanging authentication and authorization data between parties, in particular, between an identity provider and a service provider.

SAML is an XML-based markup language for security assertions. Used commonly for **enterprise users**.

AWS supports identity federation with SAML 2.0 that enables federated single sign-on (**SSO**), so users can log into the AWS Management Console or call the AWS API operations without you having to create an IAM user for everyone in your organization.

User Pool App Client

You can configure an app client for accessing Amazon Cognito from your application through SDK.

The client app either can be a public front-end or a private back-end app.

Generate the **client secret** for back-end apps that make call to your Cognito user pool. Never generate a token for a front-end app since the app is **public** anyone can see the entire code.

User Pool App Client Token types

There are 3 tokens in user pool app client:

- Refresh token - Refresh Tokens are credentials used to obtain access tokens
- **ID token** - The ID Token is a security token granted by the OpenID Provider that contains information about an End-User. This information tells your client application that the user is authenticated, and can also give you information like their username or locale.
- Access token (Authorization) - Access tokens, on the other hand, are not intended to carry information about the user. They simply allow access to certain defined server resources.

Customizing User Pool Workflows with Lambda Triggers

| User Pool Flow | Operation | Description |
|-------------------------------------|-------------------------------------|---|
| Custom Authentication Flow | Define Auth Challenge | Determines the next challenge in a custom auth flow |
| | Create Auth Challenge | Creates a challenge in a custom auth flow |
| | Verify Auth Challenge Response | Determines if a response is correct in a custom auth flow |
| Authentication Events | Pre Authentication Lambda Trigger | Custom validation to accept or deny the sign-in request |
| | Post Authentication Lambda Trigger | Event logging for custom analytics |
| | Pre Token Generation Lambda Trigger | Augment or suppress token claims |
| Sign-Up | Pre Sign-up Lambda Trigger | Custom validation to accept or deny the sign-up request |
| | Post Confirmation Lambda Trigger | Custom welcome messages or event logging for custom analytics |
| | Migrate User Lambda Trigger | Migrate a user from an existing user directory to user pools |
| Messages | Custom Message Lambda Trigger | Advanced customization and localization of messages |
| Token Creation | Pre Token Generation Lambda Trigger | Add or remove attributes in Id tokens |
| Email and SMS third-party providers | Custom sender Lambda triggers | Use a third-party provider to send SMS and email messages |

Read: [Amazon Cognito Triggers: all you need to know about them](#)

Using Amazon Pinpoint Analytics with Amazon Cognito User Pools

Amazon Cognito User Pools are integrated with Amazon Pinpoint to provide analytics for Amazon Cognito user pools and to enrich the user data for Amazon Pinpoint campaigns.

