API Gateway and Cognito

CS516 – Cloud Computing

Computer Science Department

Maharishi International University

Maharishi International University - Fairfield, Iowa

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)

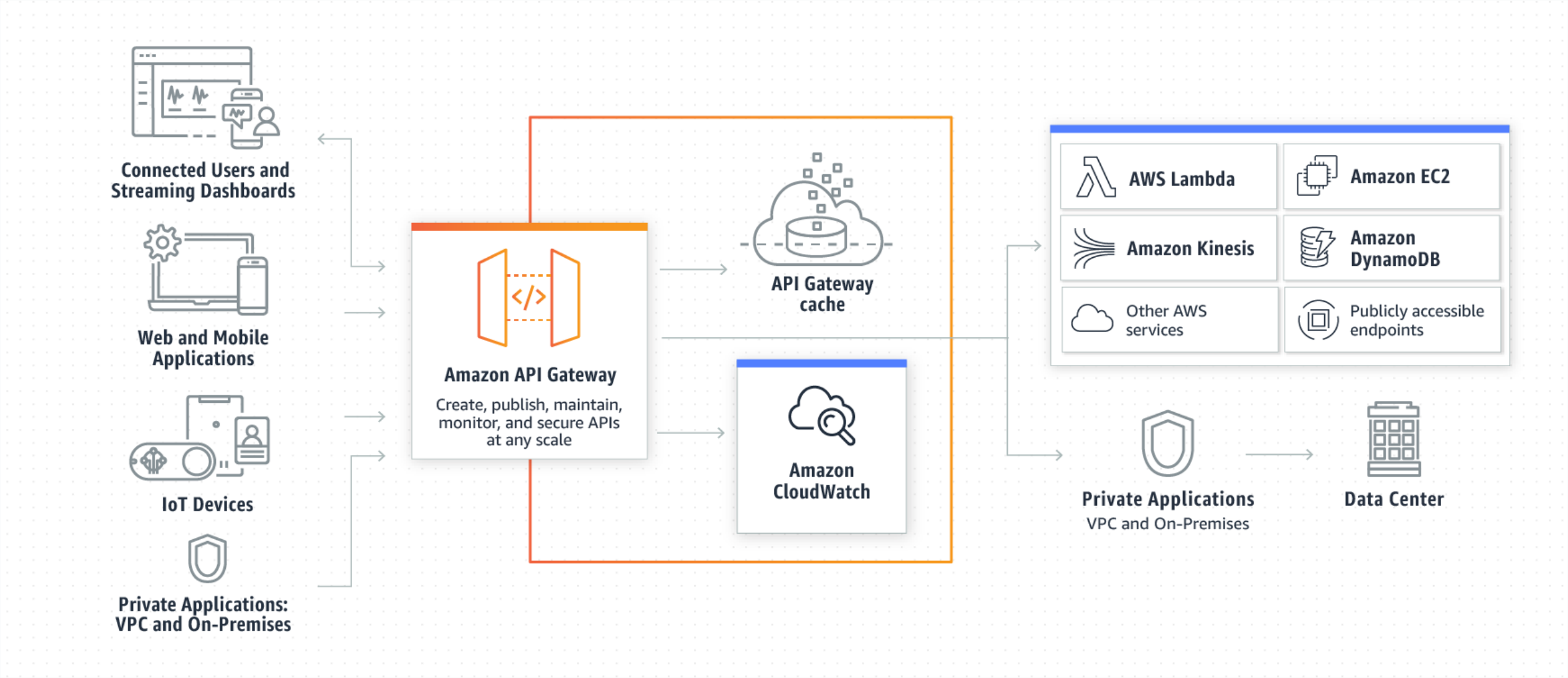
Amazon API Gateway

Amazon API Gateway is an AWS service for creating, publishing, maintaining, monitoring, and securing APIs at any scale.

There are 3 types of APIs:

- REST
- HTTP
- WebSocket

How it works


Connected Users and Streaming Dashboards
Web and Mobile Applications
IoT Devices
Private Applications: VPC and On-Premises
Amazon API Gateway
Create, publish, maintain, monitor, and secure APIs at any scale
API Gateway cache
Amazon CloudWatch
AWS Lambda
Amazon EC2
Amazon Kinesis
Amazon DynamoDB
Other AWS services
Publicly accessible endpoints
Private Applications
VPC and On-Premises
Data Center

Amazon API Gateway benefits

- **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.
- **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.

Amazon API Gateway benefits

- **AWS Authorization** – It helps you leverage signature version 4 authentication. You can use IAM, 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.
- **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.

Rest API

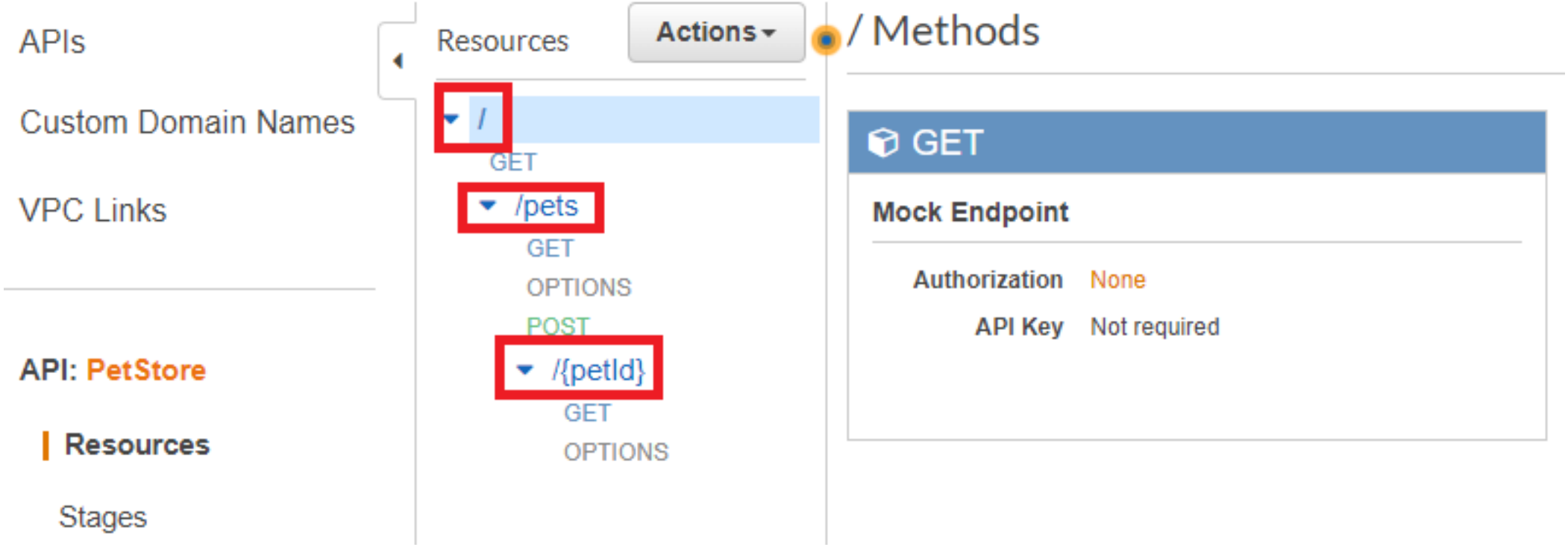
Representational state transfer (REST) is a software architectural style which uses a subset of HTTP.

