PLATFORM9

The Melting Pot in Virtualized Data Centers

PLATFORM9

Table of Contents

1. Introduction 1
2. Virtualization Landscape 1
    1. Search Interest 2
    2. Containers vs Virtual Machines 4
3. Comparative Analysis of Different Virtualization Platforms 7
4. Strategies
    1. Strategy 1: Stay Put 9
    2. Strategy 2: Leverage Diversity with Silos 10
    3. Strategy 3: Leverage Diversity with a Single Pane 12
5. Recommendations 16
6. Summary 17

1. Introduction

Virtualization, a phenomenon that began in the 1960s by logically dividing a mainframe's resources for different applications, has seen numerous advancements that have broadened the usage of the term. This paper talks about recent developments in virtualization that have led to better management of data centers.

This paper also highlights three strategies that could help you adapt to the increasing diversity of choices. The recommendation section highlights our suggestions to succeed in this age of diverse virtualization technologies.

2. Virtualization Landscape

The current virtualization landscape has been dominated by VMware, one of the first players in the segment. However, Hyper-V, a Microsoft product and formerly known as Windows Server Virtualization, has been a constant challenger to VMWare's domination. Hyper-V is a native hypervisor and has a huge base of enterprise customers, a market segment Microsoft where has been successful for many decades.

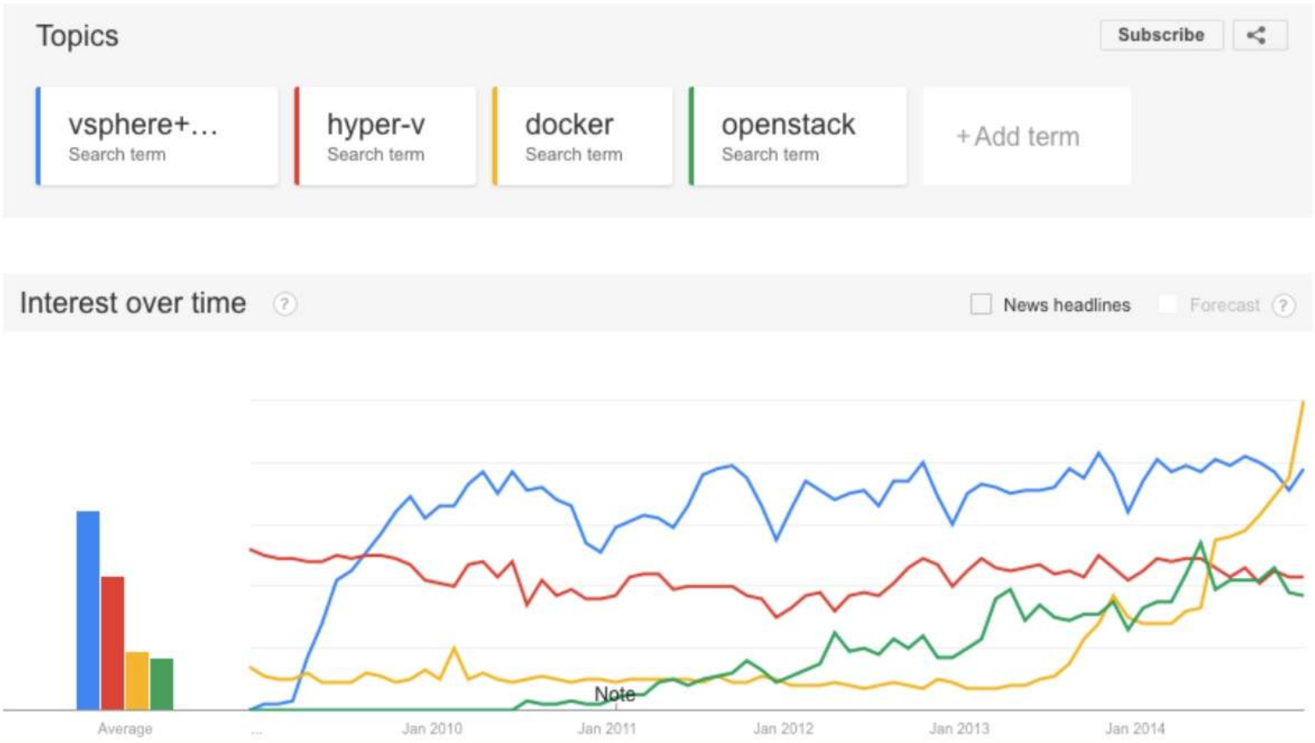
The emergence of OpenStack, an open source cloud computing software platform backed by more than 200 companies, has also fueled more interest in the space. Organizations across the spectrum have deployed OpenStack as an 'Infrastructure as Service Solution' (IaaS) in the last few years.

