TECH GUIDE

OpenStack Deployment Models

PLATFORM9

Introduction

OpenStack's phenomenal growth has made it the industry's de facto private cloud management platform and the second largest open source project of all time. As OpenStack adoption has grown, several OpenStack deployment models have emerged, each with its own pros and cons. This guide provides an overview of OpenStack private cloud architecture, explains some of OpenStack's limitations, and explores several OpenStack deployment alternatives.

OpenStack Architecture Overview

OpenStack's design is inspired by Amazon Web Services (AWS), with well documented REST APIs that enable a self-service, elastic Infrastructure-as-a Service (IaaS) cloud. In addition, OpenStack is fundamentally agnostic to the underlying infrastructure, integrating with various compute, virtualization, network and storage technologies.

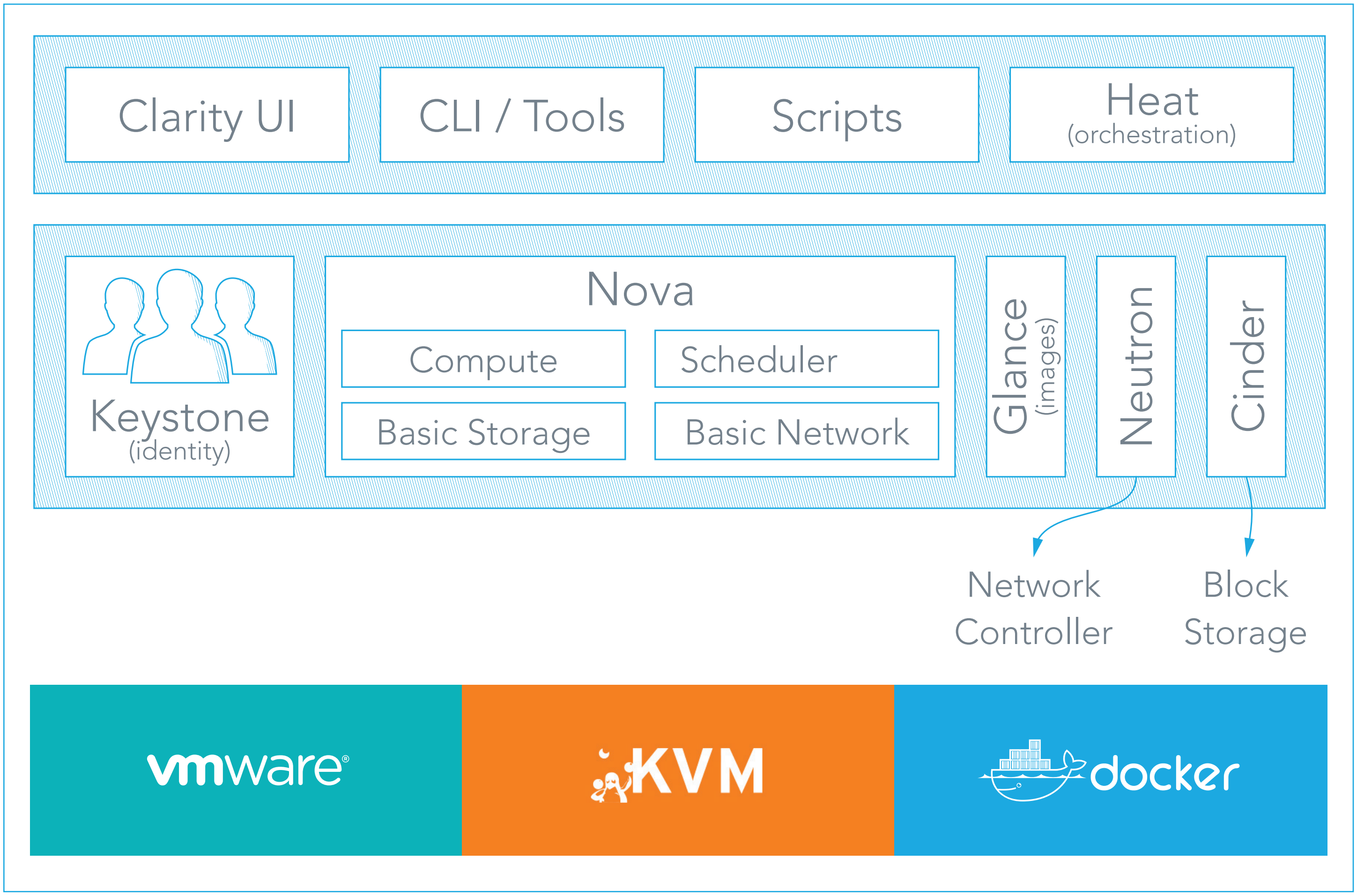
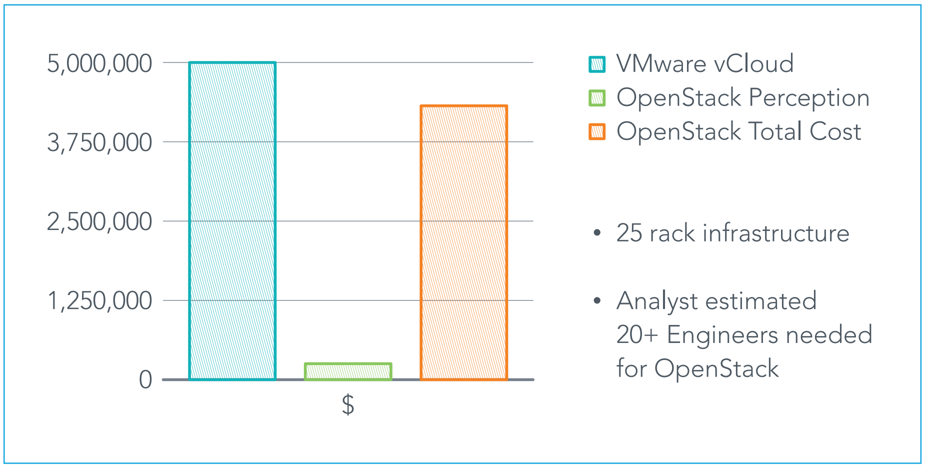