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.



HTTP API

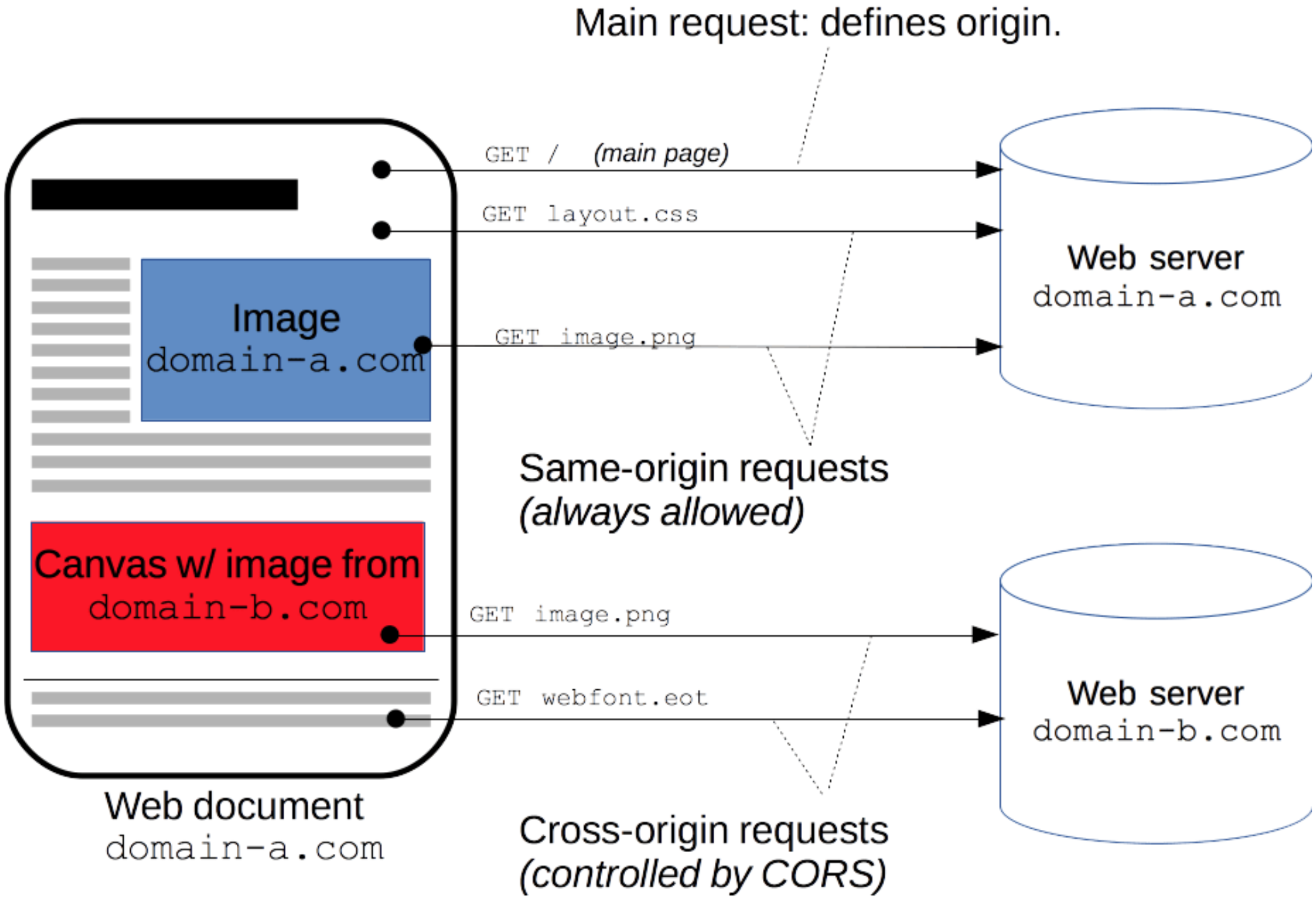
Build low-latency and cost-effective REST APIs with built-in features such as OIDC and OAuth2, and native CORS support.

For more info: Choosing between HTTP APIs and REST APIs

What is CORS?

CORS (Cross-Origin Resource Sharing) is a security mechanism on web browsers that allows other origins (domains) load its resources. For example, if you are accessing from your *localhost* to the API behind AWS API gateway, it won’t allow the request unless you enable CORS.

Browsers make an implicit call with OPTIONS method before sending the actual request. If the web server (in this case AWS API Gateway) enabled CORS, the request will be sent. Otherwise, it throws CORS error.


Main request: defines origin.
GET / (main page)
GET layout.css
GET image.png
Image
domain-a.com
Web server
domain-a.com
Same-origin requests
(always allowed)
Canvas w/ image from
domain-b.com
GET image.png
GET webfont.eot
Web server
domain-b.com
Web document
domain-a.com
Cross-origin requests
(controlled by CORS)

WebSocket API

The WebSocket API is an advanced technology that makes it possible to open a **two-way** interactive communication **session** (persistent connections) between the user's browser and a server for real-time use cases such as chat applications or dashboards.