The rise of Docker, another open source project, aims to provide application portability through software containers and has added new dimensions to the virtualization mix.

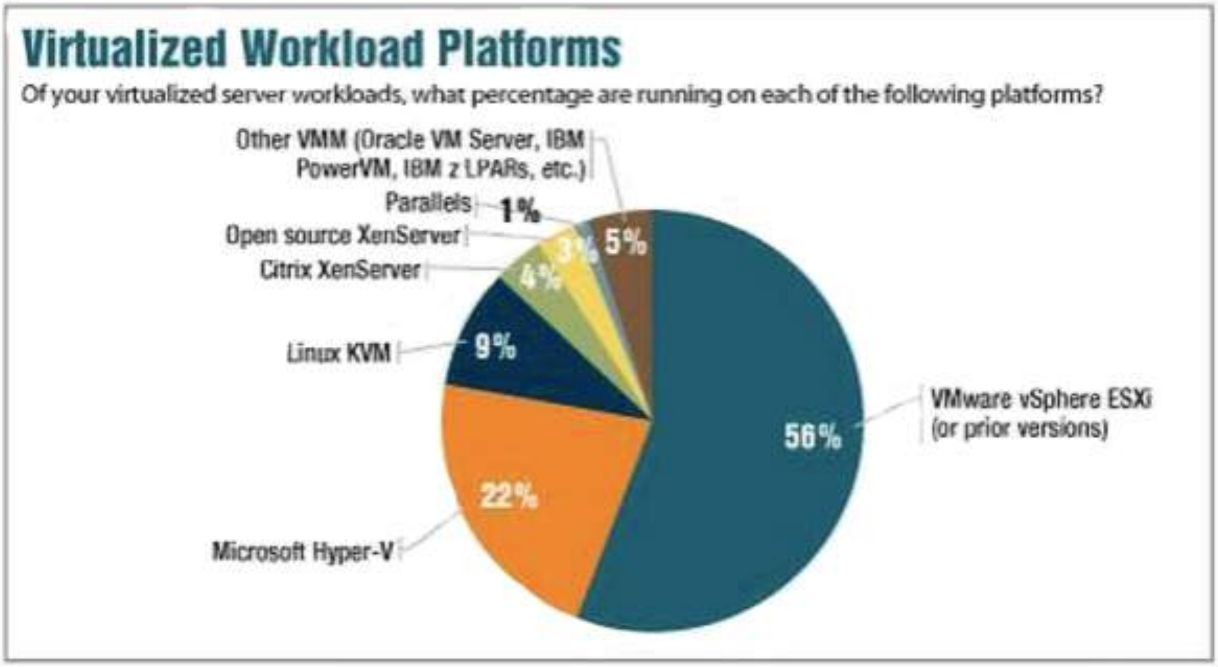
2.1 Search Interest

The picture below shows Google search trends for multiple virtualization technologies. While vSphere (and other associated terms) and Hyper-V have held the fort in terms of sheer search interest numbers, the increase in search interest for these two virtualization technologies has been pretty flat in the last two years.

On the other hand, the green trend line representing OpenStack and the yellow trend line representing Docker have seen increased interest in the last two years. It would be fair to state that the high volume of search interest for Docker could be attributed to it's growing interest in the developer community.



Another source pointing to the dwindling interest in VMware's vSphere ESXi is a recent survey[1] of 175 business technology professionals at organizations with 'Virtualized Server Workloads'. The survey shows that only 56% of these organizations were running VMware's vSphere ESXi (or prior versions) in 2014, a stark decrease from an 85% adoption in 2007. It will be interesting to see how these numbers will stack up in the next few years.



Virtualized Workload Platforms

Of your virtualized server workloads, what percentage are running on each of the following platforms?

Note: Mean average percentages

R8181914/21

Data: InformationWeek 2015 Windows Survey of 175 business technology professionals at organizations with virtualized server application workloads, September 2014

Data from recent search interest results and other similar surveys have shown that while VMware ESXi is still the leader, its might has been ably challenged by Hyper-V, KVM and Docker.

[1] http://www.informationweek.com/software/operating-systems/2015-server-os-outlook-cloudy-chance-of-containers/d/d-id/1316553

2.2 Containers vs Virtual Machines

Containers as a technology has created multiple ripples in the virtualization space, with Docker leading the chorus.

