General

Q: What is 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. 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, such as applications running on Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Container Service (Amazon ECS) or AWS Elastic Beanstalk, code running on AWS Lambda, or any web application. Amazon API Gateway handles all of 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. Amazon API Gateway has no minimum fees or startup costs. For HTTP APIs and REST APIs, you pay only for the API calls you receive and the amount of data transferred out. For WebSocket APIs, you pay only for messages sent and received and for the time a user/device is connected to the WebSocket API.

Show less

Q: Why use Amazon API Gateway?

Amazon API Gateway provides developers with a simple, flexible, fully managed, pay-as-you-go service that handles all aspects of creating and operating robust APIs for application back ends. With API Gateway, you can launch new services faster and with reduced investment so you can focus on building your core business services. API Gateway was built to help you with several aspects of creating and managing APIs:

1) **Metering**. API Gateway helps you define plans that meter and restrict third-party developer access to your APIs. You can define a set of plans, configure throttling, and quota limits on a per API key basis. API Gateway automatically meters traffic to your APIs and lets you extract utilization data for each API key.

2) **Security**. API Gateway provides you with multiple tools to authorize access to your APIs and control service operation access. API Gateway allows you to leverage AWS administration and security tools, such as AWS Identity and Access Management (IAM) and Amazon Cognito, to authorize access to your APIs. API Gateway can verify signed API calls on your behalf using the same methodology AWS uses for its own APIs. Using custom authorizers written as AWS Lambda functions, API Gateway can also help you verify incoming bearer tokens, removing authorization concerns from your backend code.

3) **Resiliency**. API Gateway helps you manage traffic with throttling so that backend operations can withstand traffic spikes. API Gateway also helps you improve the performance of your APIs and the latency your end users experience by caching the output of API calls to avoid calling your backend every time.

4) **Operations Monitoring**. After an API is published and in use, API Gateway provides you with a metrics dashboard to monitor calls to your services. The API Gateway dashboard, through integration with Amazon CloudWatch, provides you with backend performance metrics covering API calls, latency data and error rates. You can enable detailed metrics for each method in your APIs and also receive error, access or debug logs in CloudWatch Logs.

5) **Lifecycle Management**. After an API has been published, you often need to build and test new versions that enhance or add new functionality. API Gateway lets you operate multiple API versions and multiple stages for each version simultaneously so that existing applications can continue to call previous versions after new API versions are published.

6) **Designed for Developers**. API Gateway allows you to quickly create APIs and assign static content for their responses to reduce cross-team development effort and time-to-market for your applications. Teams who depend on your APIs can begin development while you build your backend processes.

7) **Real-Time Two-Way Communication**. Build real-time two-way communication applications such as chat apps, streaming dashboards, and notifications without having to run or manage any servers. API Gateway maintains a persistent connection between connected users and enables message transfer between them.

Show less

Q: What API types are supported by Amazon API Gateway?

Amazon API Gateway offers two options to create RESTful APIs, HTTP APIs and REST APIs, as well as an option to create WebSocket APIs.

HTTP API: HTTP APIs are optimized for building APIs that proxy to AWS Lambda functions or HTTP backends, making them ideal for serverless workloads. They do not currently offer API management functionality.

REST API: REST APIs offer API proxy functionality and API management features in a single solution. REST APIs offer API management features such as usage plans, API keys, publishing, and monetizing APIs.

WebSocket API: WebSocket APIs maintain a persistent connection between connected clients to enable real-time message communication. With WebSocket APIs in API Gateway, you can define backend integrations with AWS Lambda functions, Amazon Kinesis, or any HTTP endpoint to be invoked when messages are received from the connected clients.

Show less

Q: How do I get started with HTTP APIs in API Gateway?

To get started with HTTP APIs, you can use the Amazon API Gateway console, the AWS CLI, AWS SDKs, or AWS CloudFormation. To learn more about getting started with HTTP APIs, visit our documentation.

Show less

Q. How do I get started with REST APIs in API Gateway?

To get started with REST APIs, you can use the Amazon API Gateway console, the AWS CLI, or AWS SDKs. To learn more about getting started with REST APIs, visit our documentation.