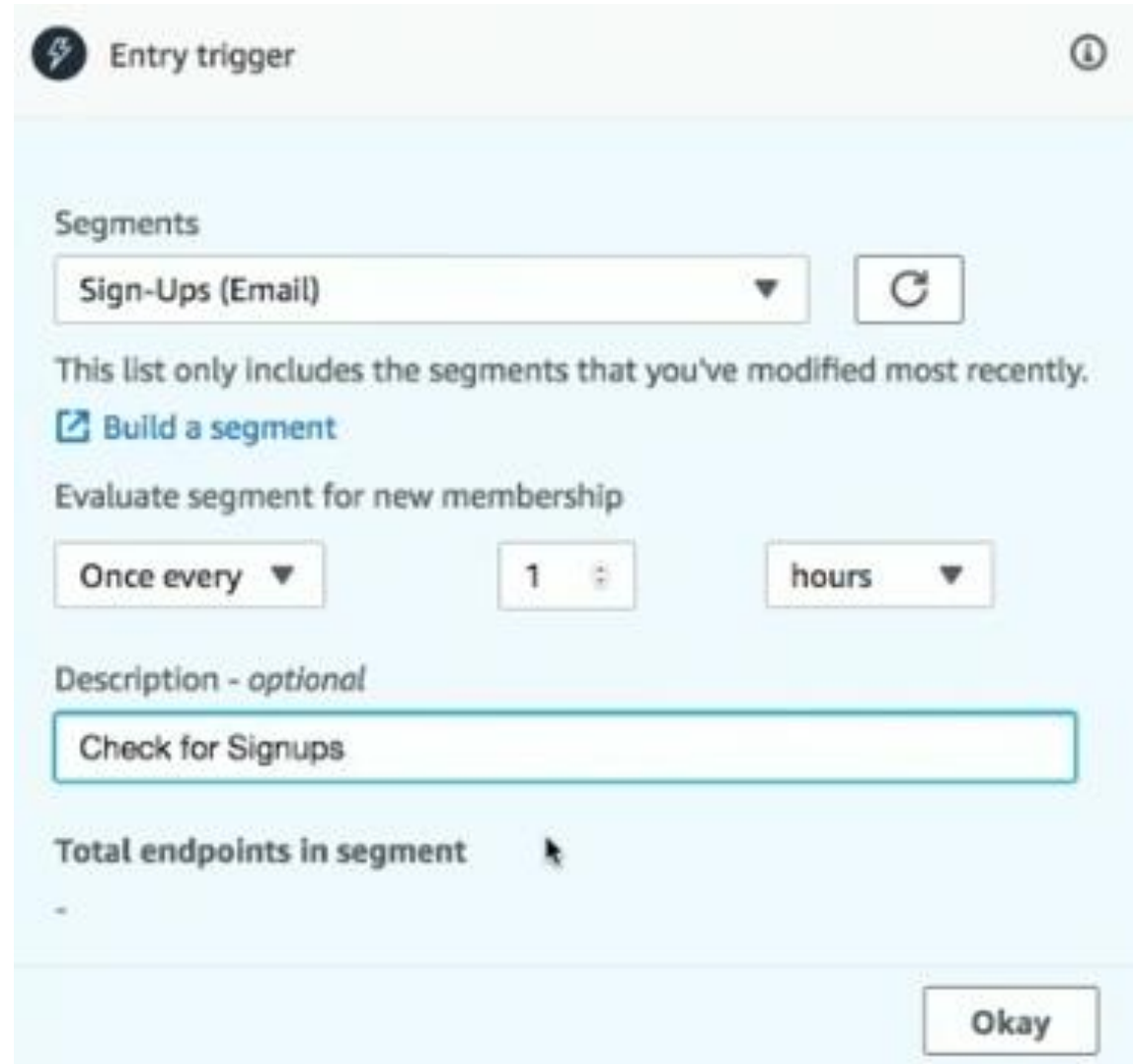

Amazon Pinpoint provides analytics and targeted campaigns to drive user engagement in mobile apps using push notifications.

You can drill into the data for different date ranges or attributes, such as device platform, device locale, and app version.


For example, send a message to users asking them to buy today's special dinner after work at 5 pm.


Amazon Pinpoint Example

Check every hour if there are new users. Then send them a welcome email.

 Entry trigger 

Segments


Sign-Ups (Email) 





This list only includes the segments that you've modified most recently.

[Build a segment](#)

Evaluate segment for new membership

Once every 

1 

hours 

Description - optional

Check for Signups

Total endpoints in segment

-

Okay

Amazon Pinpoint Example

If a user opened the welcome email, then do this; otherwise do that after 2 days.

Full demo: [Amazon Pinpoint Journeys](#)

The screenshot shows the 'Yes/No split' configuration window in the Amazon Pinpoint console. The window has a title bar with a green icon and the text 'Yes/No split'. Below the title bar, there are several sections: 'Select a condition type' with a dropdown menu set to 'Event'; 'Events' with a dropdown menu set to 'Email open'; 'Choose an activity' with a dropdown menu set to 'Send Welcome Email'; a '+ Add condition' link; 'Condition evaluation' with a description 'The amount of time that Amazon Pinpoint waits before it evaluates the condition.' and a dropdown menu set to 'Evaluate after'; a numeric input field set to '2' and a unit dropdown menu set to 'days'; a 'Description - optional' text area with the placeholder text 'Add description I'; and an 'Okay' button at the bottom right. Below the window, there are two colored boxes: a green box labeled 'Yes' and a red box labeled 'No', connected by lines to the 'Yes/No split' window.

Yes/No split

Select a condition type

Event

Events

Email open

Choose an activity

Send Welcome Email

+ Add condition

Condition evaluation

The amount of time that Amazon Pinpoint waits before it evaluates the condition.

