ActualTech Media

Tintri

Current Trends Impacting Hybrid Cloud

Author: Scott D. Lowe
Partner, ActualTech Media
November 2016

Contents

Introduction

You already know that the hybrid cloud—a compelling combination of hand-picked public cloud services combined with strategic retention of a local private cloud—is the future of IT. Although there is a lot of focus on the word cloud these days, cloud is not really an endgame. In reality, the journey toward cloud principles is the key driver for organizational improvement. In this paper, you will learn about how current trends—part of the journey—are impacting people's hybrid cloud plans.

Operations Trends

The data center is a trendy place right now (it feels kind of weird and awesome to write that). There are a number of local data center trends that deserve to be called out for their role in helping organizations bring cloud-like characteristics to the local data center.

Increasing Virtualization

Virtualization does not equal cloud. Yes, we said this earlier in this book as well, and it's still true. However, virtualization is perhaps the most critical element in getting to an environment that can be considered cloud-like. As you look at cloud providers out there, one fact becomes abundantly clear: All of them run workloads that are virtualized in some way. Amazon's workloads run on Xen; Microsoft Azure's workloads run on a combination of Azure VMs and Hyper-V.

Why is this? There are a whole lot of reasons, and here are the most important:

- **Hardware efficiency**. Virtualization has made it possible to push hardware to its very limits. This is a good thing and it reduces overall hardware cost since you don't need to buy individual servers for every workload.
- **Abstraction**. From an operation perspective, the ability to abstract operating systems and applications from hardware has made it possible to consider underlying hardware as almost an afterthought. As long as it has the capability to support its workloads, it doesn't matter on which system workloads actually operate. Here's the key item to remember: abstraction turns hardware-based servers and components into software. Software is far more easily managed and manipulated than hardware, which brings us to...
- **Workload mobility**. Enabled by abstraction, mobility is the ultimate outcome from virtualizing workloads and has imbued supported workloads with the ability to be shifted to new platforms and data centers for availability and copied to various places for disaster recovery purposes.

Continuously increasing the level of virtualization for new workloads enables more and more automation and efficiency in data center operations.

Analytics-Driven Monitoring and Planning

You can't make good decisions without good data! Executive teams have known this forever and they're always looking for good organizational metrics. In more recent years, however, this kind of thinking has come to IT infrastructure as well. Today, IT decision makers consider what's actually happening in their infrastructure environments in order to plan for the future and to react to operational performance issues. The ability to access key performance indicators at *every* level of the infrastructure and application stack is absolutely critical to maintaining a high level of performance and avoiding downtime.

Autonomous Operation

The traditional data center requires a ton of attention from a variety of people, each with vastly different, and often expensive, skill sets. The ultimate destination for the local data center, however, is one that manages itself without the need for a lot of people.

This is not to say that there won't be an IT staff that has to take some part in managing the data center, but it does mean that the skills that these IT pros bring to bear may look very different than the ones we see today.

Autonomous data center operations is actually viable, too… when you fill your data center chock full of virtualization and then sprinkle on a dash of analytics. With the right foundation—virtualized—combined with the right level of analytics, it is possible to have a data center that is able to handle its own routine operations. For example, is your CRM application being overloaded by too many new connections from potential customers? Allow your autonomous data center to spin up a series of new virtual web servers to help handle the load. Is your key customer-facing application beginning to run out of storage capacity? Allow your autonomous data center to provision temporary capacity in the public cloud or move low-value data from the local data center to a cloud provider for archiving to free up local capacity.

Industry Trends

Operations trends are one thing, but there are several overall industry trends that are important to understand as organizations make their way to the hybrid cloud.

DevOps

A portmanteau of "development" and "operations", DevOps is a movement that joins together development processes and infrastructure with eventual product outcomes. Under a DevOps-driven methodology, the infrastructure becomes a part of the development process, which streamlines the lifecycle and leads to more frequent deployments with higher levels of success. The ultimate goal of DevOps is to increase an organization's overall business performance.

According to Puppet Lab's 2016 State of DevOps report, highly performing organizations get a number of benefits, including:

- 200 times more frequent code deployments
- 2,555 times faster lead times
- 24 times faster mean time to recover
- 3 times lower change failure rate

All of this leads to lower costs, increased efficiency, much faster time-to-value, and increased revenue.

In a DevOps world, you may hear the phrase *programmable infrastructure*. In essence, programmable infrastructure is generally software-defined and enables developers to treat various infrastructure components just like they would treat a software object. They can treat infrastructure as code, thus enabling their software creations to fully manipulate the very foundation on which those creations operate. Remember the operations trend of autonomous operations?

Web-services

Because there were not enough confusing terms in the IT landscape, someone decided to throw another one into the mix. We're used to big, monolithic architectures, but we also know that such environments are far from flexible and agile. As we make changes to such systems, we risk unintended consequences. Here's a little limerick for you:

99 little bugs in the code.

Take one down, patch it around,

127 little bugs in the code...

Unfortunately, that's a reality for a lot of developers of big systems. To try and combat this and to make it easier to develop systems, web-services have entered the equation.