Show less

Q. When creating RESTful APIs, when should I use HTTP APIs and when should I use REST APIs?

You can build RESTful APIs using both HTTP APIs and REST APIs in Amazon API Gateway.

HTTP APIs are optimized for building APIs that proxy to AWS Lambda functions or HTTP backends, making them ideal for serverless workloads. HTTP APIs are a cheaper and faster alternative to REST APIs, but they do not currently support API management functionality. REST APIs are intended for APIs that require API proxy functionality and API management features in a single solution.

HTTP APIs are ideal for:

1. Building proxy APIs for AWS Lambda or any HTTP endpoint
2. Building modern APIs that are equipped with OIDC and OAuth 2 authorization
3. Workloads that are likely to grow very large
4. APIs for latency sensitive workloads

REST APIs are ideal for:

1. Customers looking to pay a single price point for an all-inclusive set of features needed to build, manage, and publish their APIs.

Show less

Q. Which features come standard with HTTP APIs from API Gateway?

HTTP APIs come standard with CORS support, OIDC and OAuth2 support for authentication and authorization, and automatic deployments on stages.

Show less

Q. Can I import an OpenAPI definition to create a HTTP API?

Yes, you can import an API definition using OpenAPI 3. It will result in the creation of routes, integrations, and API models. For more information on importing OpenAPI definitions, see our documentation.

Show less

Q. How can I migrate from my current REST API to a HTTP API?

To migrate from your current REST API to a HTTP API in Amazon API Gateway, do the following:

1. Check that all the features you need are available in HTTP. To see the complete feature list, visit our documentation.
2. Go to your REST API and export the OpenAPI definition from your REST API
3. Go to your HTTP API and import the OpenAPI definition from the previous step
4. Test the API functions as expected
5. Update your clients with the new URL

While your API might work, you may notice some missing features. To identify any missing features, review the **Info**, **Warning**, and **Error** fields from the Import operation. For more information about migrating REST APIs to HTTP APIs, see our documentation.

Show less

Q. How do I know if my current REST API will work as a HTTP API?

First, go to your REST and export the OpenAPI definition from your REST API. Then, go to your HTTP API and import the OpenAPI definition from the previous step. While your API might work, you may notice some missing features. To identify any missing features, review the **Info**, **Warning**, and **Error** fields from the Import operation. The AWS CLI will return information about your API within your info and warning fields. For more, read our documentation.

Show less

Q: How do I get started with WebSocket APIs in Amazon API Gateway?

To get started, you can create a WebSocket API using the AWS Management Console, AWS CLI, or AWS SDKs. You can then set WebSocket routing to indicate the backend services such as AWS Lambda, Amazon Kinesis, or your HTTP endpoint to be invoked based on the message content. Refer to the documentation for getting started with WebSocket APIs in API Gateway.

Show less

Q: Can I create HTTPS endpoints?

Yes, all of the APIs created with Amazon API Gateway expose HTTPS endpoints only. Amazon API Gateway does not support unencrypted (HTTP) endpoints. By default, Amazon API Gateway assigns an internal domain to the API that automatically uses the Amazon API Gateway certificate. When configuring your APIs to run under a custom domain name, you can provide your own certificate for the domain.

Show less

Q: What data types can I use with Amazon API Gateway ?

APIs built on Amazon API Gateway can accept any payloads sent over HTTPS for HTTP APIs, REST APIs, and WebSocket APIs. Typical data formats include JSON, XML, query string parameters, and request headers. You can declare any content type for your APIs responses, and then use the transform templates to change the back-end response into your desired format.

Show less

Q: With what backends can Amazon API Gateway communicate?

Amazon API Gateway can execute AWS Lambda functions in your account, start AWS Step Functions state machines, or call HTTP endpoints hosted on AWS Elastic Beanstalk, Amazon EC2, and also non-AWS hosted HTTP based operations that are accessible via the public

Internet.API Gateway also allows you to specify a mapping template to generate static content to be returned, helping you mock your APIs before the backend is ready. You can also integrate API Gateway with other AWS services directly – for example, you could expose an API method in API Gateway that sends data directly to Amazon Kinesis.

Show less

Q: For which client platforms can Amazon API Gateway generate SDKs?

