API Gateways Comparison

express gateway

vs

Amazon API Gateway

Comparison Guide by the API experts over at:

LunchBadger

API Gateways:
Express Gateway Vs. AWS

ABSTRACT

We've created this comparison page to make it easy to understand the major differences (and similarities) between two popular projects for the API Gateway use case. In this review we'll be comparing Express Gateway and Amazon API Gateway across multiple dimensions, "at-a-glance."

What is a microservices API Gateway?

A microservices API Gateway sits in front of microservices and performs the external functions of an API Gateway to clients including other microservices. Gateways in general make it much simpler to develop, secure, manage, and scale endpoints by moving most of the infrastructure level logic from both the backend microservices and client, into the gateway.

Further reading: microservices.io

Express Gateway

Express Gateway is an API Gateway that can sit at the heart of any microservices architecture, regardless of what language or platform you're using. Express Gateway secures your microservices and exposes them through APIs using Node.js, ExpressJS and Express middleware. Developing microservices, orchestrating and managing them can now be done insanely fast, on one platform, seamlessly, and without having to introduce additional infrastructure. Many policies for enterprise authentication schemes and other enterprise features for other open source based gateways are available only in their paid versionsThe maintainers of Express Gateway intend to make enterprise policies and many enterprise features for Express Gateway - free.An API Gateway is a critical infrastructure component in the enterprise that makes available backend services to mobile, web and other external clients via a set of protocols and commonly through a set of RESTful application programming interfaces (APIs). An API Gateway makes it much simpler to develop, secure, manage, and scale endpoints by moving most of the required logic from the client, into the gateway.

The project can do so because it leverages the 3,000+ ExpressJS middleware components to quickly build functionality that is rapidly becoming commoditized anyway.Express Gateway's single core strength is its existing ecosystem to leverage and limitless customization and extensibility not just through plugins but because its core is completely written in JavaScript, a language that is ubiquitous among developers.

Learn more about Express Gateway.

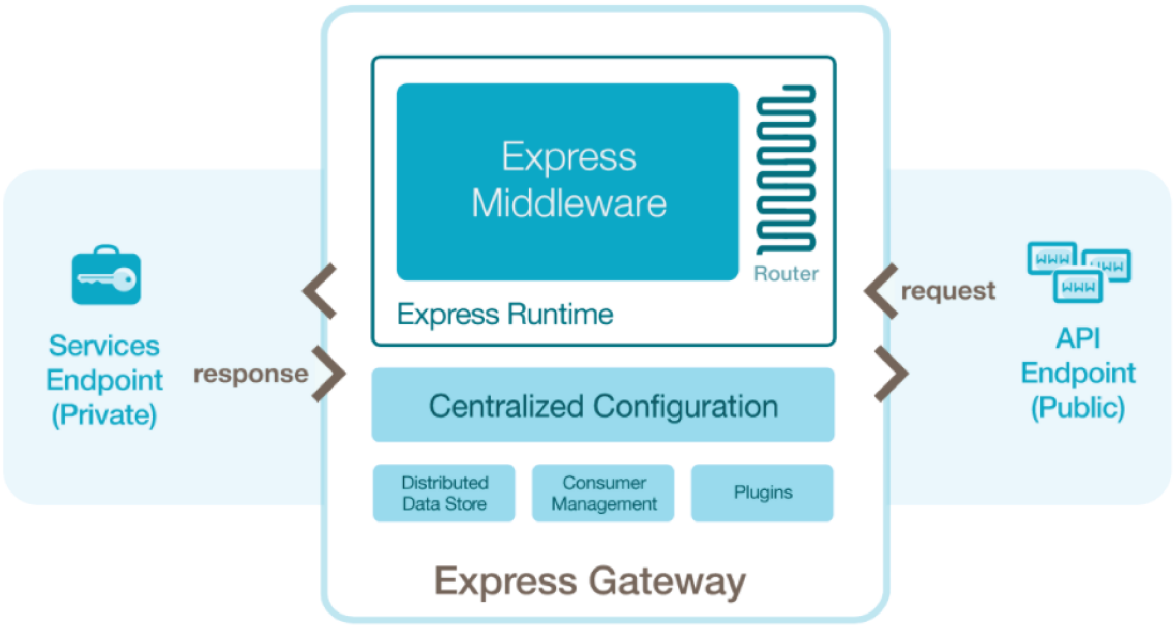
AWS API Gateway

Amazon API Gateway

Amazon's API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. With a few clicks 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 workloads running on Amazon Elastic Compute Cloud (Amazon EC2), code running on AWS Lambda, or any Web application. Amazon API Gateway 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.

