The Benefits of Serverless Applications in a Data-Rich World

Here's a breakdown of the many pros (and several cons) to keep in mind when considering adopting a serverless architecture.

by Dennis O'Reilly MVB · Jun. 24, 16 · Cloud Zone

Deploy and scale data-rich applications in minutes and with ease. Mesosphere DC/OS includes everything you need to elastically run containerized apps and data services in production.

Serverless applications are the source of much confusion of late. For one thing, they're not "serverless" at all. In fact, it's more accurate to refer to them as "multiserver apps" — their components are distributed among cloud servers far and wide and assembled on demand.

This begs the question, "Do you know where your app is?" In today's microservices-based computing environments, the question becomes meaningless. Exactly where particular pieces of a full-blown application live matters much less than the network's ability to retrieve and assemble the required components accurately, and in a timely manner, when and where they're needed.

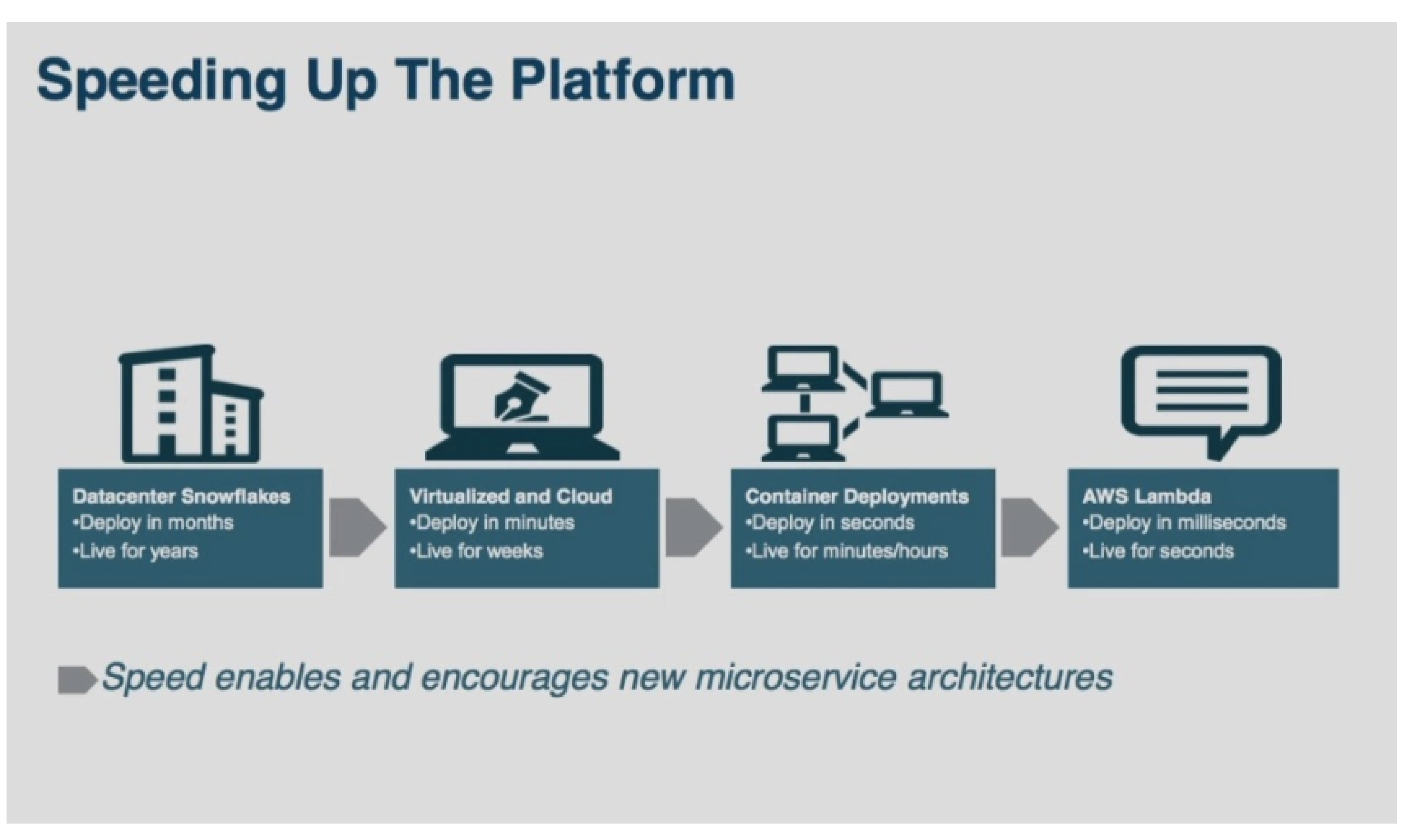
Such widely distributed applications demand a new approach to development, deployment, maintenance, and updates. Forbes' Janakiram wrote in a March 22, 2016, article that the growing interest in serverless apps is driven by two trends: mobility and the Internet of Things. In both cases, applications need to follow the IFTTT model: respond in an instant as current circumstances dictate in terms of location, time, input, and context.