API Gateway generates custom SDKs for mobile app development with Android and iOS (Swift and Objective-C), and for web app development with JavaScript. API Gateway also supports generating SDKs for Ruby and Java. Once an API and its models are defined in API Gateway, you can use the AWS console or the API Gateway APIs to generate and download a client SDK. Client SDKs are only generated for REST APIs in Amazon API Gateway.

Show less

Q: In which AWS regions is Amazon API Gateway available?

To see where HTTP APIs, REST APIs, WebSocket APIs are available, view the AWS region table here.

Show less

Q: What can I manage through the Amazon API Gateway console?

Through the Amazon API Gateway console, you can define the REST API and its associated resources and methods, manage the API lifecycle, generate client SDKs and view API metrics. You can also use the API Gateway console to define your APIs' usage plans, manage developers' API keys, and configure throttling and quota limits. All of the same actions are available through the API Gateway APIs.

Show less

Q: What is a resource?

A resource is a typed object that is part of your API's domain. Each resource may have associated a data model, relationships to other resources, and can respond to different methods.You can also define resources as variables to intercept requests to multiple child resources.

Show less

Q: What is a method?

Each resource within a REST API can support one or more of the standard HTTP methods. You define which verbs should be supported for each resource (GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS) and their implementation. For example, a GET to the cars resource should return a list of cars. To connect all methods within a resource to a single backend endpoint, API Gateway also supports a special “ANY” method.

Show less

Q: What is a usage plan?

Usage plans help you declare plans for third-party developers that restrict access only to certain APIs, define throttling and request quota limits, and associate them with API keys. You can also extract utilization data on an per-API key basis to analyze API usage and generate billing documents. For example, you can create a basic, professional, and enterprise plans – you can configure the basic usage plan to only allow 1,000 requests per day and a maximum of 5 requests per second (RPS).

Show less

Q: What is the Amazon API Gateway API lifecycle?

With Amazon API Gateway, each REST API can have multiple stages. Stages are meant to help with the development lifecycle of an API -- for example, after you’ve built your APIs and you deploy them to a development stage, or when you are ready for production, you can deploy them to a production stage.

Show less

Q: What is a stage?

In Amazon API Gateway, stages are similar to tags. They define the path through which the deployment is accessible. For example, you can define a development stage and deploy your cars API to it. The resource will be accessible at https://www.myapi.com/dev/cars. You can also set up custom domain names to point directly to a stage, so that you don’t have to use the additional path parameter. For example, if you pointed myapi.com directly to the development stage, you could access your cars resource at https://www.myapi.com/cars. Stages can be configured using variables that can be accessed from your API configuration or mapping templates.

Show less

Q: What are stage variables?

Stage variables let you define key/value pairs of configuration values associated with a stage. These values, similarly to environment variables, can be used in your API configuration. For example, you could define the HTTP endpoint for your method integration as a stage variable, and use the variable in your API configuration instead of hardcoding the endpoint – this allows you to use a different endpoint for each stage (e.g. dev, beta, prod) with the same API configuration. Stage variables are also accessible in the mapping templates and can be used to pass configuration parameters to your Lambda or HTTP backend.

Show less

Q: What is a Resource Policy?

A Resource Policy is a JSON policy document that you attach to an API to control whether a specified principal (typically an IAM user or role) can invoke the API. You can use a Resource Policy to enable users from a different AWS account to securely access your API or to allow the API to be invoked only from specified source IP address ranges or CIDR blocks. Resource Policies can be used with REST APIs in Amazon API Gateway.

Show less

Q: What if I mistakenly deployed to a stage?

Amazon API Gateway saves the history of your deployments. At any point, using the Amazon API Gateway APIs or the console, you can roll back a stage to a previous deployment.

Show less

Q: Can I use my Swagger API definitions?

Yes. You can use our open source Swagger importer tool to import your Swagger API definitions into Amazon API Gateway. With the Swagger importer tool you can create and deploy new APIs as well as update existing ones.

Show less

Q: How do I monetize my APIs on API Gateway?

You can monetize your APIs on API Gateway by publishing them as products in AWS Marketplace. You will first need to register as a seller in AWS Marketplace, and submit your usage plans on API Gateway as products. Read here to learn more about API Monetization.