Learn more about Amazon API Gateway.

Express Gateway Features & Architecture



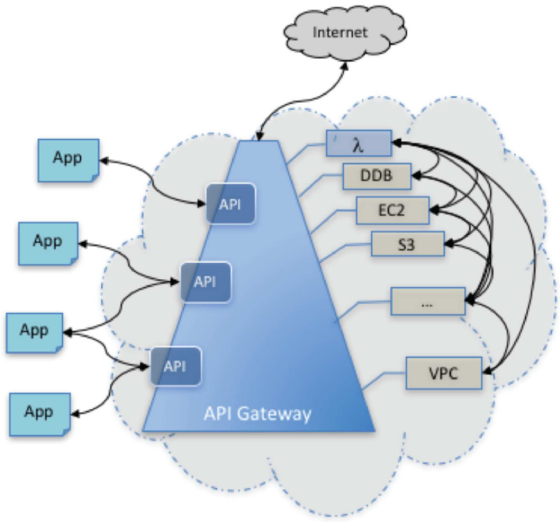
Express Gateway is an open source API Gateway written in Node.js and built on top of Express.js and Express Middleware. Express.js is the most popular framework for Node.js applications. Express Gateway is Apache 2.0 licensed with commercial support available from LunchBadger.

Features include:

- Simple configuration via a YAML file
- Plugin architecture
- Extensible with over 3,000 modules and growing
- Runs anywhere (Docker, private or public cloud)
- Auto-detects and hot-reload of configuration changes
- Define multiple policy sets (pipeline) per API endpoint
- Supports any language
- Supports any framework
- Supports many microservice use cases, patterns and designs
- Works with any orchestrator
- Works with any service mesh
- Plugs directly into existing DevOps tooling and pipelines

Further reading: Express Gateway GitHub repository

AWS Features & Architecture



Amazon API Gateway is a closed-source software-as-a-service (SaaS) product written in Node.js available only on AWS. Amazon API Gateway can be considered a backplane in the cloud to connect AWS services and other public or private websites. It provides consistent RESTful application programming interfaces (APIs) for mobile and web applications to access AWS services. Together with AWS Lambda, API Gateway forms the app-facing part of the AWS serverless infrastructure. For an app to call publicly available AWS services, you can use Lambda to interact with the required services and expose the Lambda functions through API methods in API Gateway. AWS Lambda runs the code on a highly available computing infrastructure. It performs the necessary execution and administration of the computing resources. To enable the serverless applications, API Gateway supports streamlined proxy integrations with AWS Lambda and HTTP endpoints.

Features include:

- Build, deploy and manage APIs
- Resiliency
- API lifecycle management
- SDK generation
- API operations monitoring
- AWS authorization
- API keys for third-party developers

It should be noted that with Amazon API Gateway there is no way to separate out hosting costs from the cost of the software. Amazon charges you per million API calls, per GB of data transfer, and optionally for caching. You can find pricing information here. Amazon API Gateway does have a free tier, so you will pay nothing for your first 12 months as long as you are under 1 million API calls per month.

Express Gateway Getting Started

Getting started with Express Gateway is very easy because all you need is Node.js with an optional backend data store. Getting up and running with Express Gateway is a simple, four step process executed at the command-line (detailed here.)