You can run other AWS services behind API Gateway

Amazon API Gateway Features | / | api gateway and ecs - Google Se | API Gateway

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/y1i85rpt27/resources/3mdiwa/methods/GET

aws Services ▼ Search for services, features, marketplace products, and docs [Alt+S] utumenbayar_miu ▼ N. Vi

Amazon API Gateway | APIs > PetStore (y1i85rpt27) > Resources > /pets (3mdiwa) > GET | Show

APIs
Custom Domain Names
VPC Links

API: PetStore
- Resources
- Stages
- Authorizers
- Gateway Responses
- Models
- Resource Policy
- Documentation
- Settings

Usage Plans
API Keys
Client Certificates
Settings

Resources | Actions

- /
  - GET
  - /pets
    - GET
    - OPTIONS
    - POST
    - /{petId}
      - GET
      - OPTIONS

← Method Execution /pets - GET - Integration Request

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type
- ○ Lambda Function
- ○ HTTP
- ○ Mock
- ◉ AWS Service
- ○ VPC Link

AWS Region us-east-1

AWS Service Simple Storage Service (S3)

AWS Subdomain

HTTP method

Action Type

Path override (optional)

Execution role

Content Handling

Use Default Timeout

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)
Simple Systems Management (SSM)

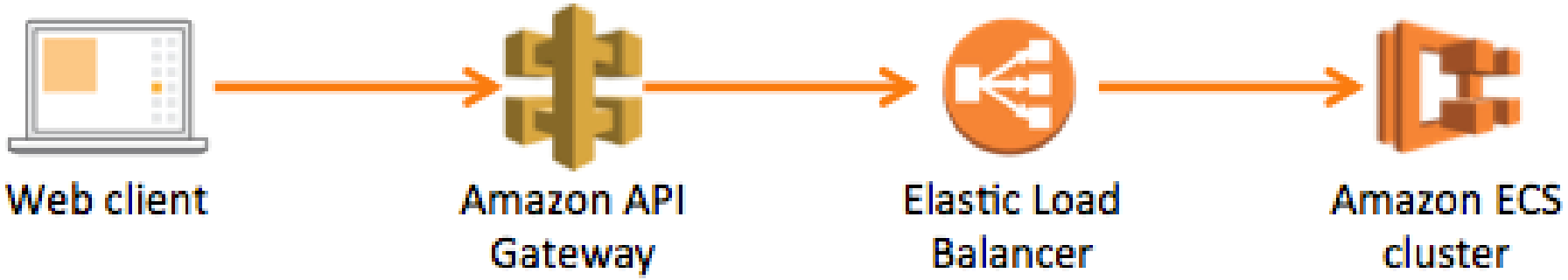
▸ URL Path Parameters

▸ URL Query String Parameters

▸ HTTP Headers

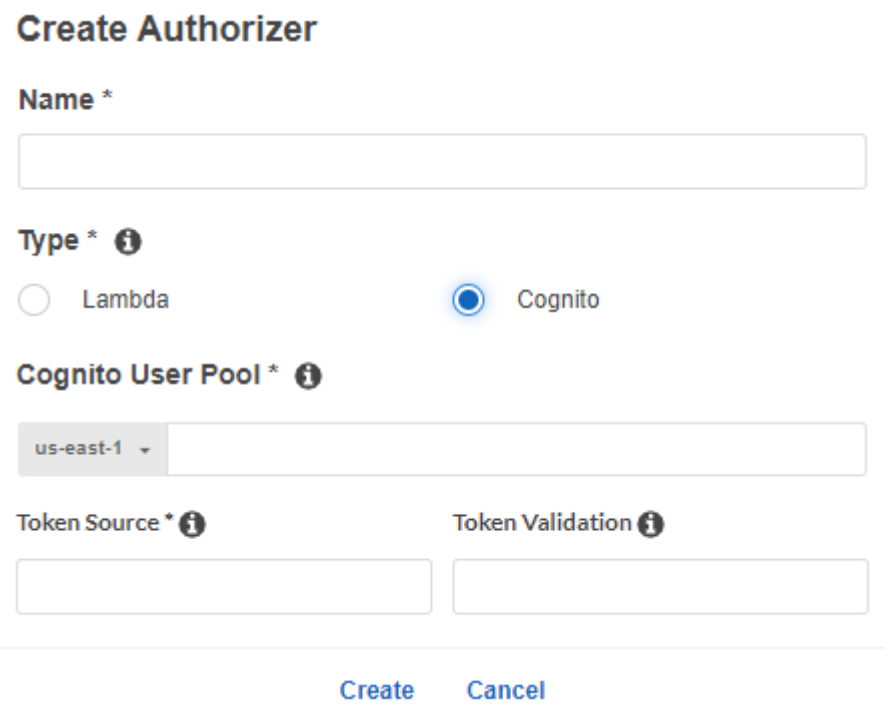
HTTP proxy mode

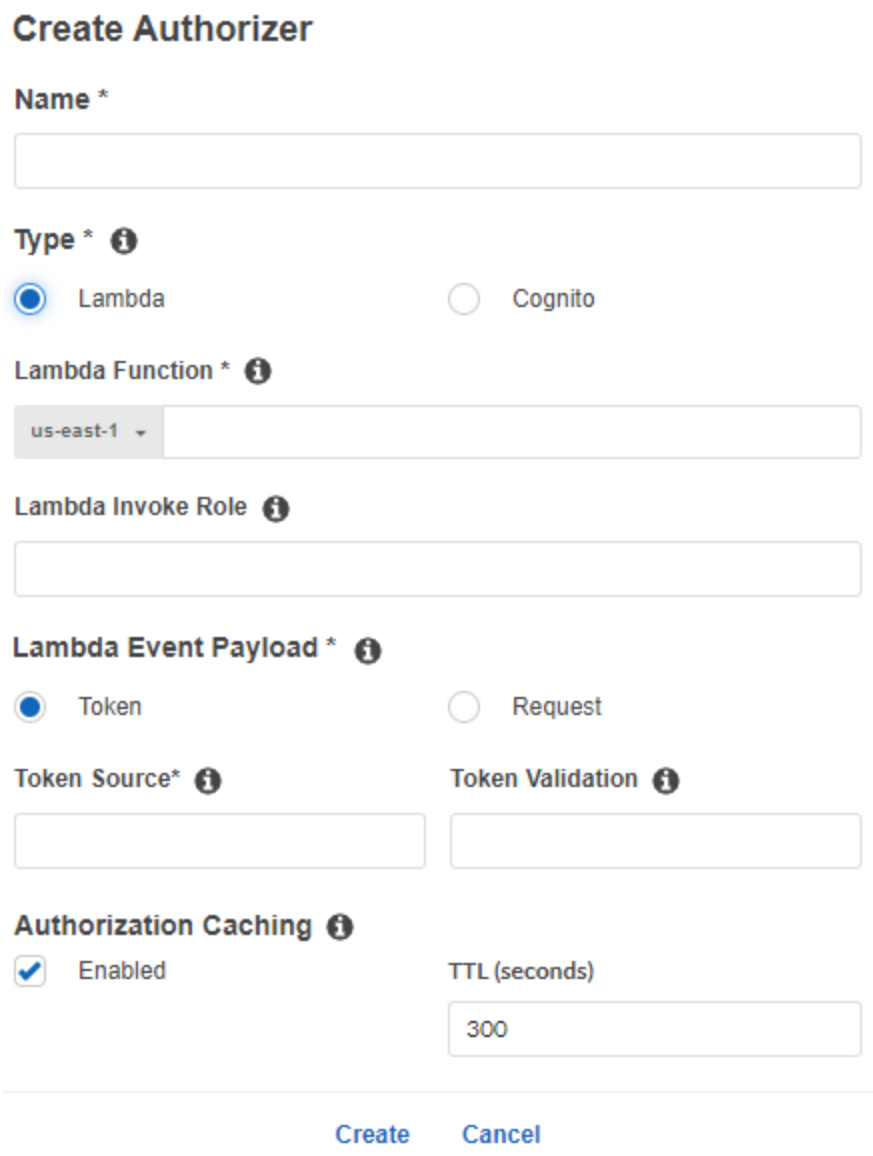
Amazon API Gateway can make proxy calls to any **publicly accessible** endpoint; for example, an Elastic Load Balancer endpoint in front of a microservice that is deployed on Amazon EC2 or ECS.



Authorization

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





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.

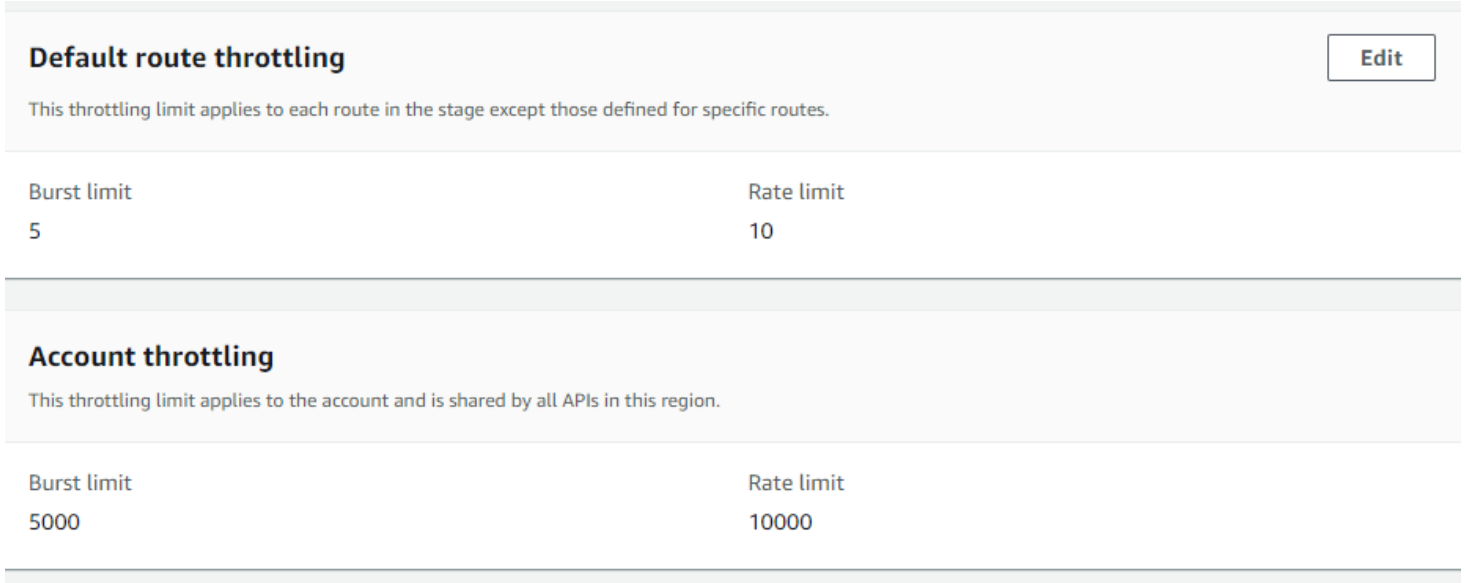
It also limits the burst (the maximum number of concurrent requests) across all APIs within an AWS account, per Region. If it reaches the limit, clients get *429 Too Many Requests*.

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.



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

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.

Amazon Cognito user pools

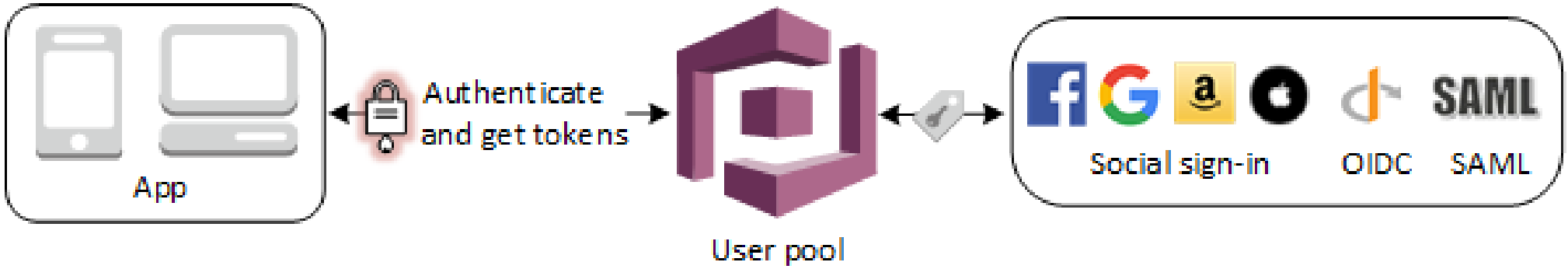
A user pool is a user directory in Amazon Cognito.