Show less

Q: How do I document my API on Amazon API Gateway?

API Gateway offers the ability to create, update, and delete documentation associated with each portion of your API, such as methods and resources. You can access documentation-related APIs through the AWS SDKs, CLI, via RESTful calls, or by editing the documentation strings directly in the API Gateway console. Documentation can also be imported as a Swagger file, either as part of the API or separately, allowing you to add or update the documentation without disturbing the API definition. API Gateway conforms to the Open API specification for documentation imported from, or exported to, Swagger files. Documentation is supported for REST APIs in API Gateway.

Show less

Q: How can I avoid creating redundant copies of error messages and other documentation that recurs frequently in my API?

In addition to offering standards-conformant API documentation support, API Gateway additionally supports documentation inheritance, making it simple to define a documentation string once and then use it in multiple places. Inheritance simplifies the process of defining API documentation, and can be converted to the standard representation when exporting the API as a Swagger file.

Show less

Q: Can I restrict access to private APIs to a specific Amazon VPC or VPC endpoint?

Yes, you can apply a Resource Policy to an API to restrict access to a specific Amazon VPC or VPC endpoint. You can also give an Amazon VPC or VPC endpoint from a different account access to the Private API using a Resource Policy.

Show less

back to top >>

Security and Authorization

Q: How do I authorize access to my APIs?

With Amazon API Gateway, you can optionally set your API methods to require authorization. When setting up a method to require authorization you can leverage AWS Signature Version 4 or Lambda authorizers to support your own bearer token auth strategy.

Show less

Q: How does AWS Signature Version 4 work?

You can use AWS credentials - access and secret keys - to sign requests to your service and authorize access like other AWS services. The signing of an Amazon API Gateway API request is managed by the custom API Gateway SDK generated for your service. You can retrieve temporary credentials associated with a role in your AWS account using Amazon Cognito.

Show less

Q: What is a Lambda authorizer?

Lambda authorizers are AWS Lambda functions. With custom request authorizers, you will be able to authorize access to APIs using a bearer token auth strategy such as OAuth. When an API is called, API Gateway checks if a Lambda authorizer is configured, API Gateway then calls the Lambda function with the incoming authorization token. You can use Lambda to implement various authorization strategies (e.g. JWT verification, OAuth provider callout) that return IAM policies which are used to authorize the request. If the policy returned by the authorizer is valid, API Gateway will cache the policy associated with the incoming token for up to 1 hour.

Show less

Q: Can Amazon API Gateway generate API keys for distribution to third-party developers?

Yes. API Gateway can generate API keys and associate them with an usage plan. Calls received from each API key are monitored and included in the Amazon CloudWatch Logs you can enable for each stage. However, we do not recommend you use API keys for authorization. You should use API keys to monitor usage by third-party developers and leverage a stronger mechanism for authorization, such as signed API calls or OAuth.

Show less

Q: How can I address or prevent API threats or abuse?

API Gateway supports throttling settings for each method or route in your APIs. You can set a standard rate limit and a burst rate limit per second for each method in your REST APIs and each route in WebSocket APIs. Further, API Gateway automatically protects your backend systems from distributed denial-of-service (DDoS) attacks, whether attacked with counterfeit requests (Layer 7) or SYN floods (Layer 3).

Show less

Q: Can I verify that it is API Gateway calling my backend?

Yes. Amazon API Gateway can generate a client-side SSL certificate and make the public key of that certificate available to you. Calls to your backend can be made with the generated certificate, and you can verify calls originating from Amazon API Gateway using the public key of the certificate.

Show less

Q: Can I use AWS CloudTrail with Amazon API Gateway?

Yes. Amazon API Gateway is integrated with AWS CloudTrail to give you a full auditable history of the changes to your REST APIs. All API calls made to the Amazon API Gateway APIs to create, modify, delete, or deploy REST APIs are logged to CloudTrail in your AWS account.

Show less

Q: How does Amazon API Gateway work with an Amazon Virtual Private Cloud (Amazon VPC)?

In Amazon API Gateway, you can proxy requests to backend HTTP/HTTPS resources running in your Amazon VPC by setting up Private Integrations using VPC Links. Client-side SSL certificates in Amazon API Gateway can be used to verify that requests to your backend systems were sent by API Gateway using the public key of the certificate. You can also create Private APIs in Amazon API Gateway which can only be accessible by resources within your Amazon VPC through Amazon VPC Endpoints.

