Amazon API Gateway

Introduction

- Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services.
- Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications.
- You pay for the API calls you receive and the amount of data transferred out and, with the API Gateway tiered pricing model, you can reduce your cost as your API usage scales.

API Gateway - Features

- With API Gateway, you can route requests to private resources in your VPC. Using HTTP APIs, you can build APIs for services behind private ALBs, private NLBs, and IP-based services registered in AWS Cloud Map, such as ECS tasks.
- API Gateway helps you manage traffic to your backend systems by allowing you to set throttling rules based on the number of requests per second for each HTTP method in your APIs.
- API Gateway handles any level of traffic received by an API, so you are free to focus on your business logic and services rather than maintaining infrastructure. If you’re using REST APIs, you can also set up a cache with customizable keys and time-to-live in seconds for your API data to avoid hitting your backend services for each request.

API Gateway - Features

- With API Gateway, you can create RESTful APIs using either HTTP APIs or REST APIs. HTTP APIs are the best way to build APIs that do not require API management features. HTTP APIs are optimized for serverless workloads and HTTP backends—they offer up to 71% cost savings and 60% latency reduction compared to REST APIs from API Gateway.
- For workloads that require API proxy functionality and API management features in a single solution, such as usage plans and API keys, API Gateway offers REST APIs.

API Gateway - Authentication with Cognito

- As an alternative to using IAM roles and policies or Lambda authorizers (formerly known as custom authorizers), you can use an Amazon Cognito user pool to control who can access your API in Amazon API Gateway.
- To use an Amazon Cognito user pool with your API, you must first create an authorizer of the COGNITO_USER_POOLS type and then configure an API method to use that authorizer. After the API is deployed, the client must first sign the user in to the user pool, obtain an identity or access token for the user, and then call the API method with one of the tokens, which are typically set to the request's Authorization header.

API Gateway - VPC Endpoints

- API Gateway private endpoints are made possible via AWS PrivateLink interface VPC endpoints. Interface endpoints work by creating elastic network interfaces in subnets that you define inside your VPC.
- Those network interfaces then provide access to services running in other VPCs, or to AWS services such as API Gateway.
- When configuring your interface endpoints, you specify which service traffic should go through them. When using private DNS, all traffic to that service is directed to the interface endpoint instead of through a default route, such as through a NAT gateway or public IP address.

API Gateway - VPC Links

- The private integration uses an API Gateway resource of VpcLink to encapsulate connections between API Gateway and targeted VPC resources.
- As an owner of a VPC resource, you are responsible for creating a Network Load Balancer in your VPC and adding a VPC resource as a target of a Network Load Balancer's listener.
- As an API developer, to set up an API with the private integration, you are responsible for creating a VpcLink targeting the specified Network Load Balancer and then treating the VpcLink as an effective integration endpoint.

API Gateway - HTTP API

- 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 routable HTTP endpoint.
- For example, you can create an HTTP API that integrates with a Lambda function on the backend. When a client calls your API, API Gateway sends the request to the Lambda function and returns the function's response to the client.

API Gateway - Rest API

- A REST API in API Gateway is a collection of resources and methods that are integrated with backend HTTP endpoints, Lambda functions, or other AWS services. You can use API Gateway features to help you with all aspects of the API lifecycle, from creation through monitoring your production APIs.
- API Gateway REST APIs use a request/response model where a client sends a request to a service and the service responds back synchronously. This kind of model is suitable for many different kinds of applications that depend on synchronous communication.

API Gateway - 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 for a stage, API Gateway caches responses from your endpoint for a specified time-to-live (TTL) period, in seconds. API Gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint.
- The default TTL value for API caching is 300 seconds. The maximum TTL value is 3600 seconds. TTL=0 means caching is disabled.

API Gateway - Canary Deployment

- Canary release is a software development strategy in which a new version of an API (as well as other software) is deployed for testing purposes, and the base version remains deployed as a production release for normal operations on the same stage.
- In a canary release deployment, total API traffic is separated at random into a production release and a canary release with a pre-configured ratio. Typically, the canary release receives a small percentage of API traffic and the production release takes up the rest.

API Gateway - Lambda Integration

- You can integrate an API method with a Lambda function using Lambda proxy integration or Lambda non-proxy (custom) integration.
- In Lambda proxy integration, the setup is simple. If your API does not require content encoding or caching, you only need to set the integration's HTTP method to POST, the integration endpoint URI to the ARN of the Lambda function invocation action of a specific Lambda function, and the credential to an IAM role with permissions to allow API Gateway to call the Lambda function on your behalf.
- In Lambda non-proxy integration, in addition to the proxy integration setup steps, you also specify how the incoming request data is mapped to the integration request and how the resulting integration response data is mapped to the method response.

API Gateway - WAF and Logs

- You can use AWS WAF to protect your API Gateway API from common web exploits, such as SQL injection and cross-site scripting (XSS) attacks. These could affect API availability and performance, compromise security, or consume excessive resources.
- There are two types of API logging in CloudWatch: execution logging and access logging.
- In execution logging, API Gateway manages the CloudWatch Logs. The process includes creating log groups and log streams, and reporting to the log streams any caller's requests and responses.
- In access logging, you, as an API developer, want to log who has accessed your API and how the caller accessed the API. You can create your own log group or choose an existing log group that could be managed by API Gateway.

API Gateway - Pricing

- With Amazon API Gateway, you only pay when your APIs are in use. There are no minimum fees or upfront commitments. For HTTP APIs and REST APIs, you pay only for the API calls you receive and the amount of data transferred out.
- There are no data transfer out charges for Private APIs. However, AWS PrivateLink charges apply when using Private APIs in API Gateway.
- API Gateway also provides optional data caching charged at an hourly rate that varies based on the cache size you select. For WebSocket APIs, you only pay when your APIs are in use based on number of messages sent and received and connection minutes.
- The API Gateway free tier includes one million HTTP API calls, one million REST API calls, one million messages, and 750,000 connection minutes per month for up to 12 months.