DEVELOPER SURVIVAL GUIDE

EXPRESS SERVERLESS PLATFORM AND AZURE

LunchBadger

Express Serverless Platform and Microsoft Azure

ABSTRACT

We’ve created this comparison 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 for AWS and "vanilla" Microsoft Azure across multiple dimensions and “at-a-glance".

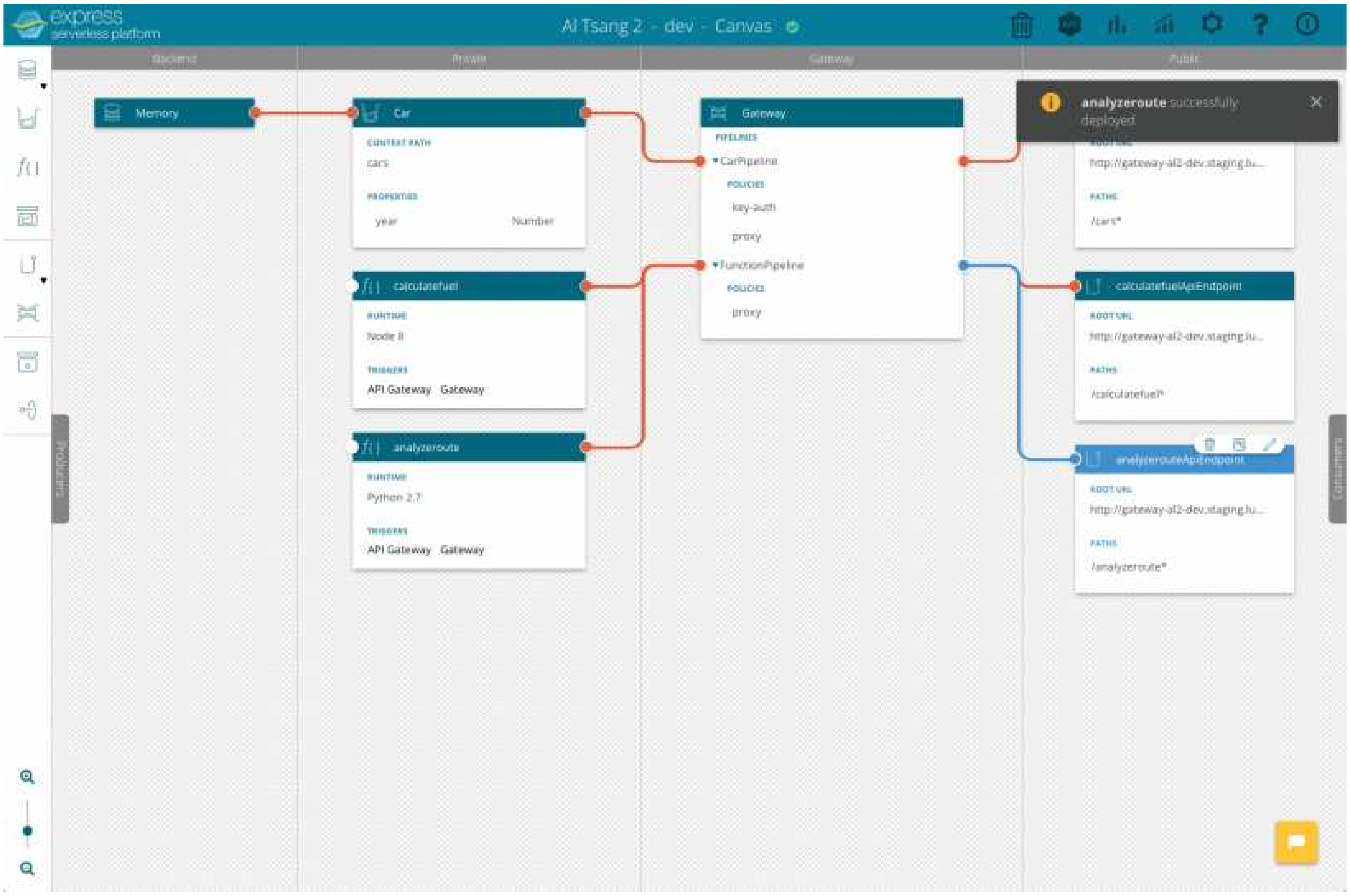
What is Express Serverless Platform?