Show less

Q: Can I restrict access to private APIs to a specific Amazon VPC or VPC endpoint?

Yes, you can apply a Resource Policy to an API to restrict access to a specific Amazon VPC or VPC endpoint. You can also give an Amazon VPC or VPC endpoint from a different account access to the Private API using a Resource Policy.

Show less

Q: Can I configure my REST APIs in API Gateway to use TLS 1.1 or higher ?

If you're using REST APIs, you can set up a CloudFront distribution with custom SSL certificate in your account and use it with Regional APIs in API Gateway. You can then configure the Security Policy for the CloudFront distribution with TLS 1.1 or higher based on your security and compliance requirements.

Show less

back to top >>

Management, Metrics, and Logging

Q: How can I monitor my Amazon API Gateway APIs?

Amazon API Gateway logs API calls, latency, and error rates to Amazon CloudWatch in your AWS account. The metrics are also available through the Amazon API Gateway console in a REST API dashboard. API Gateway also meters utilization by third-party developers, the data is available in the API Gateway console and through the APIs.

Show less

Q: Can I set up alarms on the Amazon API Gateway metrics?

Yes, Amazon API Gateway sends logging information and metrics to Amazon CloudWatch. You can utilize the Amazon CloudWatch console to set up custom alarms.

Show less

Q: How can I set up metrics for Amazon API Gateway?

By default, Amazon API Gateway monitors traffic at a REST API level. Optionally, you can enable detailed metrics for each method in your REST API from the deployment configuration APIs or console screen. Detailed metrics are also logged to Amazon CloudWatch and will be charged at the CloudWatch rates.

Show less

Q: Can I determine which version of the API my customers are using?

Yes. Metric details are specified by REST API and stage. Additionally, you can enable metrics for each method in your REST API.

Show less

Q: Does Amazon API Gateway provide logging support?

Yes. Amazon API Gateway integrates with Amazon CloudWatch Logs. You can optionally enable logging for each stage in your API. For each method in your REST APIs, you can set the verbosity of the logging, and if full request and response data should be logged.

Show less

Q: How quickly are logs available?

Logs, alarms, error rates and other metrics are stored in Amazon CloudWatch and are available near real time.

Show less

back to top >>

Throttling and Caching

Q: How can I protect my backend systems and applications from traffic spikes?

Amazon API Gateway provides throttling at multiple levels including global and by service call. Throttling limits can be set for standard rates and bursts. For example, API owners can set a rate limit of 1,000 requests per second for a specific method in their REST APIs, and also configure Amazon API Gateway to handle a burst of 2,000 requests per second for a few seconds. Amazon API Gateway tracks the number of requests per second. Any requests over the limit will receive a 429 HTTP response. The client SDKs (except Javascript) generated by Amazon API Gateway retry calls automatically when met with this response.

Show less

Q: Can I throttle individual developers calling my APIs?

Yes. With usage plans you can set throttling limits for individual API keys.

Show less

Q: How does throttling help me?

Throttling ensures that API traffic is controlled to help your backend services maintain performance and availability.

Show less

Q: At which levels can Amazon API Gateway throttle inbound API traffic?

Throttling rate limits can be set at the method level. You can edit the throttling limits in your method settings through the Amazon API Gateway APIs or in the Amazon API Gateway console.

Show less

Q: How are throttling rules applied?

API Gateway throttling related settings are applied in the following order: 1) per-client per-method throttling limits that you set for an API stage in a usage plan, 2) per-client throttling limits that you set in a usage plan, 3) default per-method limits and individual per-method limits that you set in API stage settings, 4) account-level throttling per region.

Show less

Q: Does Amazon API Gateway provide API result caching?

Yes. You can add caching to API calls by provisioning an API Gateway cache and specifying its size in gigabytes. The cache is provisioned for a specific stage of your APIs. This improves performance and reduces the traffic sent to your back end. Cache settings allow you to control the way the cache key is built and the time-to-live (TTL) of the data stored for each method. API Gateway also exposes management APIs that help you invalidate the cache for each stage. Caching is available for REST APIs in API Gateway.