So how different are containers from virtual machines?

A container just like a virtual machine (VM) can hold an application, allowing the developer to expend his energies on building the application, rather than writing the underlying system software. Unlike a virtual machine, a container does not include an Operating System (OS), but rather shares the OS of its host — in Docker's case, Linux.

To summarize, containers provide virtualization at the application level without the overhead of running guest operating systems as in the case of virtual machines.

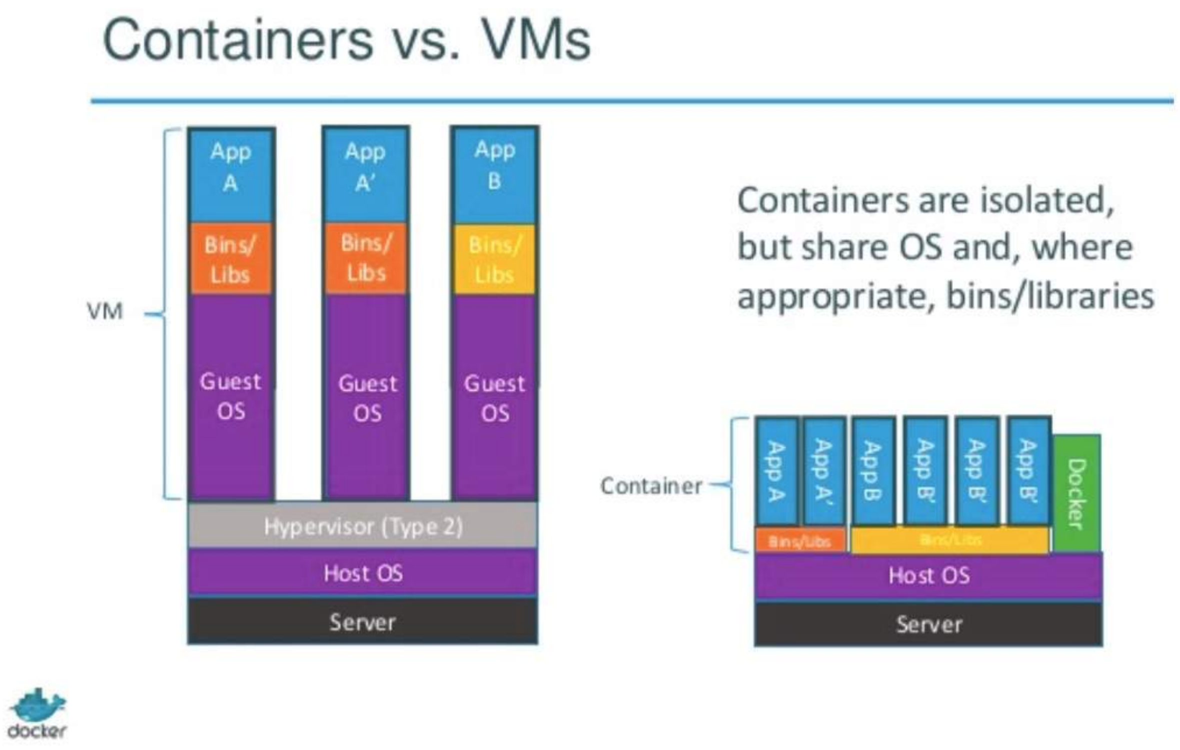


Image source: www.docker.com

Pros and Cons of containers

Pros

- Performance: Containers have equal to or better performance than virtual machines, a comparison validated by an IBM Research experiment[2]. The experiment tested both the virtualization technologies against a suite of workloads that stressed the CPU, memory, storage and networking resources
- Consolidation Density: Containers have considerably higher consolidation density than those of VMs
- Workflow: Docker has been built by developers for developers. A developer centric workflow (Something similar to that of Git) has also been an attractive proposition that containers (Docker in particular) offer [3]
- Less Overhead: Containers have significantly less overhead compared to VMs as they share the operating system of the host and don't have any guest operating systems

An example would be a physical layer with 50 applications. In the case of virtual machines, you would be required to run 50 different guest operating systems. However in the case of containers you would be required to run just 1 OS (that of the host) for all of these 50 applications.

Cons

- Clarity on Usage Paradigm: The usage paradigm of containers is still evolving[4] and has been a topic of contention since its introduction. Some experts opine that a container should be treated like a VM. Other experts have countered this opinion by stating that containers should be treated differently, where in you only run one microservice per container. Historically, there have been multiple services to run each application, but in the case of containers you need to figure out ways of distributing these services across applications.