Express Serverless Platform provides a uniform and seamless development experience for microservices and APIs as simple 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 multi cloud and or hybrid cloud strategy.

Each deployed instance of Express Serverless Platform allows you to take advantage of a cloud’s native infrastructure without coupling your code to its proprietary interfaces.



analyzeroute successfully deployed

Express Serverless Platform’s visual interface is called the Canvas. The Canvas gets everyone on the same page on what microservices are there and where and how they are running.

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.

Development can also be done locally and pushed via git bypassing the Canvas, leaving it for only orchestration and visualization if desired.

Once development is complete, the microservices application is deployed to any number of environments for testing, staging and production. Express Serverless Platform saves massive amounts of time by automating this complicated and unwieldy process while integrating into CI/CD processes in place.

Express Serverless Platform is the first of its kind platform to automate and manage both container and serverless microservices. By doing so, the platform seamlessly gives you a unified view of what your microservices application looks like and how they’re orchestrated across different pieces of infrastructure.

How Express Serverless Platform makes Azure Better

There are several key factors why customers have chosen to use Express Serverless Platform on top of Azure even if it is their only cloud.

Speed

Writing microservices as Azure Functions and manually wiring them up to be exposed as APIs through as HTTP endpoints is a very time consuming process. Each API endpoint has to be manually defined, changed and managed. Defining API endpoints is extremely low level with a myriad of details needing to be specified.

Express Serverless Platform speeds up the process by utilizing templates, conventions and automation through superior tooling to build and expose functions, written as Lambda and container based microservices, as APIs.

Flexibility

Function based microservices may need to be in long running containers or called on demand through serverless infrastructure or through some combination based on the use case. Azure does not provide any integrated way to do this easily.

Express Serverless Platform provides a seamless and flexible experience for the developer so that they can focus on the application logic and utilize both types of infrastructure for the right use cases and applications.

Extensibility

Azure Functions allow you to invoke via an HTTP endpoint; however, If you want enterprise grade API management features, be prepared to pay for them through Azure API Management. Azure API Management is probably one of the easier to use API gateways from a public cloud provider, but fronting Azure functions means going through separate processes that are not well tied together and adding on considerable cost as your production deployment grows.

Express Serverless Platform utilizes Express Gateway as its built in API gateway. Express Gateway has a rich ecosystem of prebuilt plugins to easily meet enterprise requirements without starting from scratch through its Express open source community roots. Express Gateway is entirely Node.js and JavaScript making it completely transparent and extensible from plugins all the way through to its core.

