



# Amazon API Gateway



## At the core of the lesson

---

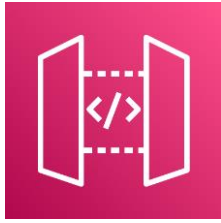
You will learn how to do the following:

- Define Amazon API Gateway and its benefits.
- Describe API Gateway architecture and how it is used.



# API Gateway overview

## What is API Gateway?



Amazon API  
Gateway

API Gateway is a service that developers can use to create and maintain APIs. For example, you can create a REST API for an application that you run on AWS.

API Gateway is a fully managed service that handles the following:

- Scaling
- Access control
- Monitoring



©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

4

Amazon API Gateway handles all the tasks that are involved in accepting and processing concurrent API calls at scale. These tasks include traffic management, authorization and access control, monitoring, and API version management. You pay for only the API calls that you receive and the amount of data that is transferred out.

API Gateway provides you with a dashboard to visually monitor calls to your services so that you can see performance metrics.

API Gateway works with AWS Lambda so that you can create serverless APIs:

- First, you create REST APIs with API Gateway.
- Then, your mobile and web applications can use these APIs to call publicly available AWS services through the code that you run in Lambda.

With API Gateway, you can create RESTful resource-based APIs. Then, you can use the data transformation capabilities to generate the requests in the language that your target services expect.

## API Gateway benefits

---

Benefits of API Gateway include the following:

- Efficient API development
- Performance at any scale
- Cost savings at scale
- Monitoring
- Flexible security controls
- RESTful API options



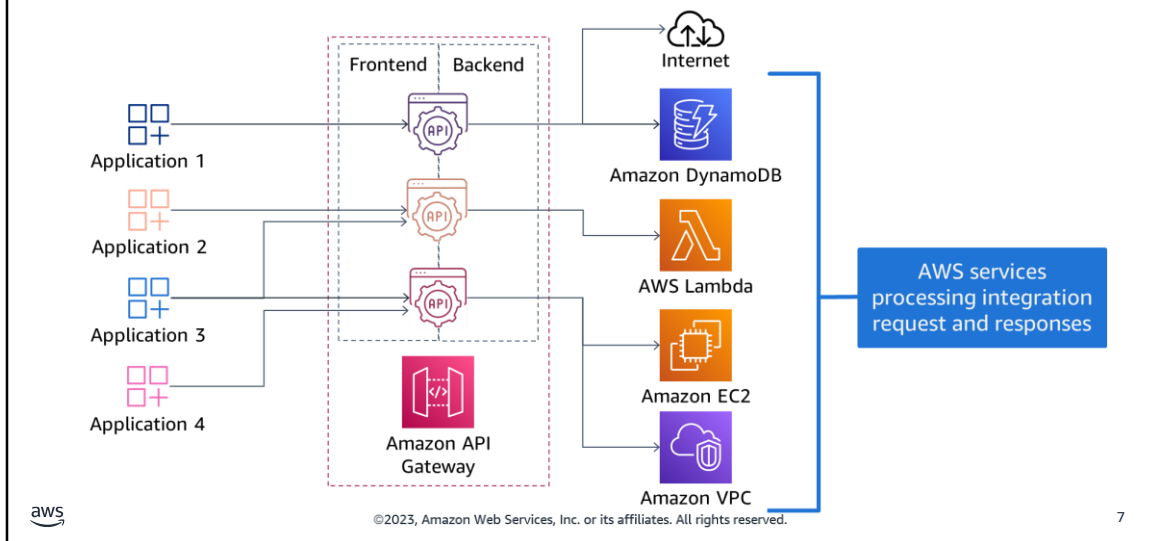
Using API Gateway offers many benefits, including the following:

- **Efficient API development:** Run multiple versions of the same API simultaneously with API Gateway, which gives you the ability to quickly iterate, test, and release new versions.
- **Performance at any scale:** Provide end users with the lowest possible latency for API requests and responses by taking advantage of the AWS global network of edge locations using Amazon CloudFront.
- **Cost savings at scale:** Decrease your costs as your API usage increases per Region across your AWS accounts using the tiered pricing model for API requests.
- **Monitoring:** Monitor performance metrics and information on API calls, data latency, and error rates from the API Gateway dashboard.
- **Flexible security controls:** Authorize access to your APIs with AWS Identity and Access Management (IAM) and Amazon Cognito.
- **RESTful API options:** HTTP APIs are the best way to build APIs for a majority of use cases because they can be significantly cheaper than REST APIs.



# API Gateway architecture

## Architecture example



This diagram shows how client applications gain programmatic access to AWS services (or a website on the internet) through one or more APIs. These APIs are made accessible through API Gateway.