A serverless framework meets the needs of such on-the-spot delivery of services to users via microservice-based apps. The framework is comprised of a distributed repository of code snippets that assemble themselves in response to an external event – the "if this, then that" trigger.

How Serverless Apps Can Benefit Businesses of All Types

In a February 26, 2016, post, InformationWeek's Charles Babcock describes scenarios in which the serverless approach meets a real-time business need. A customer placing an order wants confirmation that a critical part is available before committing to the purchase. A business analyst uploads a JavaScript snippet (a microservice) to AWS that adds a parts-confirmation box to the order form. Selecting the box fires the snippet that triggers the required inventory check via the AWS Lambda cloud service, which is based on a serverless framework.

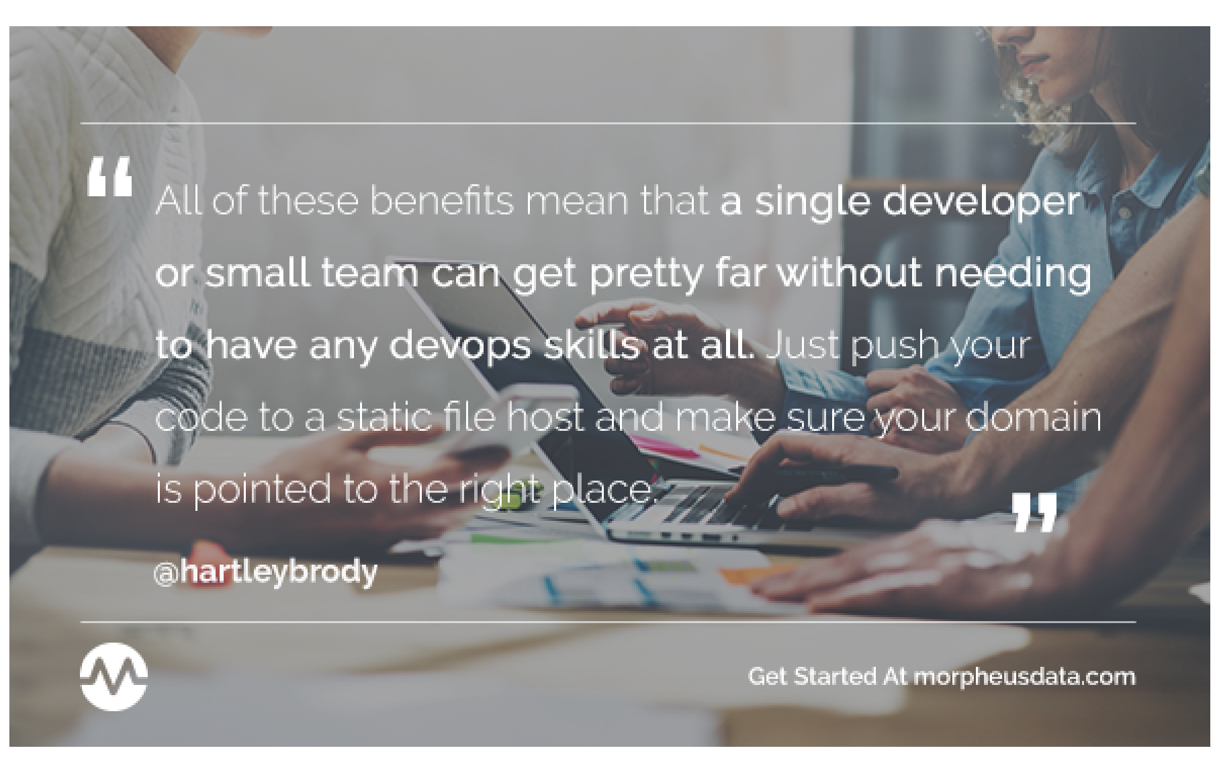
With Lambda, the task is completed in short order without involving another application or requiring that a virtual server be spun up. In addition to being done for only a few cents' worth of AWS runtime, the snippet scales automatically to accommodate changes in demand. Any number of a business' microservices would reside in the public cloud and would be called on when needed. The business receives the benefits of its applications without being responsible for any in-house servers – that's what makes the applications "serverless."