How Express Serverless Platform makes Azure Better Cont'd

Cost

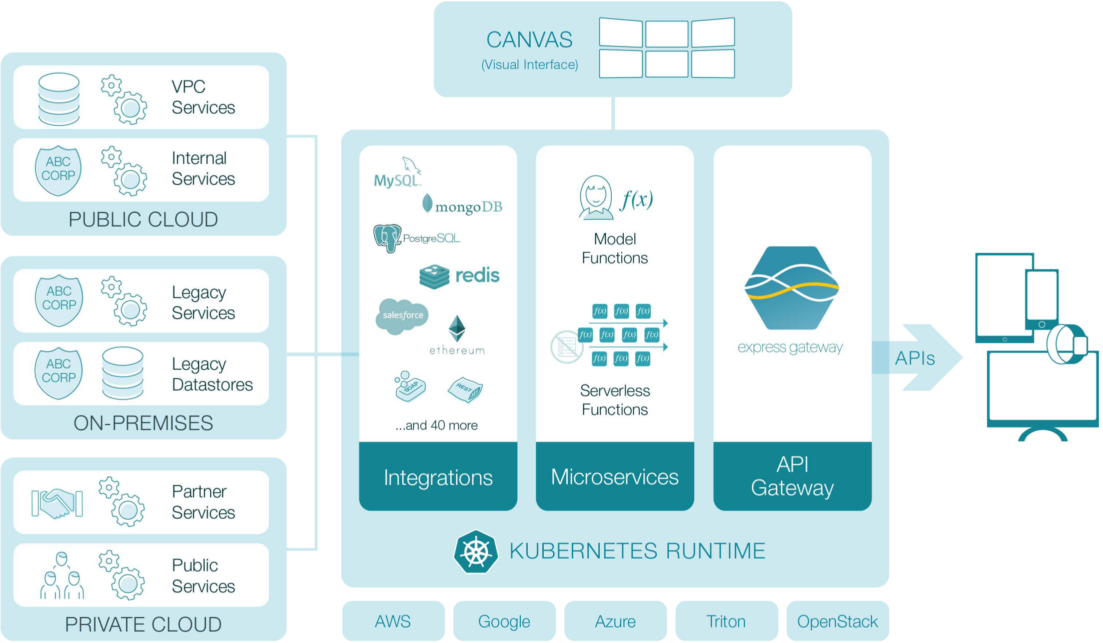
To meet enterprise requirements, it is not uncommon to be required to utilize multiple offerings when operating in Azure. Utilizing Azure offerings often cost next to nothing to get started, but increasingly are reaching a faster inflection point where costs become a significant consideration especially over the long term. When utilizing more offerings this cost is significantly magnified and quickly add up.

Express Serverless Platform helps minimize these costs by reducing the number of Azure offerings required overall. It also comes with a full complement of enterprise functionality out of the box that is already integrated.

Express Serverless Platform bypasses hidden cost while taking advantage of cost saving provisions. For example - the cost savings realized by utilizing on demand computing billed by actual utilization, such as Azure Functions, is preserved by not having to use Azure API Manageement where the majority of the operational cost lies invoking Azure Functions.

Additionally, Express Serverless Platform is licensed for unlimited use. The cost scales linearly with the value that you receive not on the volume of its usage.

Express Serverless Platform Features & Architecture



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

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

Express Serverless Platform also 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

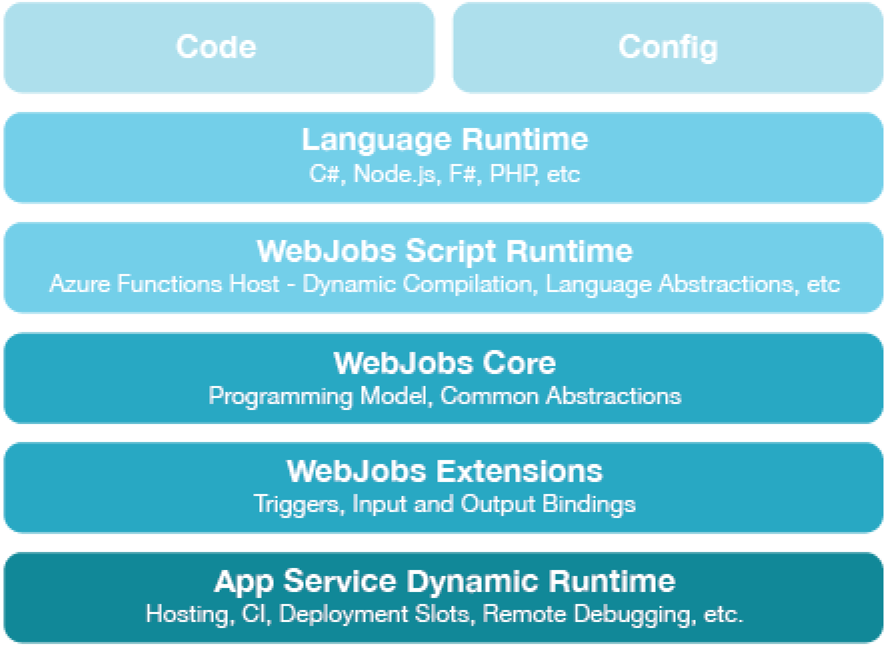
Express Serverless Platform Features & Architecture Cont.