Installation:

- Installing Express Gateway via npm
- Creating an Express Gateway
- Following the command-line prompts to configure the gateway from the available templates
- Running Express Gateway

If you choose to get started using Docker, all you need is a Node.js Docker image. You can even compile your gateway into a standalone executable using pkg. Finally, if you need a data store because you intend to provide your Express Gateway users with API keys, you have numerous options concerning Redis. These include the Redis Docker image, Redis Cloud, or Compose's hosted Redis offering.

AWS Getting Started

Deploying Amazon API Gateway is done via GUI or AWS' CLI. You need to create a new deployment and a new stage. You can think of a stage as a snapshot of the API configuration, analogous to a tag in git. It should be noted that Amazon API Gateway doesn't integrate with a database directly. In order to store data like users and API keys, you need to either use AWS' IAM users and permissions, or use custom authorizers and AWS Lambda. If you choose to use custom authorizers, you will need to set up your own database independent of Amazon API Gateway.

You can find the complete details concerning how to get started with Amazon API Gateway in their documentation.

Administration and Maintenance

Express Gateway

Amazon API Gateway and Express Gateway have very different models for managing their gateway configuration. Express Gateway's high-level configuration is defined in a YAML file that you can track in GitHub. There is also a CLI and admin API for managing users and credentials. At the moment there is no officially supported GUI for the Admin API. However, unless you need to create users and credentials, you can configure Express Gateway entirely using a YAML file. The YAML file is one such feature that makes Express Gateway cloud native and ready to be run containers and orchestrators like Kubernetes to take advantage of cloud native features like ConfigMaps.

LunchBadger offers Express Gateway Enterprise which adds a GUI called the Canvas that allows a developer or devops engineer to configure and manage multiple Express Gateway instances and automatically deploy instances in Kubernetes for scale and cloud native management.

Scaling and load-balancing Express Gateway is done via Kubernetes. Check out this demo, with source code, to understand how to set-up Express Gateway using Kubernetes.

AWS

Amazon API Gateway is configured and managed through either a CLI or the AWS console GUI. Stages serve a similar purpose to releases and allow you to have a separate development API to test against. You can also manage Amazon API Gateway separately by maintaining a Swagger configuration in JSON and importing the configuration into Amazon API Gateway every time you make a change. This would enable you to track your API gateway configuration in a version control system like git.

A limitation of Amazon API Gateway is that your gateway has a separate configuration for each resource. There is no way to configure your gateway to handle requests differently based on the request hostname or the IP address of the incoming request and there's no way to share logic between resources, unless you write your own code in AWS Lambda. With Express Gateway, you can turn on features like rate-limiting by hostname or by IP address with a single line in your config file, and policies enable you to reuse logic in different pipelines.

Built-in Plugin Support

Express Gateway supports plugins to extend the core platform and enable custom protocol handlers, request / response transformations, authentication modules, etc. While Amazon API Gateway relies heavily on mostly raw AWS Lambda functions combined with libraries.

As an example - it is worth noting that a key difference between Amazon API Gateway and Express Gateway is that Amazon API Gateway requires you to build your own OAuth 2 support via custom authorizers and general request transformation via AWS Lambda. In order to use OAuth 2 with Amazon API Gateway, you need to use Lambda or another API provider to host your OAuth 2 logic.

Features Comparison

Features	Express Gateway	Amazon API Gateway
Public or Private Cloud	✓	** (1)
On Premises	✓	✗
Configuration & Administration	YAML, CLI, API	CLI, GUI, API
Database (Persistent Storage)	Redis	**(2)
HTTPS	✓	✓
CORS	✓	✓
Basic Authentication	✓	**(3) ✗
Key Authentication	✓	**(3) ✗
OAuth 2	✓	**(3) ✗
JWT	✓	**(3) ✗
Finegrain Access Control	✓	**(3) ✗
Rate Limiting	✓	✓
Request Transformation	✓	✓
Response Transformation	✓	✓
Pipeline Driven Conditional Actions	✓	**(3) ✗
Pipeline Driven Expressions	✓	**(3) ✗
Consumer Management	✓	**(3) ✗

1. Amazon Only
2. If you choose to use custom authorizers, you will need to set up your own database independent of Amazon API Gateway.
3. No built-in solution. Requires implementation through AWS Lambda, AWS Incognito or AWS IAM which are additional products

Custom Plugin Support

When built-in plugins are not enough, both Amazon API Gateway and Express Gateway have a mechanism for extending your gateway with custom functionality. Amazon API Gateway's mechanism relies primarily on AWS Lambda. In addition to being able to use API Gateway to proxy requests to AWS Lambda, you can also use custom authorizers to call AWS Lambda functions.