[2] An Updated Performance Comparison of
virtual machines and Linux containers - Wes Felter, Alexandre Ferreira, Ram Rajamony, Juan Rubio

[3] https://www.docker.com/tryit/

[4] http://phusion.github.io/baseimage-docker/

- Massive Flux: New technologies revolving around containers have been announced almost every week since its inception. An example of flux in this part of the ecosystem can be seen from the recent announcement made by CoreOS. CoreOS, a partner of Docker until recently, announced its own alternative to Docker, called Rocket. Rocket aims to be more open, decoupled and composable than Docker. Google announced its foray in the space with the introduction of Kubernetes, a technology which allows you to orchestrate and schedule applications across containers. Mesos is another technology that has gained traction in the last few years. Developed by the Apache Software Foundation, Mesos provides a different level of abstraction compared to the Linux Kernel while attempting to build distributed systems in a more easy and effective way
- OS compatibility: Unlike VMs, containers have limited OS compatibility and require you to run the latest version of Linux
- Security: In the case of containers, the application can escalate to root privilege on the host, a factor that has led many users to shy away from using containers on the basis of security concerns. It is likely that one rogue application could bring down your host impacting other applications that use the Host OS
- Performance guarantees: In the case of VMs performance guarantees are more predictable and reliable than in the case of containers

3. Comparative Analysis of Different Virtualization Platforms

Platform	Type	Pros	Cons
VMware vSphere (ESXi)	Hypervisor	1. Time tested and stable 2. Familiar Technology 3. OS / HCL support 4. Non disruptive operational workflows (vMotion)	1. Costs can add up, especially for a private cloud environment; vCloud licenses are expensive 2. vSphere does not work on hardware that is not supported (A comprehensive list of supported hardware can be found in the vSphere website)
Linux KVM	Hypervisor	1. Comparatively much cheaper than vSphere (Price can be as low as $0 if you are running Linux KVM with CentOS) 2. Increasingly stable and performant 3. Supports a wide range of operating systems 4. OpenStack as the management platform is beneficial in many ways	1. Windows guest support is not as good as ESXi's support 2. Operational workflows (vMotion) are maturing 3. Deploying and running OpenStack as a management platform can be more complex and would not be the best option for a resource sensitive organization

Platform	Type	Pros	Cons
Microsoft Hyper-V	Hypervisor	1. Cheaper than vSphere (Price can be $0 for Microsoft customers) 2. Increasingly performant. Has been found to perform better than VMware on some metrics 3. Operational Workflows are almost matured	1. Linux does not have the best support 2. Operation Workflows are not as good as vSphere's 3. Perceived as less 'cloud centric' as KVM 4. OpenStack support for Hyper-V is not as advanced as it is for KVM and vSphere
Docker/ Containers	Container runtime	1. Better performance density 2. No overhead from Guest OSes 3. Developer centric workflow	1. Usage paradigm is still evolving 2. Massive Flux around recent developments in the container space 3. OS compatibility issues 4. Security concerns 5. Performance guarantees are not predictable and reliable

4. Strategies for Navigating this Landscape

4.1 Strategy #1: Stay Put!

Technological and other ecosystem developments across containers, in particular Docker, might lead you to question your strategy of using containers. If you used Docker as your container technology, you would also have to grapple with the choice of the best platform to manage it; OpenStack has not been time tested and is not as battle-ready as VMware's offerings.

We believe it would take 2 - 5 years for OpenStack and containers to mature as technologies and address the shortcomings each has.

In this strategy, you would delay your decision of moving to a new virtualization technology and would instead opt for a more matured partner such as VMware for your virtualization needs until clarity emerges on the newer technologies.

Pros

- Perceived reliability of opting for an experienced virtualization technology makes it easier to get organizational approval

Cons

- Possible chaos if other players in the ecosystem start trying other platforms
- Opting for an expensive technology such as VMware would result in lesser savings and a greater budgetary impact. Some of Platform9's customers have experienced significant savings by opting for open source platforms such as CentOS

- A conservative measure such as this would result in you losing time value benefits. An example of this would be the realization that using Docker would have resulted in significantly less overheads and an improvement of overall performance
- Newer technologies such as Docker have created significant interest in the developer community. Delaying a switch to a newer/advanced technology could result in discontent among your own employees (developers in particular), who would like to expand their skill sets by working on new technologies

This strategy could be implemented with the following technologies:

Hypervisor: ESXi/Microsoft Hyper-V
Management System: vSphere / vCloud Director/ vCloud Automation Center

4.2 Strategy #2: Leverage Diversity with Silos

This strategy calls for your organization to embrace the diversity of different platforms using different silos. Silos would allow you to better test the merits of different technologies.