Features Include

- Composition of Serverless Functions
- Composition 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 API gateway, serverless engine, container orchestrator, and GUI

Further reading: LunchBadger Documentation

Azure Functions Features & Architecture


Code
Config
Language Runtime
C#, Node.js, F#, PHP, etc
WebJobs Script Runtime
Azure Functions Host - Dynamic Compilation, Language Abstractions, etc
WebJobs Core
Programming Model, Common Abstractions
WebJobs Extensions
Triggers, Input and Output Bindings
App Service Dynamic Runtime
Hosting, CI, Deployment Slots, Remote Debugging, etc.

Features Include

Azure Functions are Microsoft Azure's equivalent to AWS Lambda. Like Lambda, Azure Functions let you write custom functions and expose them via an HTTP interface using Azure API Management. Express Serverless Platform offers a similar stack that lets you write Node.js APIs and expose those APIs through Express Gateway without vendor lock in.

- Deploys, runs and scales your code on a fully managed compute platform
- Spin up Cloud Functions on-demand and back down in response to events in the environment
- Build HTTP-based API endpointsLog and monitor performance
- Trigger Functions by leveraging CRON expressions, changes in Azure Storage Blob containers, changes to Azure Queues, messages from Service Bus, or HTTP triggers.
- Create, Manage and Deploy Functions provisioned with a Git endpoint Access local development and Continuous Integration using services like Visual Studio Team Services, GitHub and BitBucket.
- Access local development and Continuous Integration using services like Visual Studio Team Services, GitHub and BitBucket.
- Browser-based user interface allowing you to create scheduled or triggered pieces of code in JavaScript, C#, Python and PHP
- Implement code triggered by events occurring in Azure or third party service as well as on-premises systems

Further reading: Azure Function Features

How Express Serverless Platform works with Azure

Installing Express Serverless Platform on Azure

Run in any Kubernetes cluster on any cloud - private or public. It’s a true multi cloud solution.

Express Serverless Platform provisions and installs itself in your Azure account through your Azure keys. The installer is automated and requires minimal interaction.

It is installed on Kubernetes running on your Azure account. It is your choice on how Kubernetes is run, but the recommendation is Azure Kubernetes Service (AKS), Azure's managed Kubernetes offering. The installer provisions and utilizes AKS by default.

Once Kubernetes is up and running the automated installer deploys ESP through Helm, the Kubernetes package manager.

Running Express Serverless Platform on Azure

By default, Express Serverless Platform runs wholly self contained within Kubernetes, That means, application and infrastructure components all are easily managed through one common runtime.

These components include:

- model functions
- serverless functions
- integrations (connectors)
- API gateways

Serverless functions are abstracted using the Serverless Framework behind the scenes automatically for you.

Serverless functions run side by side as containers within your Kubernetes cluster, managed and controlled. Express Serverless Platform does this automatically and seamless through Kubeless an open source serverless engine that is Kubernetes native.

Using the Serverless Framework, Express Serverless Platform will also allow you to run serverless functions in Azure Functions to take advantage of cost of compute incurred only while running.

Getting Started with the Express Serverless Platform Trial

Getting started with Express Serverless Platform is dead simple.

LunchBadger offers a free 14-day trial of Express Serverless Platform with no credit card required.

The trial runs in LunchBadger's cloud so you don't need to have a public cloud account nor worry about installing anything on premises - simply sign up

The trial provides a complete walkthrough building a microservices application and its API through a series of guided and interactive steps - all in less than five minutes.