User pools provide:

- Sign-up and sign-in services.
- A built-in, customizable web UI to sign in users.
- Social sign-in with Facebook, Google, Amazon, Apple, as well as sign-in with SAML identity providers from your user pool.
- User directory management and user profiles.
- Security features such as multi-factor authentication (MFA), checks for compromised credentials, account takeover protection, and phone and email verification.
- Customized workflows and user migration through AWS Lambda triggers.

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

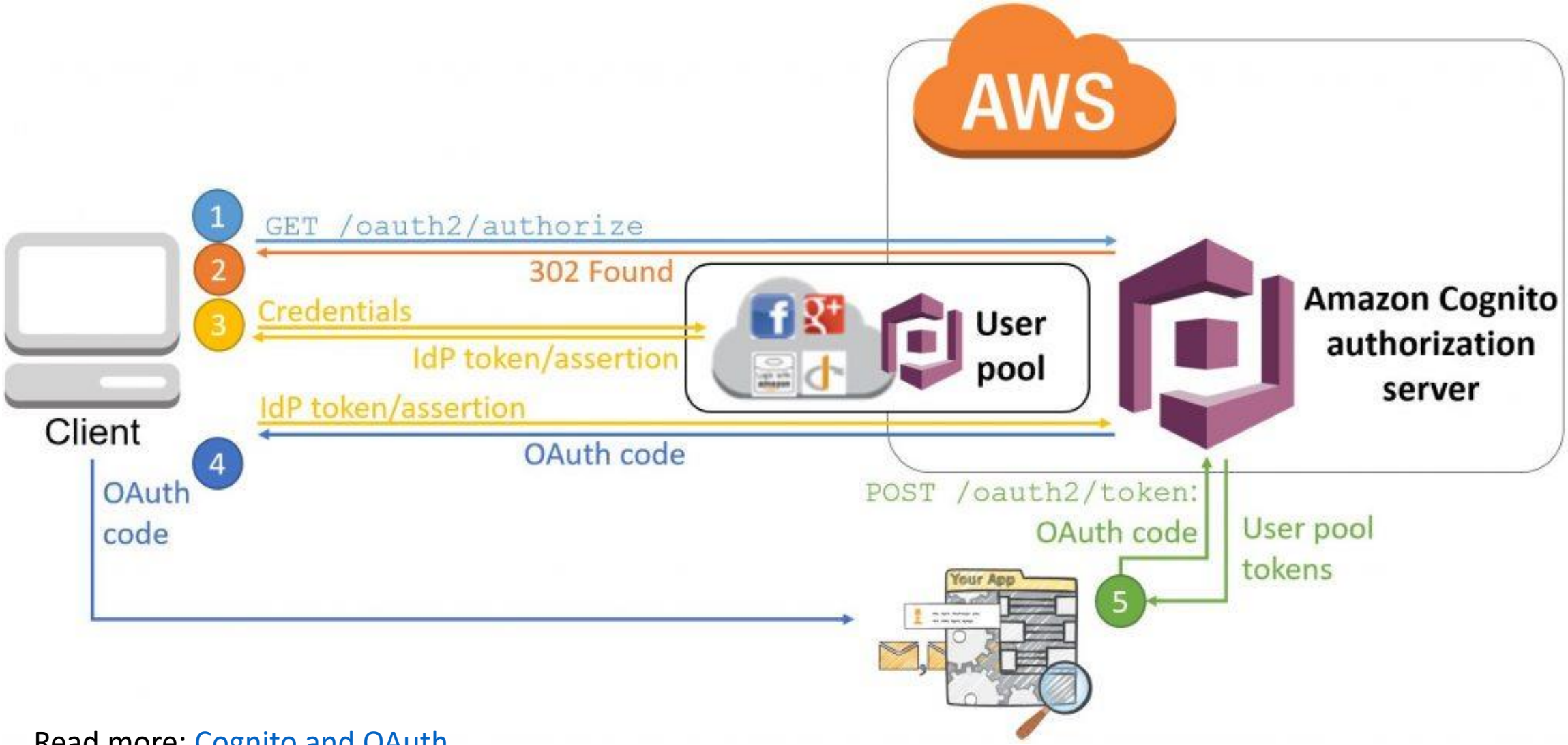

```
{
  "sub": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "device_key": "aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee",
  "cognito:groups": [
    "admin"
  ],
  "token_use": "access",
  "scope": "aws.cognito.signin.user.admin",
  "auth_time": 1562190524,