Evaluate after

2 days

Description - optional

Add description I

Okay

Yes No

Amazon Cognito Identity Pools

Currently, your users can do all actions with JWT token. You want to make it more fine-grained. For example, admin user can write and other users only read. That is when the identity pools comes into the picture.

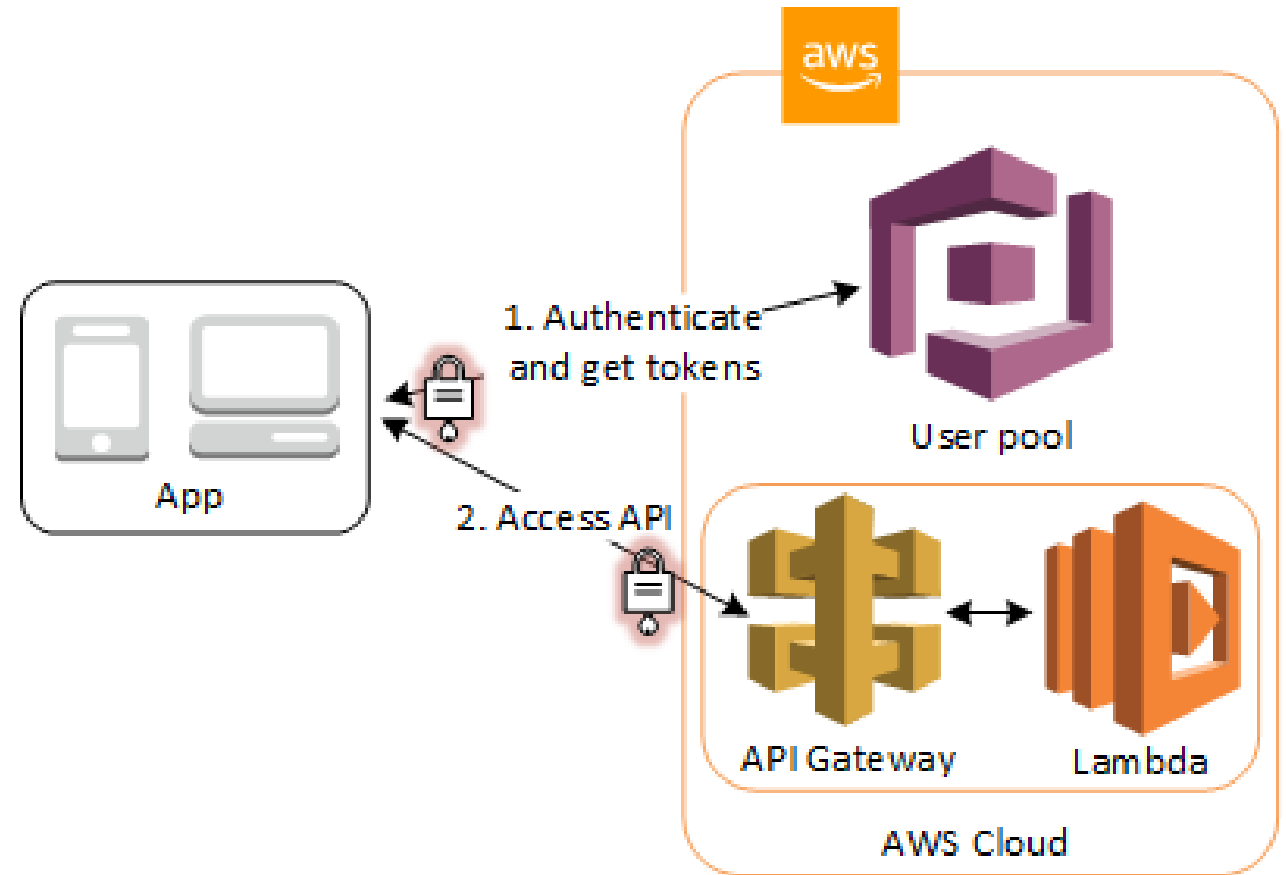
Amazon Cognito identity pools (federated identities) enable you to create unique identities for your users and federate them with identity providers. With an identity pool, you can obtain temporary, limited-privilege AWS credentials to access other AWS services.

