Developer's Comparison Guide

Express Serverless Platform vs. AWS

presented by LunchBadger

Express Serverless Platform vs. AWS

ABSTRACT

We've created this comparison page to make it easy to understand the major differences (and similarities) between two popular platforms for the serverless based microservice development use case. In this review we'll be comparing Express Serverless Platform and AWS Lambda across multiple dimensions and "at-a-glance.

What is Express Serverless Platform?

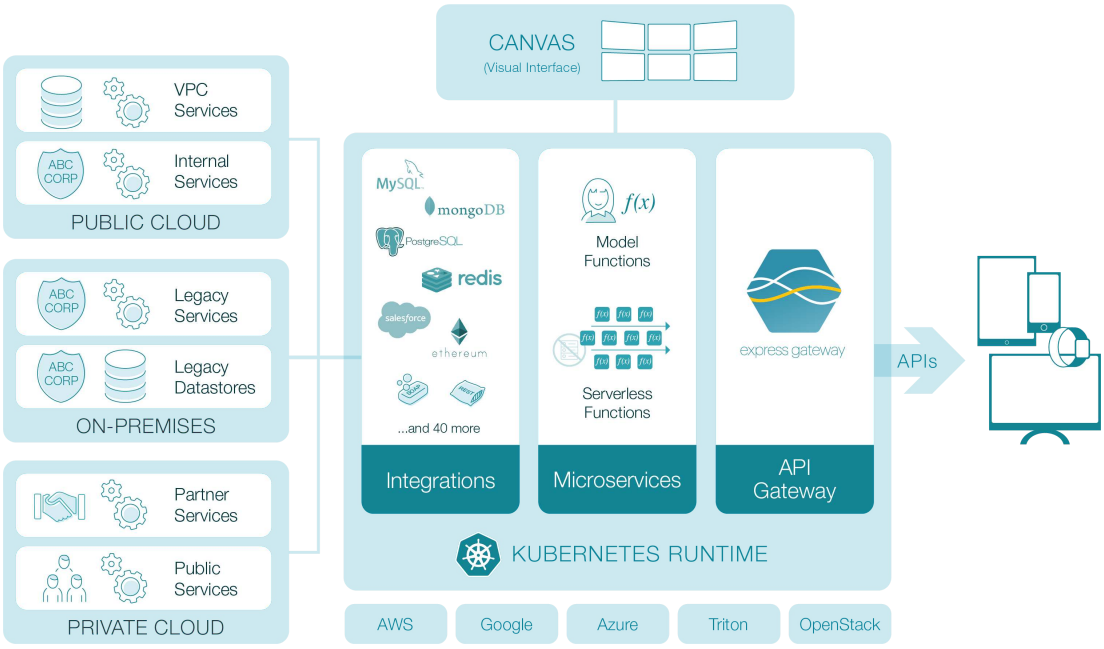
Express Serverless Platform provides a uniform and seamless development experience of microservices and APIs as functions.
Express Serverless Platform can be deployed to ANY public or private cloud.

Enterprises can have multiple deployments of Express Serverless Platform at the same time to achieve a multicloud and or hybrid cloud strategy. Each deployed instance allows you to take advantage of a cloud's native infrastructure without coupling your code to its proprietary interfaces.

Express Serverless Platform's GUI is called the Canvas. Actions done inside the Canvas are orchestrated and automated within Kubernetes as pods running container based microservices in real time for development and modeling purposes. Other actions in the Canvas can also automate and deploy native cloud infrastructure such as serverless offerings (e.g. AWS Lambda) running alongside the cloud's Kubernetes offering.
Express Serverless Platform automates and manages both container and serverless microservices seamlessly giving you unified view of what your microservices application looks like and how they're orchestrated across different pieces of infrastructure.

Features & Architecture

Express Serverless Platform Features & Architecture



Express Serverless Platform is fully self contained. Express Serverless Platform can be installed on bare metal or VMs with all the necessary infrastructure components included: an API Gateway, Kubernetes Runtime, Serverless Engine and visual Canvas that serves as a "single pane of glass" to visualize your microservices and API development.

Express Serverless Platform is fully modularized and integration ready. Any deployment of Express Serverless Platform can take full advantage of a particular public cloud's native proprietary offerings such as its own serverless infrastructure.

The Express Serverless Platform utilizes the best of breed open source technologies in its reference implementation that is ready to be used out of the box as is:

- Express Gateway, an open source API Gateway written in Node.js and built on top of Express.js
- Loopback.js, and enterprise Node.js framework for building microservices as model based functions and vast microservice integration library of connectors
- Kubeless, a Kubernetes native serverless engine to run polyglot functions
- Serverless (the framework), a framework that provides abstraction of functions to multiple serverless implementations - AWS Lambda, Azure Functions, Google Cloud Functions, Openwhisk, Kubeless and others
- Kubernetes, the leading container orchestrator to run microservices running in containers, supported by all public and private clouds

Express Serverless Platform Features & Architecture Cont.