  "iss": "https://cognito-idp.us-west-2.amazonaws.com/us-west-2_example",
  "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.

Accessing AWS via OAuth



Read more: [Cognito and OAuth]

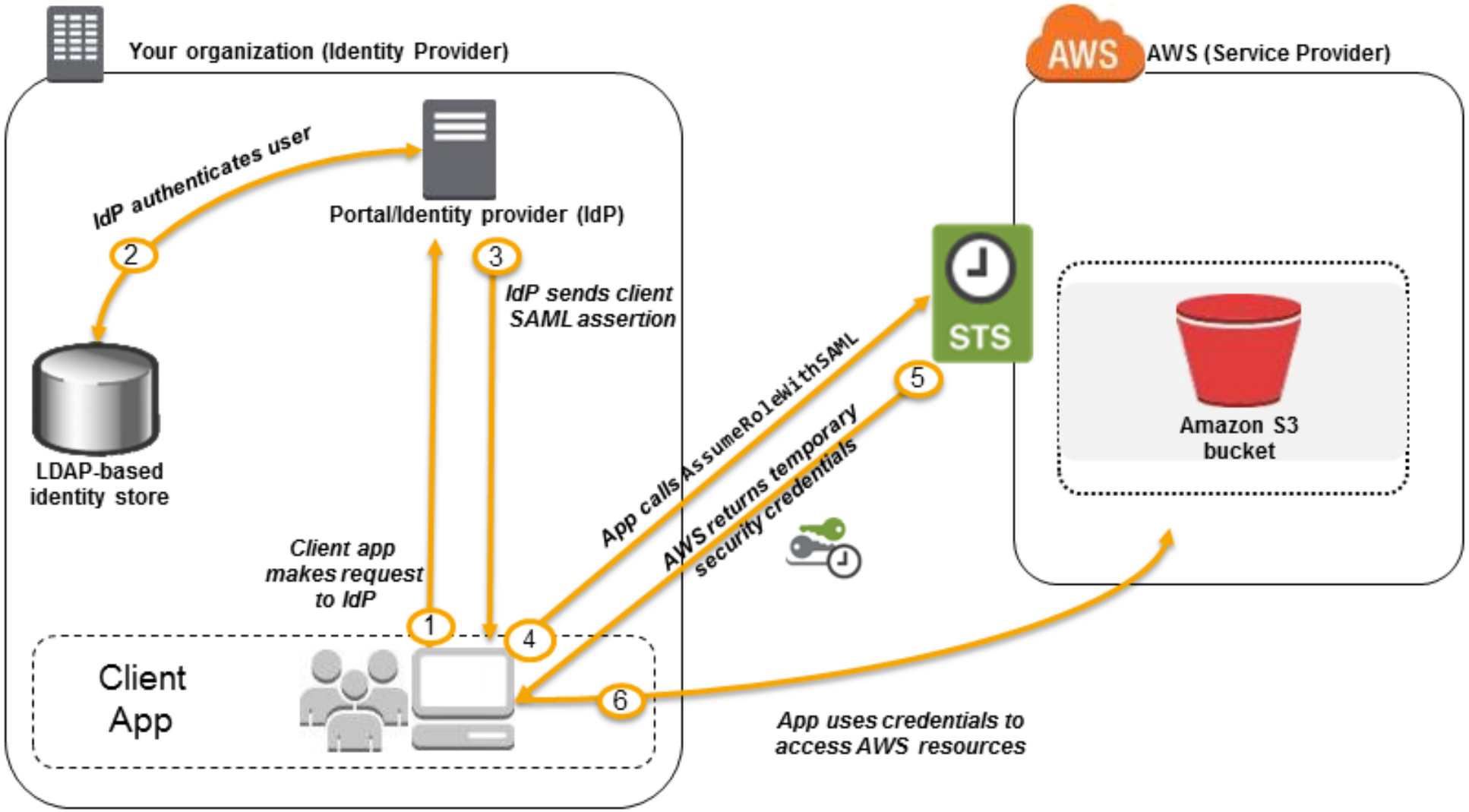
SAML 2.0

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.

Accessing AWS via SAML



User Pool App Client

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

You can also generate the **client secret** that is used by only application and authentication server (or another app), not application and user! Never issue a client secret for public apps. Instead, use only when authenticating microservice to microservice communication.

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.

App client name

my-app-client

Refresh token expiration

30 days and 0 minutes

Must be between 60 minutes and 3650 days

Access token expiration

0 days and 60 minutes

Must be between 5 minutes and 1 day. Cannot be greater than refresh token expiration