Show less

Q: What happens if a large number of end users try to invoke my API simultaneously?

If caching is not enabled and throttling limits have not been applied, then all requests will pass through to your backend service until the account level throttling limits are reached. If throttling limits are in place, then Amazon API Gateway will shed the necessary amount of requests and send only the defined limit to your back-end service. If a cache is configured, then Amazon API Gateway will return a cached response for duplicate requests for a customizable time, but only if under configured throttling limits. This balance between the backend and client ensures optimal performance of the APIs for the applications that it supports. Requests that are throttled will be automatically retried by the client-side SDKs generated by Amazon API Gateway. By default, Amazon API Gateway does not set any cache on your API methods.

Show less

Q: How do APIs scale?

Amazon API Gateway acts as a proxy to the backend operations that you have configured. Amazon API Gateway will automatically scale to handle the amount of traffic your API receives. Amazon API Gateway does not arbitrarily limit or throttle invocations to your backend operations and all requests that are not intercepted by throttling and caching settings in the Amazon API Gateway console are sent to your backend operations.

Show less

back to top >>

Billing

Q: How am I charged for using Amazon API Gateway?

Amazon API Gateway bills per million API calls, plus the cost of data transfer out, in gigabytes. If you choose to provision a cache for your API, hourly rates apply. For WebSocket APIs, API Gateway bills based on messages sent and received and the number of minutes a client is connected to the API. Please see the API Gateway pricing page for details on API calls, data transfer, and caching costs per region.

Show less

Q: Who pays for Amazon API Gateway API calls generated by third-party developers?

The API owner is charged for the calls to their APIs on API Gateway.

Show less

Q: If an API response is served by cached data, is it still considered an API call for billing purposes?

Yes. API calls are counted equally for billing purposes whether the response is handled by your backend operations or the Amazon API Gateway caching operation.

Show less

back to top >>

WebSocket APIs

Q: What is WebSocket routing in Amazon API Gateway?

WebSocket routing in Amazon API Gateway is used to correctly route the messages to a specific integration. You specify a routing key and integration backend to invoke when defining your WebSocket API. The routing key is an attribute in the message body. A default integration can also be set for non-matching routing keys. Refer to documentation to learn more about routing.

Show less

Q: How can I send messages to connected clients from the backend service?

When a new client is connected to the WebSocket API, a unique URL, called the callback URL, is created for that client. You can use this callback URL to send messages to the client from the backend service.

Show less

Q: How can I authorize access to my WebSocket API in Amazon API Gateway?

With Amazon API Gateway, you can either use IAM roles and policies or AWS Lambda Authorizers to authorize access to your WebSocket APIs.

Show less

Q: How does my backend service know when a client is connected or disconnected from the WebSocket connection in Amazon API Gateway?

When a client is connected or disconnected, a message will be sent from the Amazon API Gateway service to your backend AWS Lambda function or your HTTP endpoint using the $connect and $disconnect routes. You can take appropriate actions like adding or removing the client for the list of connected users.

Show less

Q: How can my backend service identify if the client is still connected to the WebSocket connection??

You can use the callback URL GET method on the connection to identify if the client is connected to the WebSocket connection. Refer to documentation about using a callback URL.

Show less

Q: Can I disconnect a client from my backend service?

Yes, you can disconnect the connected client from your backend service using the callback URL.

Show less

Q: What is the maximum message size supported for WebSocket APIs?

The maximum supported message size is 128 KB. Refer to the documentation for other limits around WebSocket APIs.

Show less

Q: How am I charged for using WebSocket APIs on Amazon API Gateway?

You will be charged based on 2 metrics: Connection minutes and messages.

Connection minutes: Total number of minutes the clients or devices are connected to the WebSocket connection (rounded to a minute).

Messages: Total number of messages sent to and received from connected clients. Messages are charged in increments of 32KB. Refer to the pricing page for details about WebSocket API pricing and examples.

Show less

Q: If messages on the WebSocket connection fail authentication or authorization, do they still count toward my API usage bill?

No, if messages on the WebSocket connection fail authentication or authorization, they do not count toward your API usage bill.

Show less

back to top >>