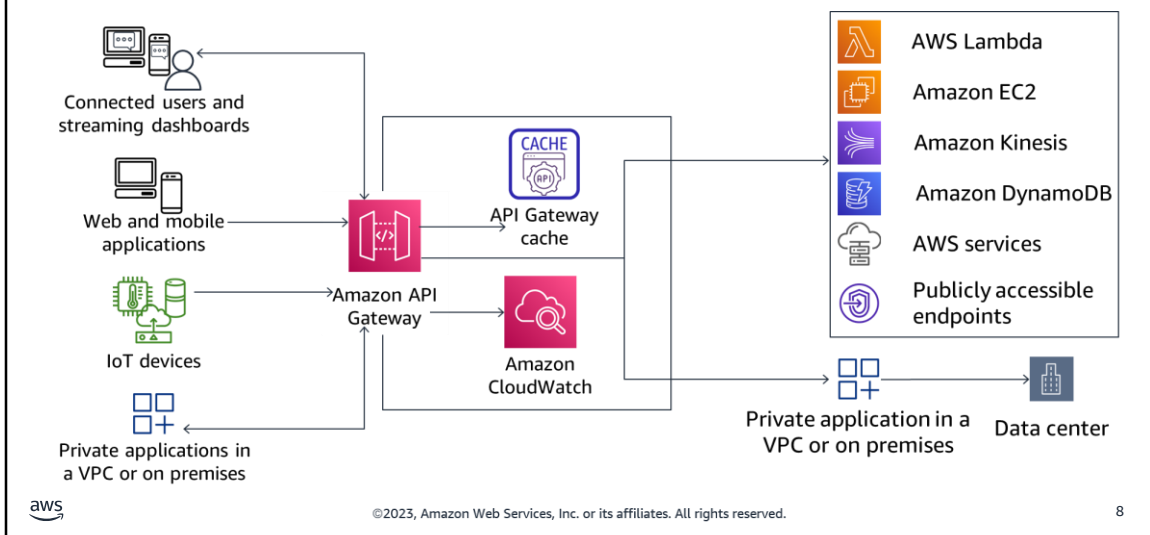
The API interfaces that you develop have a frontend and a backend. The client uses the frontend applications to make requests. The parts of the API implementation that communicate with the other AWS services are referred to as the backend.

In API Gateway, the frontend is encapsulated by method requests and method responses. The backend is encapsulated by requests and responses that work with the other AWS services. These AWS services provide the functionality that the API exposes, and they take action accordingly.

The diagram includes the following services:

- Amazon API Gateway
- Amazon DynamoDB
- AWS Lambda
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Virtual Private Cloud (Amazon VPC)

## How API Gateway is used



The diagram illustrates how APIs can be built for various applications. For example, these applications include web and mobile applications, Internet of Things (IoT) devices, and other applications that use API Gateway. In API Gateway, you can create, publish, maintain, and monitor APIs. These APIs can integrate with other AWS serverless applications.



## API Gateway example

### Serverless web application



This example shows the use of API Gateway in the implementation of a serverless web application. The application uses Amazon Simple Storage Service (Amazon S3) to hosts its presentation code and Amazon Cognito for authentication and authorization. The application also stores its data in a DynamoDB database.

The application's user interface invokes the RESTful API exposed by API Gateway. This API forwards the user's request to a Lambda function, which performs the application's functions, accesses the database, and returns a response.

Take a moment to review the components of the web application in the diagram:

1. Amazon S3 hosts static web resources—including HTML, CSS, JavaScript, and image files—that are loaded in the user's browser.
2. Amazon Cognito provides user management and authentication functions to secure the backend API.
3. The browser runs JavaScript that sends and receives data by communicating with API Gateway through REST web services. The data that is sent through API Gateway uses the backend API that was built with Lambda.
4. DynamoDB provides the persistence layer in this example. The Lambda function that the API uses can store data in the DynamoDB database.

## Checkpoint questions

---

1. What is API Gateway used for?
2. Which types of applications use APIs and API Gateway?
3. What is the difference between an API frontend and an API backend?



The answers to the questions are as follows:

1. What is API Gateway used for?  
API Gateway is used for creating, publishing, maintaining, and monitoring API calls.
2. Which types of applications use APIs and API Gateway?  
Web and mobile applications, Internet of Things (IoT), and many others
3. What is the difference between an API frontend and an API backend?  
The applications use the frontend to make requests. The parts of the API implementation that communicate with AWS services is referred to as the backend.

## Key ideas

---



- API Gateway works with Amazon CloudFront. API Gateway give you the ability to use edge locations, which provide users with the lowest latency for API requests and responses.
- After your API is deployed, API Gateway provides you with a dashboard to visually monitor calls to a service.
- The client applications use the frontend to make requests.
- The parts of the API implementation that communicate with the other AWS services are referred to as the backend.



# Thank you

Corrections, feedback, or other questions?  
Contact us at <https://support.aws.amazon.com/#/contacts/aws-training>.  
All trademarks are the property of their owners.

©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

12