Amazon API Gateway

Amazon API Gateway

Architecture of API Gateway

Features of API Gateway

API Gateway use cases

API Gateway pricing

API Gateway – Deployment Stages

Amazon API Gateway

- Amazon API Gateway is a fully managed service that makes it easy for developers to publish, maintain, monitor, and secure APIs at any scale.
  - REST APIs
  - HTTP APIs
  - WebSocket APIs
- It handles all the tasks involved in accepting and processing up to hundreds of thousands of concurrent API calls, including traffic management, authorization and access control, monitoring, and API version management.
- With a few clicks in the AWS Management Console, you can create an API that acts as a “front door” for applications to access data, business logic, or functionality from your back-end services

Architecture of API Gateway

Features of API Gateway

- Support for stateful (WebSocket) and stateless (HTTP and REST) APIs.
- Powerful, flexible authentication mechanisms
- Developer portal for publishing your APIs.
- CloudTrail logging and monitoring of API usage and API changes.
- CloudWatch access logging and execution logging, including the ability to set alarms.
- Ability to use AWS CloudFormation templates to enable API creation.
- Support for custom domain names.
- Integration with AWS WAF for protecting your APIs against common web exploits.
- Integration with AWS X-Ray for understanding and triaging performance latencies.

API Gateway use cases

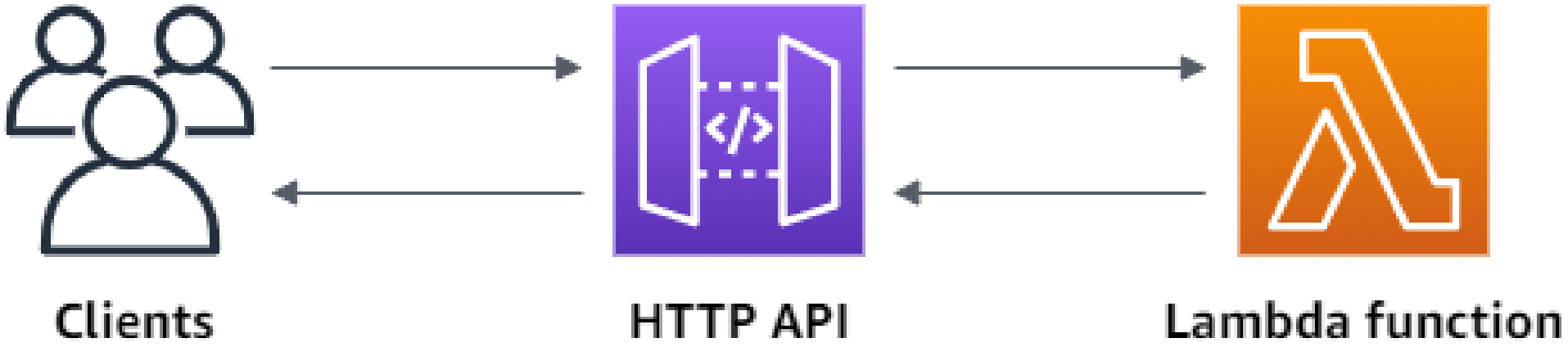
- Create HTTP APIs
  - HTTP APIs enable you to create RESTful APIs with lower latency and lower cost than REST APIs.
  - You can use HTTP APIs to send requests to AWS Lambda functions or to any publicly routable HTTP endpoint
- Create REST APIs
  - An API Gateway REST API is made up of resources and methods.
- Create WebSocket APIs
  - In a WebSocket API, the client and the server can both send messages to each other at any time.
  - Backend servers can easily push data to connected users and devices, avoiding the need to implement complex polling mechanisms.

API Gateway pricing

- API caching in Amazon API Gateway is not eligible for the AWS Free Tier.
- Requests are not charged for authorization and authentication failures.
- Calls to methods that require API keys are not charged when API keys are missing or invalid.
- API Gateway-throttled requests are not charged when the request rate or burst rate exceeds the preconfigured limits.
- Usage plan-throttled requests are not charged when rate limits or quota exceed the preconfigured limits.
- https://aws.amazon.com/api-gateway/pricing/

Getting started with API Gateway

- DEMO



API Gateway – Deployment Stages

- Making changes in the API Gateway will not effective
- You need to make a “deployment” to make the API work
- Changes are deployed to “Stages”
- Use the naming you like for stages (dev, test, prod)
- Each stage has its own configuration parameters

API Integration Type

- API integration type selected according to the types of integration endpoint you want to work
- Integration Type **AWS**
  - This type of integration lets an API expose AWS service actions
  - You must configure both the integration request and integration response
  - Setup data mapping using mapping templates for the request & response
- Integration Type **MOCK**
  - API Gateway returns a response without sending the request to the backend
- Integration Type **AWS_PROXY**
  - This integration relies on direct interactions between the client and the integrated Lambda function.
  - No mapping template
  - The HTTP response from the backend is forwarded by API Gateway

API Integration Type

- Integration Type **HTTP_PROXY**
  - You do not set the integration request or the integration response.
  - API Gateway passes the incoming request from the client to the HTTP endpoint and passes the outgoing response from the HTTP endpoint to the client
  - No mapping template
- Integration Type **HTTP**
  - You must configure both the integration request and integration response
  - Setup data mapping using mapping templates for the request & response

Mapping Templates

- AWS & HTTP Integration
- Mapping templates can be used to modify request / responses
- Rename / Modify query string parameters
- Modify body content
- Add headers
- Filter output results (remove unnecessary data)

AWS API Gateway Swagger / Open API spec

- Swagger can be written in YAML or JSON
- You can export current API as Swagger / OpenAPI spec
- Import existing Swagger / OpenAPI 3.0 spec to API Gateway

Enabling API Caching

- You can enable API caching in Amazon API Gateway to cache your endpoint's responses.
- Reduce the number of calls made to your endpoint and improve the latency of requests to your API.
- Default TTL (time to live) is 300 seconds (min: 0s, max: 3600s)
- Caches are defined per stage
- Cache capacity between 0.5GB to 237GB

API Gateway API Keys

- If you want to make an API available to your customers
- Uses API keys to identify API clients and meter access
- Alphanumeric string values to distribute to your customers
- Ex: ABCS23GF45hjvdydeg565DHGF6576tb
- Can use with usage plans to control access
- Throttling limits are applied to the API keys
- Quotas limits is the overall number of maximum requests

API Gateway - CORS

- Cross-origin resource sharing (CORS) is a browser security feature that restricts cross-origin HTTP requests
- CORS must be enabled when you receive API calls from another domain
- It can be enabled through the console
- The OPTIONS pre-flight request must contain the following headers:
  - Access-Control-Allow-Methods
  - Access-Control-Allow-Headers
  - Access-Control-Allow-Origin

API Gateway – Logging & Tracing

- CloudWatch Logs:
  - Enable CloudWatch logging at the Stage level (dev, stg, prod)
  - Create an IAM role for logging to CloudWatch
  - https://aws.amazon.com/premiumsupport/knowledge-center/api-gateway-cloudwatch-logs/
  - Can override settings on a per API basis (ex: ERROR, DEBUG, INFO)
  - Log contains information about request / response body
- X-Ray:
  - Enable tracing to get extra information about requests in API Gateway
  - X-Ray API Gateway + AWS Lambda gives you the full picture