

TECHNOLOGY / TUTORIALS / GLOBAL

Reactive Frameworks, Microservices, Docker and Other Necessities for Scalable Cloud Native Applications

2 YEARS AGO, BY [VIVEK JUNEJA](#)

+

A lot of startups go wrong when adopting cloud services. For starters, the cloud becomes yet another hosting provider with more competitive rates. I have personally seen deployments which are mere fork-liftings of existing in-house deployment to a set of virtual machines and storage services. Those who take this road get frustrated in handling the daily chores of managing their presence on a cloud environment, while some move over to other competing services. Most of these merely try to make their systems take advantage of cloud

environments by adopting certain services or retrofitting an existing application to a cloud provider. This is the just-being-cloud-ready problem and [massive outages](#) of popular services is a signal that [needs attention](#).

Cloud Native means developing services that truly embrace the nature of cloud environments — which is fragile, failure-prone and scaled horizontal.

Over the last few years, startups have come to understand that distributed infrastructures such as [cloud environments do have pitfalls](#), and the onus of making their applications work the right way on it still remains with the startups themselves. A broad range of new ideas have arrived, thanks to some early adopters who have learned the hard way how to sustain themselves in this new paradigm. Also, there has been progress among cloud providers, thanks to new entrants and swelling competition.

Key Indicators of a Cloud Native Application

The application:

1. Is aware of the infrastructure and dynamically scales resources – servers, storage and network, etc.
2. Makes fewer assumptions about the users, environment, resources that it works on. This means the software design includes measures for handling large variations in usage, fault in infrastructure, and unreliable resources.
3. Has been designed to be aware of the economics of its sustenance. This means it is cost-aware and can be configured to make decisions about its operational cost and utilization. This improves the overall cost-to-performance ratio of the application.
4. Is resilient to disaster and can work in degraded mode rather than having a complete outage.
5. Can be deployed across multiple infrastructure options or across hybrid form (private and public cloud infrastructure environments).
6. Is able to be incrementally deployed and is completely testable.

What Startups Need to Know About Building Cloud Native Applications

1. Invest in standardizing the choice of infrastructure and software platform across all environments used for the application — development, test, performance and production.
2. No one size fits all — look out for mixing cloud platform providers by identifying which one works better for your application.
3. Early investing in the monitoring and measurement of the application, and even tracking [cloud outages](#), to help with routing traffic to healthy infrastructures.
4. Building asynchronous communication wherever possible in the application architecture would allow for loose coupling amongst components. Asynchronous behavior is at the heart of cloud native applications.
5. Build transparency across all levels of application to ensure complicated systems can be built on easily.
6. Continuous Capacity Planning, rather than upfront excessive capacity or under capacity: the cloud native requires just in time capacity and continuous, automatic assessment of infrastructure requirements.

Patterns and Technologies Powering the Cloud Native Way

Reactive frameworks like [Rx](#) and [Typesafe](#) are generating strong momentum in adoption amongst development communities and becoming a common way to build applications that are inherently scalable and resilient to failures. A good place to start is by visiting the [Reactive Manifesto](#) and learning more on this.

[Netflix OSS](#) provides a toolkit for startups to build cloud scale applications without the need for heavy lifting. Netflix is one of the shining stars of public cloud infrastructure usage, and has been at the forefront of innovation in this space. Netflix OSS is the [continuous aggregation of all such best practices](#) in the form of open source toolsets, that is essentially a Swiss Army knife for a startup embracing cloud.

Software containers like [Docker](#) enable flexibility for applications across all range of infrastructure — public to private cloud deployments and even regular data center. Together with [CoreOS](#), the operating system that treats containers as first-class citizens, it is [reshaping how applications are packaged](#) and deployed.

Building on cloud scale means having multiple services orchestrated together to form service behaviors. [Microservices architecture](#) is slowly emerging as a championed way to design and build large scale systems. It not only solves the problem for loose coupling and software modularity, but is a boon to continuous integration and deployment. Changes made to one part of the application which require massive changes all across are a bane to continuous deployment. Microservices aim to resolve this situation. Some early [microservice PaaS](#) have emerged but are yet to have mainstream adoption. Microservices are also being used to [develop Anti-Fragile systems](#).

[Cost-aware applications](#) perform continuous monitoring of utilization and the costs incurred. The monitored data is essential to create decision points that the application can use to either scale or diminish the use of resources. Every cloud native application having access to the provisioning APIs means that necessary cost governance must be engineered into them. [This opens up to very interesting possibilities](#).

Making Reliable Over Unreliable

Cloud native applications should have attributes that give them the ability to scale. These apps must have inherent characteristics such as randomness, resiliency, fail safe capabilities, redundancy and cost optimization. These attributes are essential for a software system to not only work but to also excel at cloud scale. If startups need the scale and effectiveness of a cloud environment, they need to incorporate the essential building blocks of massive scalability on day one. This means embracing cloud native rather than just being cloud ready.

Vivek Juneja, an engineer based out of Seoul, is focused on cloud services and mobility. He currently works as a Solutions Architect at Symphony Teleca, and is a co-founder of the Amazon Web Services user group in Bangalore. He started working with cloud platforms in 2008, and was an early adopter of AWS and Eucalyptus. He is also a technology evangelist and speaks at various technology conferences in India. He writes @ www.cloudgeek.in and

www.vivekjuneja.in, and loves grooming technology communities. You can also reach him by email: vivekjuneja@gmail.com

Feature image [via Flickr Creative Commons](#)

[CLOUD NATIVE APPLICATIONS](#)[DOCKER](#)[MICROSERVICES](#)[REACTIVE FRAMEWORKS](#)

THE NEW STACK UPDATE

A digest of the week's most important stories & analyses.

Subscribe

We don't sell or share your email. Occasionally, we send updates and useful info.

RELATED STORIES



ANALYSIS / CONTRIBUTED / GLOBAL

How to Monitor Containers with cAdvisor and Wavefront

5 DAYS AGO, BY [EVAN PEASE](#)

ANALYSIS / EVENTS / PODCAST / NORTH AMERICA

OSCON 2016 Reaches Out for a More Diverse Audience

6 DAYS AGO, BY [C.K. OLIVER](#)

[View / Add Comments](#)

SPONSORED FEED



Virtual Reality and Augmented Reality are Almost Ready, Are You?

APRIL 22, 2016



Yahoo! JAPAN Corporation builds on Pivotal Cloud Foundry and OpenStack

APRIL 22, 2016



Greenplum Can Now Do Hybrid Queries Across On-Premise & AWS S3 Data

APRIL 20, 2016



Simplifying Microservices Deployment: Multi-Job Manifests

APRIL 20, 2016



The Mesosphere guide to getting started with DC/OS

APRIL 20, 2016



Mesosphere open-sources DC/OS container orchestration solution

APRIL 19, 2016



Apr 12 & 13 Cloud Expo Europe 2016

APRIL 10, 2016

SECTIONS

[Technology](#)

[Culture](#)

[Tutorials](#)

[Podcasts](#)

[eBooks](#)

[Events](#)

FOLLOW

[Twitter](#)

[Facebook](#)

[YouTube](#)

[Soundcloud](#)

[LinkedIn](#)

[Slideshare](#)

[RSS](#)

THE NEW STACK

[About](#)

[Contact](#)

[Sponsors](#)

[Privacy Policy](#)

© 2016 The New Stack. All rights reserved.