Speeding Up The Platform
Datacenter Snowflakes •Deploy in months •Live for years
Virtualized and Cloud •Deploy in minutes •Live for weeks
Container Deployments •Deploy in seconds •Live for minutes/hours
AWS Lambda •Deploy in milliseconds •Live for seconds
Speed enables and encourages new microservice architectures

AWS Lambda introduces on-demand, instantaneous, and short-lived application deployment. (Credit: Adrian Cockcroft, Battery Ventures)

What stays in-house is the development team — the brains of the operation. Developers are responsible for ensuring that the apps are accessible and that they run smoothly on all devices. The biggest difference between mobile apps and their web-delivered counterparts is where the bulk of the application logic resides. On tablets and phones, much of the app logic lives on the device, whereas web apps put their logic on the Internet server.

Whereas web apps get by with a thin user interface inside the browser, mobile apps rely on the device to ensure a pleasurable user experience: the phone or tablet hosts the business logic, and the Internet data center's server provides the bandwidth. A big upside for developers is that the cloud service provider is responsible for the database server, the messaging system, and security functions.


“All of these benefits mean that a single developer or small team can get pretty far without needing to have any devops skills at all. Just push your code to a static file host and make sure your domain is pointed to the right place.”
@hartleybrody
Get Started At morpheusdata.com

When a Serverless Approach May Not Be the Best Choice

Running your app with 99.99 percent availability and no servers to maintain sounds too good to be true, which naturally causes seasoned developers to think, "There's gotta be a catch." In a March 18, 2015, post, tech blogger Hartley Brody examined some of the impediments to creating serverless apps.

Topping the list is the need for a centralized datastore across multiple clients. Such situations require a database server and application logic that runs on top of it. Possible workarounds are to use a third-party API, such as Firebase or Parse, or to rely on an API from your company's CRM system or other internal data store. Brody also recommends looking for a way to do without the database, if users don't need an account. This limits the amount of data you need to collect and store.

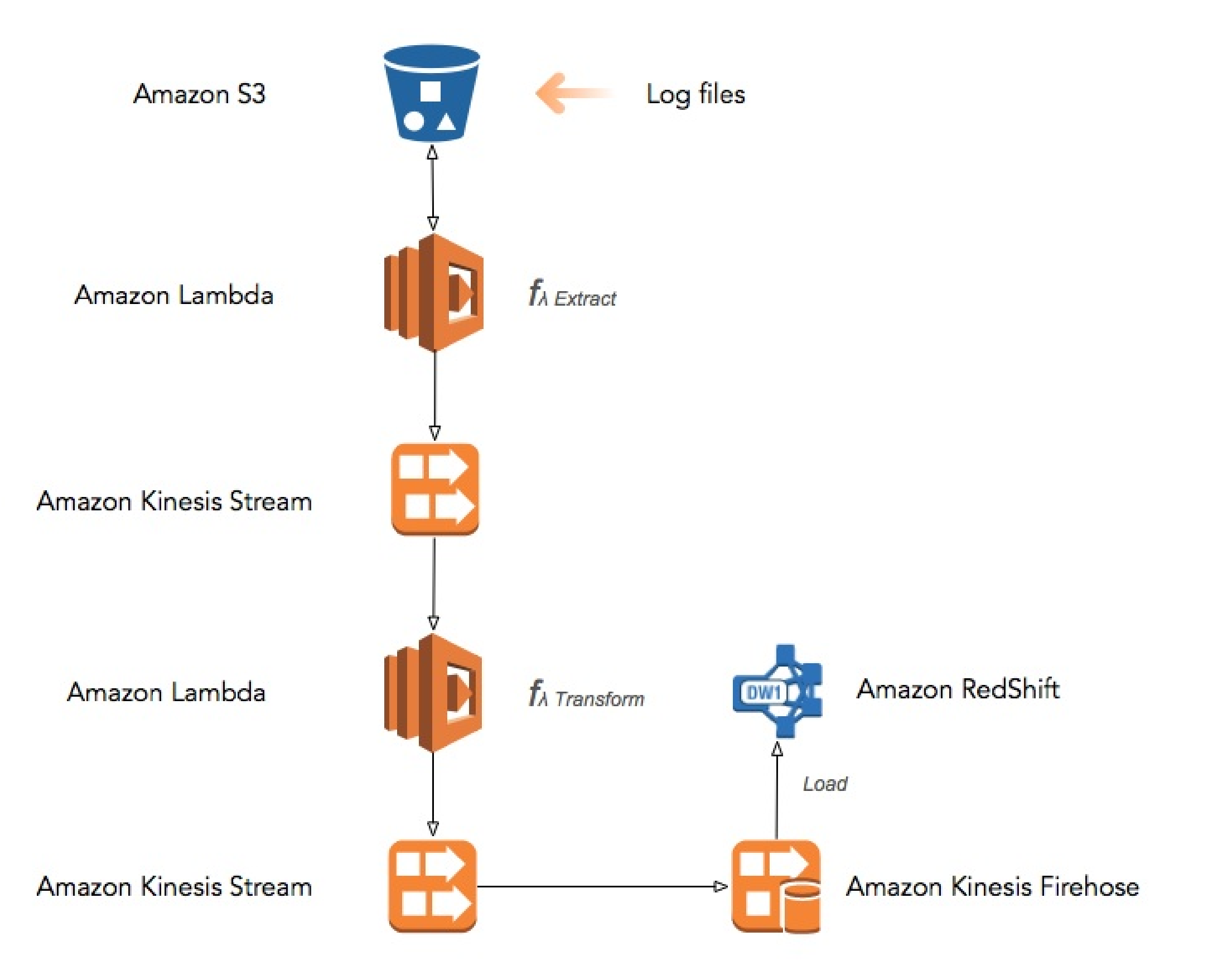
Serverless apps can also be stymied by the need to integrate with third parties for storing data, sending messages, user analytics, and other operations. Whenever you need to authenticate users, you're looking to a server to avoid sending API keys and other credentials to clients. Similarly, apps that rely on sensitive, private business logic should run that logic on a secure server.