In this strategy, you could continue using your existing virtualization technology such as VMware but build silos with different management platforms. Silos allow organizations to experiment with different virtualization technologies, by keeping each implementation private from the other.

While one of the silos could have any of the VMware owned management platforms such as vSphere or vCloud Director, other possible combinations of VMware and management platforms could be as follows:

VMware + Hyper-V (Suggested for *Windows* users)
VMware + KVM (Suggested for a *Linux* Environment)
VMware + Docker (Suggested if you would like to take advantage of the lightweight and affordability, that *Docker* offers)

Pros

- This strategy can help organizations better understand which implementation might work best by comparing performances across silos
- Implementing different technologies in your own backyard can help raise the skill levels of your team members
- Testing and implementing new technologies in silos can help you influence/own the decision on virtualization technologies in your organization
- This strategy also helps your organization to stay ahead of the curve in leveraging the benefits of new virtualization technologies

Cons

- Managing silos and different workflows would require tapping more organizational resources, resulting in higher costs
- Employing different virtualization technologies in silos would require considerable time commitment from the concerned teams to implement these technologies and from the users to learn about the workflows of each of these management platforms

How can you go diverse?

VMware + KVM:

- ESXi managed by vSphere / vCloud
- KVM managed by OpenStack

VMware + Docker:

- ESXi managed by vSphere / vCloud
- Docker managed by Docker / Kubernetes/ Mesos/ OpenStack

VMware + Hyper-V:

- ESXi managed by vSphere / vCloud
- Hyper-V managed by System Center

4.3 Strategy #3: Leverage Diversity with a Single Pane

Like Strategy 2, Strategy 3 calls for you to embrace the diversity of different virtualization technologies but use a management system that serves as a 'Single Pane' across different pools of virtualization technologies.

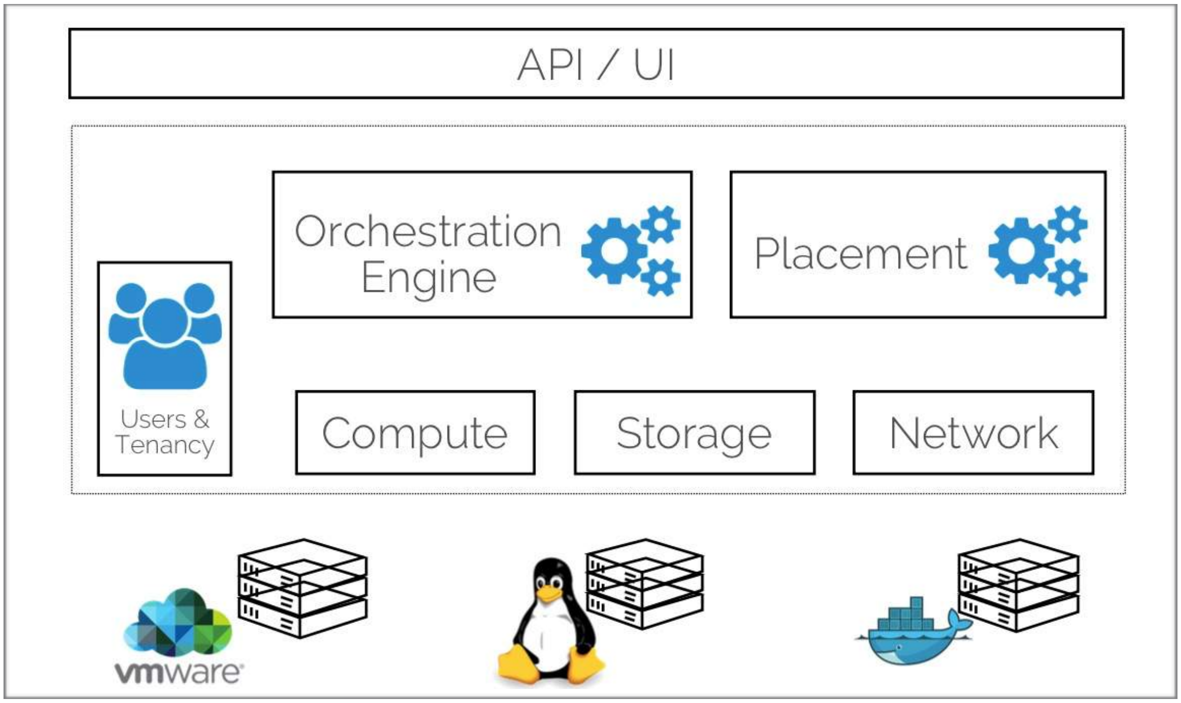
Pros

- Single consistent interface to consume different technologies
- Understand which virtualization technology works best for your organization by battle testing it in real-time