Once built, you'll trace your API workflow from public API Endpoints to your in-memory data source and be able to experiment with a live and running microservices application to see the value of the time savings and functionality it provides.

QuickStart

The following is an overview of the getting started experience provided by the self guided walkthrough.

- Deploy and use a 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

Pricing

LunchBadger

Express Serverless Platform is offered as a free 14 day a trial.

License plans come packaged with a number of users, microservice functions, and API gateways at a low cost that predictably scales as you realize the value of what the platform brings.

There are no hidden or extra compute fees for operating in Azure. Operating costs are transparently passed onto you by running in your account.

Azure Functions Pricing

Azure Functions consumption plan is billed based on per-second resource consumption and executions.

Consumption plan pricing includes a monthly free grant of 1 million requests and 400,000 GB-s of resource consumption per month per subscription in pay-as-you-go pricing across all function apps in that subscription.

Azure Functions Premium plan provides enhanced performance and is billed on a per second basis based on the number of vCPU-s and GB-s your Premium Functions consume. Customers can also run Functions within their App Service plan at regular App Service plan rates.

Azure API Management

At the time of writing this comparison, there are three plans for production workloads offered for API Management - BASIC, STANDARD and PREMIUM. The cost of utilizing the API Management features are $0.21/hour, $0.95/hour and $3.83/hour. Azure is also previewing a Per Consumption plan as well for pay per use where the first million calls are free and then undisclosed pricing per call.

Features Comparison

The feature summary below combines Azure Functions and Azure API Management compared to the Express Serverless Platform equivalent. Like most public cloud offerings, you must subscribe to multiple proprietary offerings to accomplish the same experience provided by Express Serverless Platform.

Feature	Microsoft Azure + Azure Functions + API Management	LunchBadger **Express Serverless Platform**
General		
On Premise	🔴[1]	🟢
Runs on any Public or Private Cloud	🔴	🟢
Kubernetes Support	n/a	🟢
Configuration & Administration	GUI, CLI, API	GUI, YAML, CLI and API
Auto Scaling	🟢	🟢
Visual Designer	🟢	🟢
Visual Orchestration	🔴	🟢
Git Access	🔴	🟢

Serverless		
HTTP Functions	🟢	🟢
Event Functions	🟢	🔴
Model Functions	🔴	🟢
Node.js Functions	🟢	🟢
Python Functions	🔴	🟢
F# Functions	🟢	🔴
.NET Core	🟢	🟢
Go Functions	🔴	🟢
Ruby Functions	🔴	🟢
PHP Functions	🔴	🟢
Java Functions	🔴	🟢
Docker Image	🔴	🔴
Auto REST Scaffolding	🔴	🟠[2]
Events and Triggers	🟢	🟠[3]
Pre-Built Connectors	Azure Storage and Cosmos DB	8/20+[4]

API Management		
HTTPS	🟢	🟢
CORS	🟢	🟢
Basic Auth	🟢	🟢
OAuth2	🟠[4]	🟢
Key Authentication	🟢	🟢
JWT	🟠[5]	🟢
Finegrain Access Control	🔴	🟢
Rate Limiting	🟢	🟢
Quotas	🟢	🔴
Request Transformation	🟢	🟢
Pipeline Driven Conditional Actions	🔴	🟢
Pipeline Driven Expressions	🔴	🟢
Response Transformation	🟢	🟢
Consumer Management	🔴	🟢
API Portal	🟢	🔴[6]

Plugins Framework	🔴	🟢
Open Source Ecosystem	🔴	🟢
Configuration Database	n/a	In-Memory Redis

1. *Azure Functions has a development only sandbox that can be run locally*
2. *Auto REST scaffolding for Models only*
3. *HTTP triggers only at this time (general pub/sub in roadmap)*
4. *OAuth2 is validation only and token is issued by Microsoft AD, another offering*
5. *JWT is inspection only and issued by Microsoft AD, another offering*
6. *API REST Testing Interface can be integrated into an existing portal*