ID token expiration

0 days and 60 minutes

Must be between 5 minutes and 1 day. Cannot be greater than refresh token expiration

☐ Generate client secret

Auth Flows Configuration

☐ Enable username password auth for admin APIs for authentication (ALLOW_ADMIN_USER_PASSWORD_AUTH) Learn more.

☐ Enable lambda trigger based custom authentication (ALLOW_CUSTOM_AUTH) Learn more.

☑ Enable username password based authentication (ALLOW_USER_PASSWORD_AUTH) Learn more.

☑ Enable SRP (secure remote password) protocol based authentication (ALLOW_USER_SRP_AUTH) Learn more.

☑ Enable refresh token based authentication (ALLOW_REFRESH_TOKEN_AUTH) Learn more.

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

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.

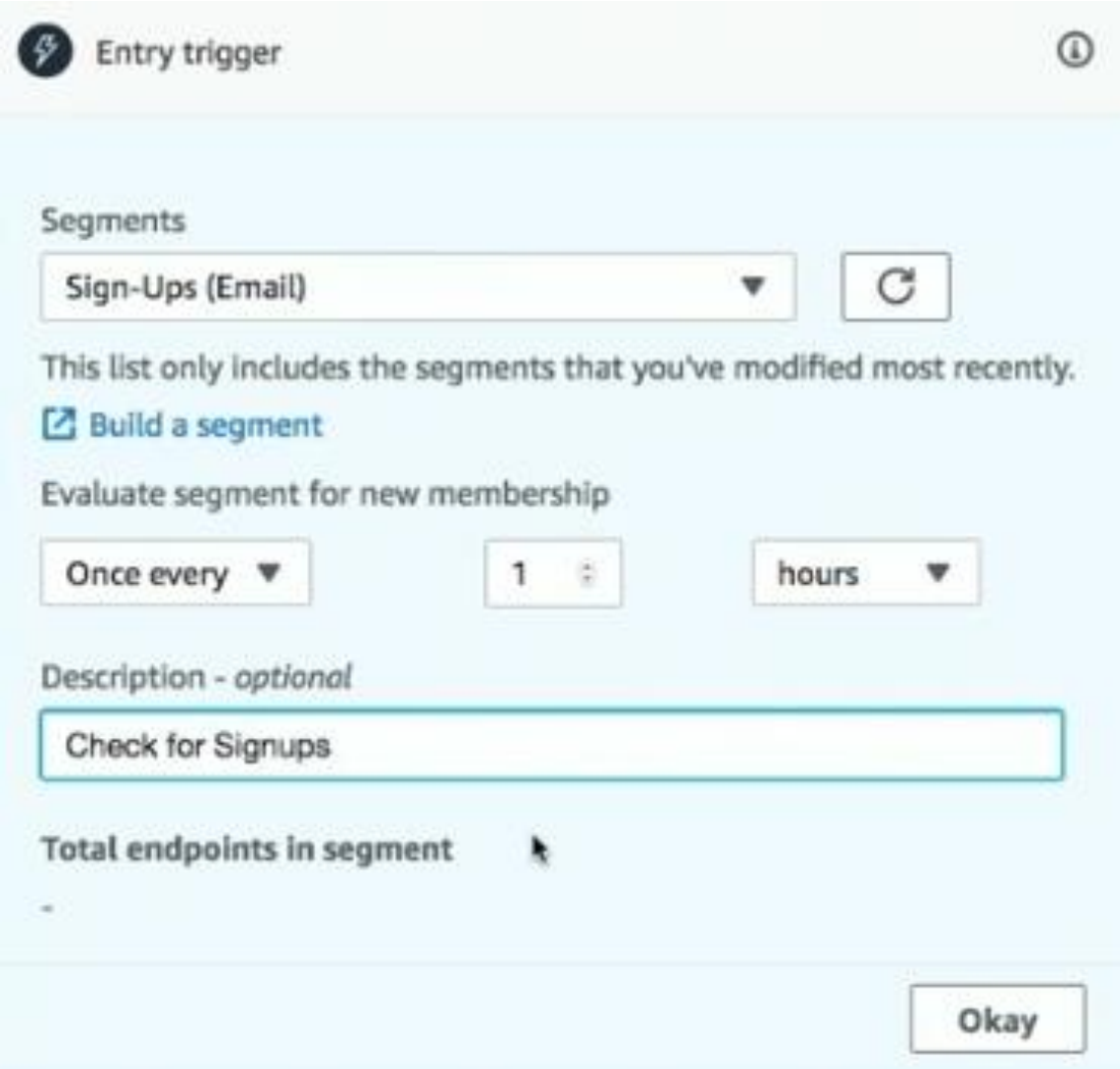
Amazon Pinpoint provides analytics and targeted campaigns to drive user engagement in mobile apps using push notifications.