An IAM role defines the permissions for your users to access AWS resources, like Amazon Cognito Sync. Users of your application will assume the roles you create. You can specify different roles for authenticated and unauthenticated users.

Please watch this demo: [Fine-grained Access Control with Amazon Cognito Identity Pools](#)

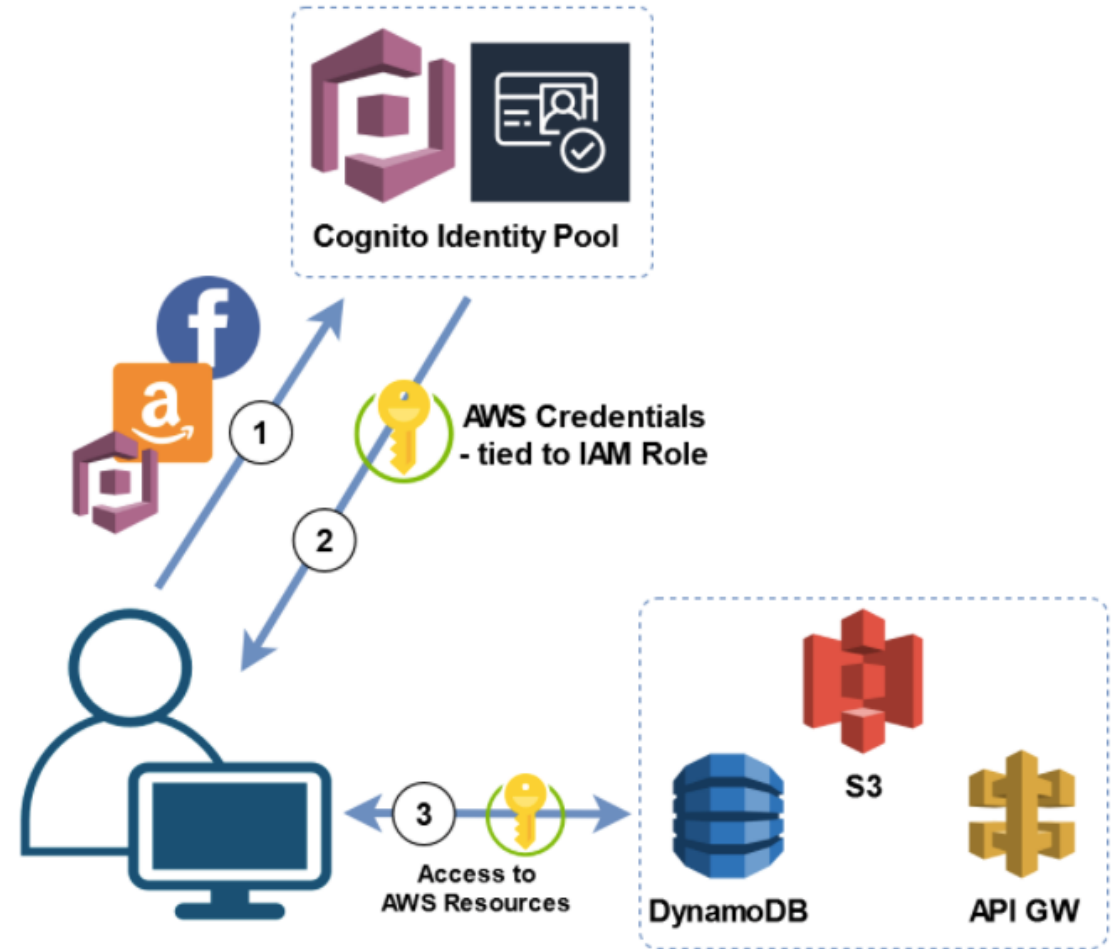
Without identity pool

One token for all users. Then the app has the IAM role that has all permissions.



With identity pool

The IAM role is tied with the user, not app. Some users will have read only access whereas admins will have all permissions. That is known as fine-grained access.



Pricing

| Pricing Tier (MAUs) | Price per MAU |
|-------------------------|---------------|
| First 50,000 | Free |
| Next 50,000 | \$0.00550 |
| Next 900,000 | \$0.00460 |
| Next 9,000,000 | \$0.00325 |
| Greater than 10,000,000 | \$0.00250 |