Another shortcoming of the serverless approach is the lack of information following an application crash. A 503 occurring on a web server generates a pretty comprehensive stack trace from the server logs, but when a JavaScript fails on a client, you're left completely in the dark. That means you have to anticipate all the ways the app could crash or otherwise misbehave beforehand, and devise a way to communicate what happened back to your system, and to the user.

Finally, JavaScript can be overwhelmed when tasked with processing large amounts of data. On a web server, you can move some of that processing offline, then store the result and notify the user when it's available. The Web Worker API can help prevent overtaxing a JavaScript app.

Forecast for Serverless: Sunny but Variable

In his March 22, 2016, post listing the five top serverless frameworks, Forbes' Janakiram puts AWS Lambda in the top spot for its integration with other AWS services, and its support for mobile and IoT developers. For voice applications in particular, AWS Lambda seamless integration with the Alexa Skills Kit makes it a good choice for Amazon Echo voice-activated apps.



AWS Lambda integrates smoothly with other Amazon services, such as Kinesis Stream, Kinesis Firehose, and RedShift to create a serverless ETL pipeline, as in this example. Credit: Abhishek Tiwari

Google's recently acquired Cloud Functions will be hosted as containers running on the Google Compute Engine, and while it will support only Node.js initially, other languages will likely be added over time. An obvious advantage of Google Cloud Functions is built-in API support for Gmail, Maps, Cloud Messaging, and other Google services.

Another likely competitor in serverless frameworks is Iron.io's Project Kratos, which runs existing AWS Lambda functions packaged as Docker containers in public, private, and mixed cloud environments. IBM positions its OpenWhisk project as an open source alternative to AWS Lambda; OpenWhisk supports Node.js and runs snippets written in Swift; it is integrated with the IBM BlueMax PaaS environment based on CloudFoundry and integrates with third-party services via WebHooks.

While Microsoft has not yet joined Amazon, Google, and IBM in offering a serverless framework, the Azure Web Jobs runtime environment can be used to invoke code snippets on demand or via scheduled cron jobs. Analysts expect a complete serverless framework from Microsoft based on the company's Service Fabric PaaS environment.

Discover new technologies simplifying running containers and data services in production with this free eBook by O'Reilly. Courtesy of Mesosphere.

Topics: AWS LAMBDA , FUNCTIONS , FRAMEWORKS , SERVERLESS , PIPELINE , CLOUD , APPS , IOT APP DEVELOPMENT