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

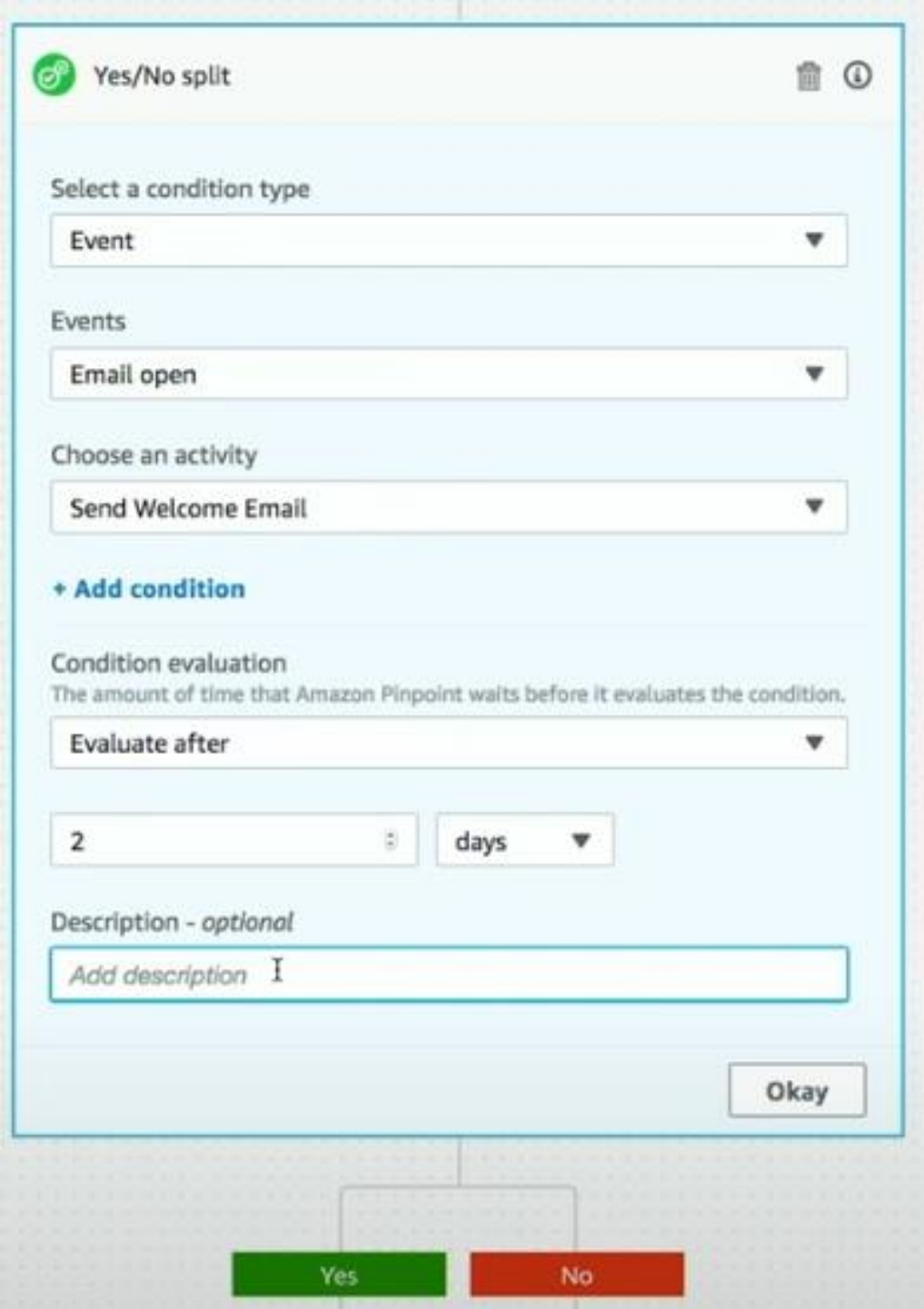
Check every hour if there are new users. Then send them a welcome email.



Amazon Pinpoint Example

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

Full demo: [Amazon Pinpoint Journeys]



Amazon Cognito Identity Pools

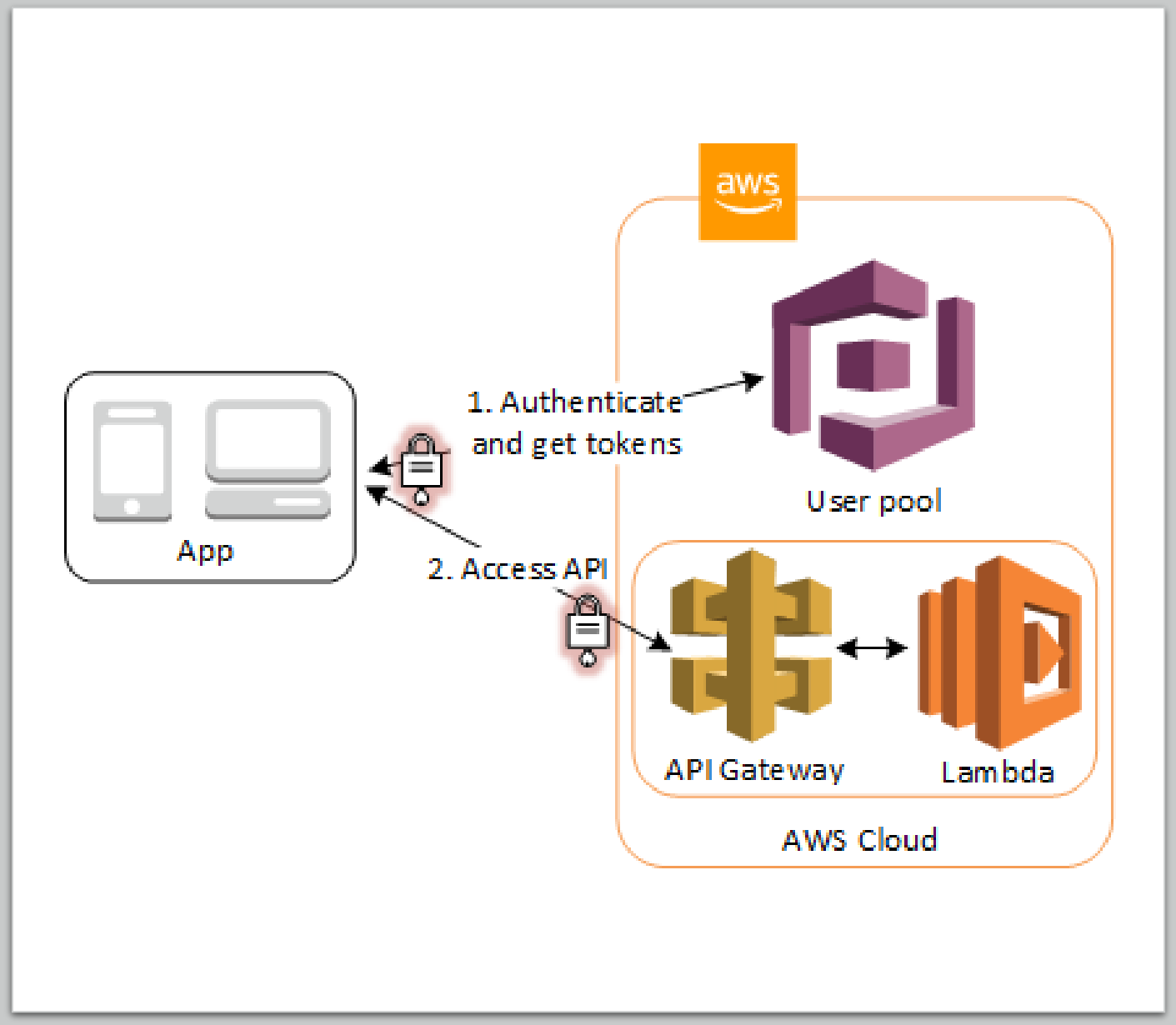
Amazon Cognito identity pools (federated identities) enable you to create unique identities for your users and federate them with identity providers. With an identity pool, you can obtain temporary, limited-privilege AWS credentials to access other AWS services.

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

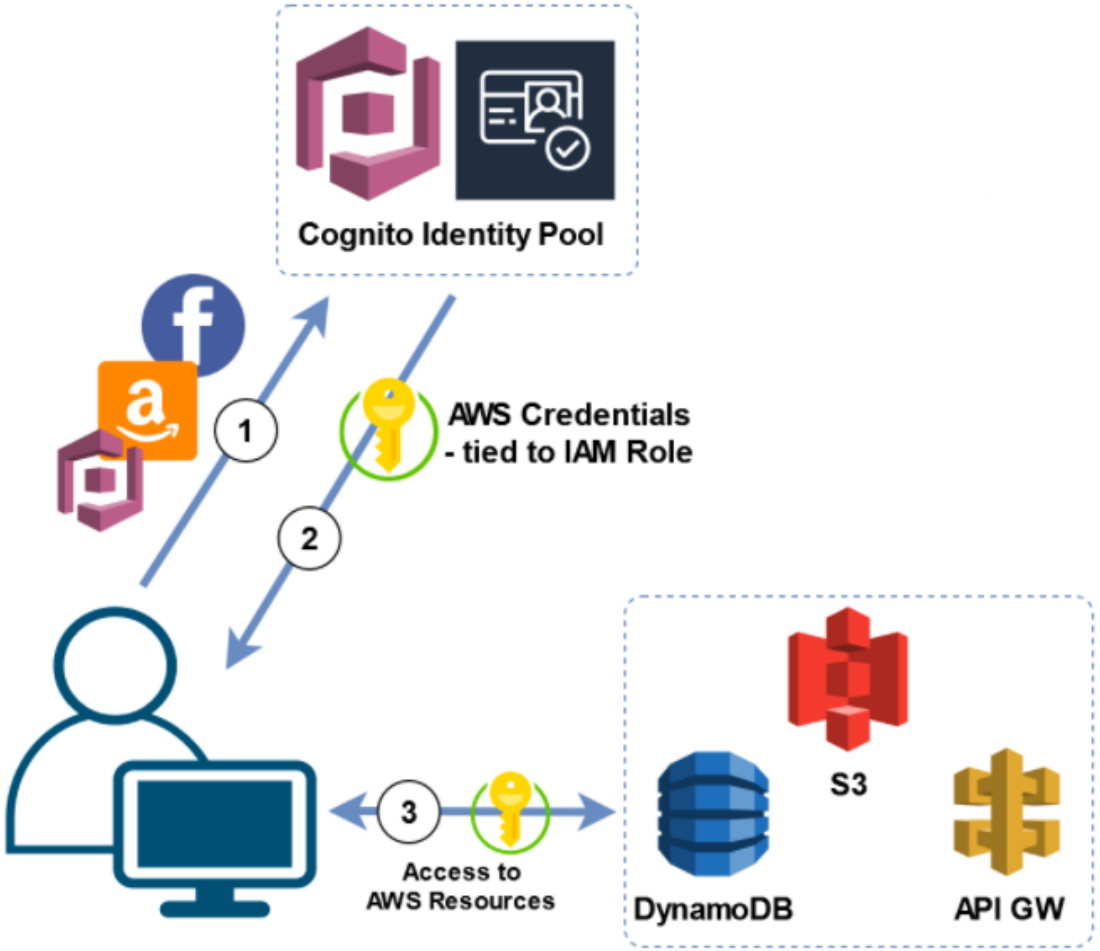
Without identity pool

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 knows 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