Features Include

- Composition of Serverless FunctionsComposition of Model based Functions
- Supports Node.js, Python, Ruby, Go, PHP, .Net Core, and Java
- Completely extensible through open source modules
- Suite of Enterprise connectors to tap into legacy systems for Model based Functions
- Connects multi-cloud environments (private or public cloud)
- Supports any microservice use cases, patterns and designs
- Auto deployment to a Kubernetes Runtime
- Works with any Kubernetes cluster (EKS, GKE, etc)Auto exposes functions as APIs through the gateway
- Plugs directly into existing DevOps tooling and pipelines
- Can take advantage of public cloud's proprietary infrastructure services
- Reference implementation comes complete with all infrastructure components like APIgateway, serverless engine, container orchestrator, and GUI

Further reading: LunchBadger Documentation:

AWS Lambda Features & Architecture

AWS Lambda is Amazon Web Service's serverless offering. AWS Lambda started the whole serverless movement with a focus around functions. Lambda lets you write custom functions and expose them via an HTTP interface and through other events. AWS automatically scales your backend functions and takes care of other details like high availability.

Features Include

- Deploys, runs and scales your code with no server management
- Spin up AWS Lambda on-demand and back down in response to events in the environment
- Supports multiple languages including Node.js, Python, Java, and .Net Core
- Call your code directly from any web, mobile, or backend application via HTTP through AWS API Gateway1
- Built in automatic scalingLog and monitor performanceDebugging and Error ReportingPay per use
- 1 Little known fact that the real cost of running Lambda is actually the API Gateway - please see the Medium article for details.Deploys, runs and scales your code with no server management
- Spin up AWS Lambda on-demand and back down in response to events in the environment
- Supports multiple languages including Node.js, Python, Java, and .Net CoreCall your code directly from any web, mobile, or backend application via HTTP through AWS API Gateway1
- Built in automatic scalingLog and monitor performanceDebugging and Error ReportingPay per use

1 Little known fact that the real cost of running Lambda is actually the API Gateway - please see the Medium article for details.

Further reading: AWS Lambda Features

Getting Started with the Express Serverless Platform

Getting started with Express Serverless Platform is easy because the trial environment provides a complete walkthrough to take you through the following steps: how to create a microservice, deploy and configure an API Gateway, connect a microservice to the API Gateway, and expose it as an API Endpoints. Then, you'll trace your API workflow from public API Endpoints to your in-memory data source.Getting started with Express Serverless Platform is easy because the trial environment provides a complete walkthrough to take you through the following steps: how to create a microservice, deploy and configure an API Gateway, connect a microservice to the API Gateway, and expose it as an API Endpoints. Then, you'll trace your API workflow from public API Endpoints to your in-memory data source.

QuickStart

- Deploy and use an Memory Connector to connect to an in-memory database
- Create and deploy a "Car" that will be a Model based microservice
- Connect the Car Model to the Memory Connector to read and write Car data
- Deploy and configure an API Gateway – an instance of Express Gateway
- Connect the Car Model to the API Gateway
- Expose the Car Model microservice as a Car API Endpoint that we can call through an API Request using cURL
- Deploy a Function called MyFunction that will be a "serverless" Function based microservice
- Connect the MyFunction Function to the API Gateway
- Expose the MyFunction microservice as a MyFunction API Endpoint that we can call through an API Request using cURL

Running Express Serverless Platform

Express Serverless Platform can run in any Kubernetes cluster on any cloud - private or public. It's a true multi cloud solution.

Out of the box, Express Serverless Platform runs serverless functions in the Kubeless serverless engine in your Kubernetes cluster via the Serverless Framework. By auto deploying serverless functions via the Serverless Framework, you can avoid vendor lock in and use any cloud provider or on premise solution.

Express Serverless Platform using the Serverless Framework will also allow you to run serverless functions in a public cloud provider's proprietary offering if you sign up for an enterprise level plan. In this comparison, the cloud provider's offering for serverless would be AWS Lambda.

Sign-up Incentives & Pricing

LunchBadger
The Express Serverless Platform is free for 14 days in a trial environment.

AWS Lambda Pricing
AWS has a free tier for 1 Million requests per month and 400000 GB seconds of compute+memory time or more specifically - the amount of memory consumed per amount of time. Beyond that you are charged $0.20 per 1 million requests and for the duration of the computed+memory time of $0.00001667 per GB/s. To invoke Lambda via HTTP, AWS API Gateway costs are $3.50 per 1 Million requests plus the data transfer costs.

Features Comparison

The feature summary below combines AWS Lambda and API Gateway compared to the Express Serverless Platform equivalent. **For more authentication and user profiles and provisioning a separate product called AWS Cognito is necessary.**

	AWS Lambda + API Gateway	Express Serverless Platform
General		
On Premise		
Runs on any Public or Private Cloud		
Kubernetes Support		
Configuration & Administration		
Auto Scaling		
Visual Designer		
Visual Orchestration		
Git Access		
Serverless		
HTTP Functions		
Event Functions		
Model Functions		
Node.js Functions		
Python Functions		
.NET Core		
Go Functions		
Ruby Functions		
PHP Functions		
Java Functions		
Docker Image		
Auto REST Scaffolding		
Events and Triggers		
Pre-Built Connectors		
API Management		
HTTPS		
CORS		

Features Comparison

	AWS Lambda + API Gateway	Express Serverless Platform
Basic Auth		
OAuth2		
Key Authentication		
JWT		
Finegrain Access Control		
Rate Limiting		
Quotas		
Request Transformation		
Pipeline Driven Conditional Actions		
Pipeline Driven Expressions		
Response Transformation		
Consumer Management		
API Portal		
Plugins Framework		
Open Source Ecosystem		
Configuration Database		

1. *Auto REST scaffolding for Models only*
2. *HTTP triggers only at this time (general pub/sub in roadmap)*
3. *Basic Auth, Key Auth, JWT, and OAuth2 is through another product AWS Incognito*
4. *API REST Testing Interface can be integrated into an existing portal*