Express Gateway Custom Plugin Support

Express Gateway plugins are written in JavaScript using the Express.js framework. Express Gateway plugins are analogous to Express middleware, which makes it easy to reuse the prolific JavaScript ecosystem in your API gateway.

Amazon API Gateway Custom Plugin Support

By combining Amazon API Gateway with AWS Lambda, you can build custom logic in JavaScript, Python, Java, or C#. AWS Lambda uses Node.js as its JavaScript runtime, so, like Express Gateway, you can easily leverage npm packages in your API gateway. However, this requires a substantial investment in coding and configuration to get comparable functionality to just a few commands and configuration parameters when extending Express Gateway.

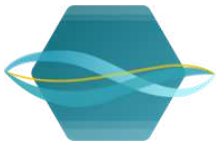
Enterprise Versions

Express Gateway Enterprise and Express Serverless Platform

LunchBadger has built a commercial product around Express Gateway with two editions:

Express Gateway Enterprise adds a visual interface called the Canvas to manage, configure and orchestrate multiple instances of Express Gateway. Express Gateway Enterprise also provides a Kubernetes based runtime and a set of microservices that automate all actions performed within the Canvas into Kubernetes in real time in development and immutable deployment for testing and production.

Express Gateway Enterprise utilizes Kubernetes for multicloud management and scale. Utilizing Kubernetes natively also makes it easier for enterprises to realize their cloud strategy and its benefits across application and infrastructure under the same leading container orchestrator.



Express Serverless Platform further extends Express Gateway Enterprise into a full blown microservices platform. It adds microservices integration and composition capabilities under the same unified experience provided by the Canvas, Kubernetes, Kubernetes automation – as ONE seamless platform to develop microservices as functions and expose them as APIs.

Amazon API Gateway

Amazon API Gateway does not have an enterprise version. Amazon API Gateway is a very low level gateway with minimal policies that you would expect and see in other API management solutions. Enterprise features such as Authentication, Authorization, Audit (AAA) are not present in the gateway. Some enterprise features like OAuth2 are provided only through another AWS offering Cognito. Amazon API Gateway can be extended with enterprise features using Lambda but you're on your own to doing the work or finding functions that can be reused for your instance and use cases.

Quick Reference Links

Express Gateway

- Github Repository
- Documentation
- Installation and Getting Started
- Plugins
- Commercial Support

AWS API Gateway

- Installation & Getting Started
- Documentation
- Commercial Support

Summary

Amazon API Gateway and Express Gateway both have their strengths and weaknesses. Amazon API Gateway has a compelling case for massively scalable API gateways at a competitive price point. With Express Gateway, you are responsible for hosting and running your own API Gateway, which may be more or less expensive depending on your experience and the load on your API Gateway. In particular, you may need to manage the number of Express Gateway servers you're running and configure a load balancer in front of your gateway servers if your API has sufficient load, which may be cumbersome and expensive.

On the other hand, Amazon API Gateway's documentation is quite cumbersome. A great deal of their documentation goes into excruciating detail on tangential topics without actually telling you how to use the functionality, like their guide to rate limiting. Setting up rudimentary API gateway tasks, like OAuth2 or key authorization, require plugging in additional AWS services, like Lambda and IAM users, so it is unlikely you'll be able to use Amazon API Gateway independent of the rest of AWS.

Express Gateway has more concise documentation, greater ease-of-use, built-in functionality that just works “out-of-the-box”, and an active blog with detailed overviews of use cases.

If all else fails, you can just read the source code because Express Gateway is open source and available on GitHub.