Loosely coupled with one another, the intent behind web-services is to enable modularity and separation of core functionality in order to improve security and to vastly accelerate development efforts. Web-services depend heavily on APIs that enable deep interfacing between components. They don't need to all be built and managed by a single company, either. Many of today's applications and services have rich APIs that enable inter-product communication, so many web-services from many sources can actually interoperate quite nicely. Of course, a single company can choose to leverage web-services for all its development, too. The beauty of a web-services framework is that it enables discrete development on small components. This is the part that makes it possible to very quickly iterate and accelerate the introduction of new functionality into a development project. There is less testing to do and less that can go wrong.

Containers

If you like virtualization because it allows you to cram more workloads onto a single server than was possible in the glory days of the physical server, then you're gonna love containers! And, if you're hooked on web-services, you'll quickly see why containers have become so popular at DevOps parties.

Imagine, if you will, a world in which you were forced to deploy a new virtual machine for every web-service component. After all, just like you do with monolithic applications, you might want to keep your individual web-services running individually as well, to improve security and performance. With the need to create a software-based server (a virtual machine) and install an operating system, patch it, and then deploy the web-service, you're looking at a *lot* of overhead, as you can see in Figure 1.

With small web-services, the overhead gets pretty intense. You need a more granular way to deploy these services. That's where containers come into the picture. Rather than a separate operating system deployment for every virtual machine, containers all run atop the same operating system instance. On top of that operating system runs a container engine—Docker is the most popular and well-known such engine—and each individual application or web-service gets its own access to the binaries and libraries that support it. Figure 1 gives you a look at how virtual machines and containers compare.

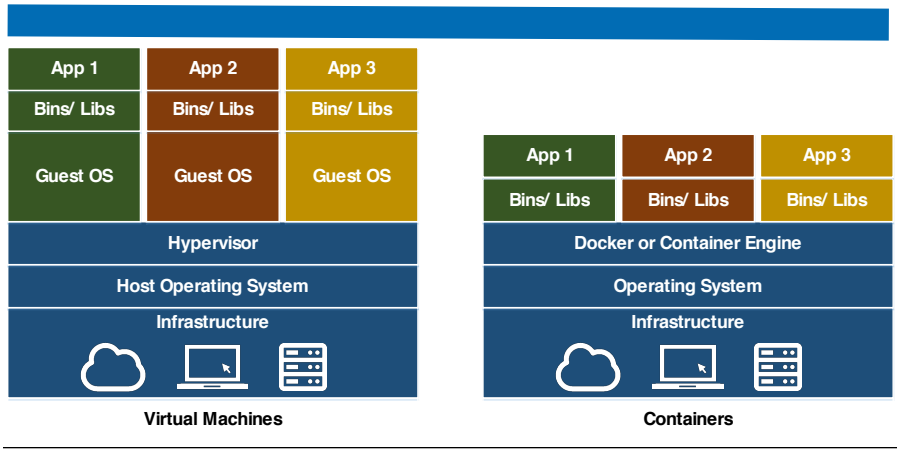

Figure 1. Virtual machines vs. containers

OpenStack

Everything you've heard so far is great, but for one thing. You need a place to run all this stuff. Further, you want that place to be something that gives you cloud-like capability and resources, including compute, storage, and networking, among other components and services. Well, back in 2010, NASA and Rackspace walked into a lab one day and walked out the next day having created the platform known as OpenStack.

OpenStack provides organizations with an infrastructure-as-a-service offering that is fully manageable via a command line, a GUI, and via powerful APIs. Every six months, the OpenStack development community releases new updates to correct problems and add new functionality. OpenStack is a highly componentized system, enabling fast development on individual modules without the risk of those development efforts impacting other modules.

Since this is a paper about storage, let's focus there for a minute. OpenStack's Cinder service provides shared block storage functionality to the platform and also provides the capability for third party storage providers to provide plugin drivers. These drivers enable OpenStack to leverage that company's storage assets as if they were a core part of OpenStack itself.

Why is this important? In an open world, one size does not fit all. The people behind OpenStack understand that other solutions can provide far more functionality than the platform itself can ever do. For example, through a

Cinder driver, OpenStack adopters can deploy a VM-Aware storage solution, which operates directly at the virtual machine level, eliminating the need for complex mapping to LUNs and volumes. This dramatically simplify and improves how you manage storage in an OpenStack cloud deployment.

Serverless Computing

Finally, let's talk about the trend known as serverless computing. Let's get one thing out of the way, though. Servers are still involved, but apparently some marketing person won the battle of the moniker. In reality though, serverless computing is much more about abstraction than it is about actually eliminating the use of servers. The goal is to enable developers to focus on their code rather than on infrastructure, which includes both physical and virtual servers. Today, developers often have to worry about these elements, but this worry shouldn't be necessary.

Let's look at a real-world serverless computing service—AWS Lambda. With Lambda, you're able to upload code to the service and execute this code without having to provision the underlying resources, such as virtual machines. The service itself silently and automatically provisions whatever resources are necessary to carry out your will. And then, once the code is done executing, all the virtual infrastructure that was provisioned is destroyed. Under such a service, you're only paying for the resources that were in use while your code was running.

Why is this important? Think about the model of simply shifting your virtual machines to a cloud provider. If those virtual machines are running 24/7, you're paying 24/7, even if you only use some of them once a week. Under a serverless computing model, you can move ever close to a true "pay as you go" economic model for workload operations.

Summary

We talked a lot about trends in this paper. All of this is great in theory, but we need action! In our next paper, you'll learn about what makes the hybrid cloud actually tick.