Cons

- A single management platform would not be the best bet if you want to take advantage of every single feature of all virtualization technologies. However, the system administrator can choose to go ahead with a particular virtualization technology for a given use case or user group. For the majority of use cases, we believe both the users and the administrator would benefit if they chose to leverage a single platform

A pictorial representation of this strategy is shown below:



Diverse Virtual Infrastructure

The bottom row represents the diverse pools of virtual infrastructure such as VMware ESXi Servers/Clusters, Linux KVM Hypervisor Clusters and Docker. This strategy entitles the user to take advantage of a virtualized workflow or a Docker workflow.

Multitenancy

The single management platform that you choose to deploy across these infrastructure should be able to offer native support for each of these different pools of infrastructure. The platform should provide you a consistent interface, for integrating existing users and laying out a structure for multitenancy, where you assign different users to different tenants. Multitenancy allows you to set different policies per tenant based on his/her needs and the type of infrastructure he/she uses. For example, the administrator should be able to assign Active Directory, setup XML Authentication Integration or setup Two Factor Authentication using the platform and the users will be able to leverage each of these regardless of the underlying platform.

Compute, Storage and Network

The management platform should allow you to pool together your resources for compute, storage and networking across these underlying diverse virtualization platforms. For example, if you were to leverage Software Defined Networking (SDN), the user should be able to use the SDN constructs across the VMware ESXi Hypervisor, Linux KVM Hypervisor or Docker.

Placement Component

On top of these building blocks, there should be an intelligent placement component, which is aware of the properties/differences of each of these different platforms. The placement component should also be cognizant of the different virtualization/container workflows so that it can help allocate resources for different functions.

Orchestration Engine

A critical component of this entire setup is an orchestration engine — the piece that interacts with both your placement component as well as the underlying building blocks (Compute, Storage and Network). This engine allows you to standardize blueprints/recipes of applications. These recipes can internally consist of one or more VMs or containers, and they should also encapsulate the metadata on how these applications should get deployed. With the Orchestration Engine, you as the administrator can standardize on one or more of these blueprints and check them into your catalog. Your end users can then check out these blueprints and deploy these applications using virtual machines or containers using different hypervisors under the hood. The complexities arising from different formats used by virtual machines vs. containers get abstracted out for you. For containers, you will be able to integrate with the Docker Registry, all of these under the same portal.

User Interface

The final piece of this platform would be a consistent User Interface(UI) that stretches across these components and offers an intuitive experience that allows you to leverage these different functionalities. The UI should serve as an effective self service portal, where the users can sign in and access pre-configured workflows. You as an administrator should be able to automate this infrastructure using its APIs.

5. Recommendation

We would recommend that you employ *Strategy 3,* which calls for you to embrace the diversity but take a strategic point of view while embracing it.

- You should look to test new platforms in silos, but remember that early proof of concepts are key to knowing what strategy would work for your organization
- Your tests should have a well defined success criteria
- Having a fail fast strategy can help avoid higher costs

While you do all of these, look for a single platform that enables you to maintain consistency in managing:

- Multitenancy for different users
- Resource pooling across compute, storage and network
- Provisioning of workflows for different applications

6. Summary

As an organization, you should be aware of the emerging diversity in the virtualization landscape. Testing out different platforms with well defined success criteria can help you navigate this diversity more effectively. Standardizing your workflows using a single management system helps you maintain consistency in managing multitenancy for different users and optimizing your resource pooling across Compute, Storage and Network components.

About Platform9

Platform9 is a private cloud company whose OpenStack based SaaS platform transforms an organization's existing infrastructure into an agile, self-service private cloud within minutes, offering the simplicity of AWS-like public clouds with complete ownership of all resources and data. The platform is the first 100% cloud managed platform for Docker, KVM and VMware vSphere. Founded in 2013 by a team of early VMware engineers, Platform9 is backed by Redpoint Ventures and headquartered in Sunnyvale, CA.

Read more about Platform9 at *www.platform9.com*