Figure 1: OpenStack architecture is loosely coupled, and extensible to support any hypervisor / container, storage, and network system

Mismatched Expectations with OpenStack

OpenStack has multiple advantages over other private cloud platforms, such as simple REST APIs, an AWS-like service-oriented architecture, and a management platform that works across multiple virtualization technologies. Despite these advantages, OpenStack adoption suffers from the following limitations:



- 25 rack infrastructure
- Analyst estimated 20+ Engineers needed for OpenStack

Figure 2: Perceived cost of deploying OpenStack << Actual cost of deploying OpenStack

OpenStack TCO is Variable

Deploying OpenStack in production is a challenging and resource-intensive exercise. Analysts estimate that a team of 20 engineers would be required to deploy OpenStack for a 25-rack infrastructure.[1] Figure 2 shows that the actual cost of deploying OpenStack is much higher than the perceived cost of deploying the software.

Upgrade Issues

Upgrading from one version of OpenStack to another is a difficult task often plagued with unplanned downtimes arising from unexpected issues. Troubled by past upgrade processes, many organizations are hesitant to migrate to the latest OpenStack release. This is evident in the user sentiment expressed in the OpenStack User Survey, as summarized in this verbatim quote:[2]

"I feel difficulties like deployment of OpenStack at a very large level are still not so easy... the migration of whole infrastructure with a new release of OpenStack is challenging."

> It takes a lot of work to decide on deployment architecture, deploying, and maintaining the software.
>
> OpenStack User, 2016 OpenStack Foundation User Survey

OpenStack Doesn't Work Out of the Box

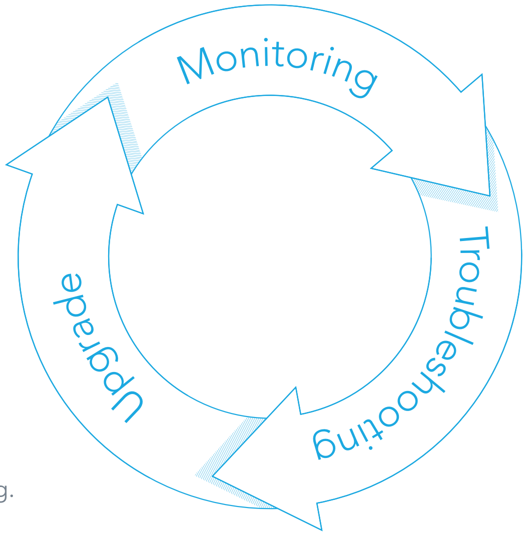
OpenStack is an open source project, not a packaged and production-ready product. This makes deploying OpenStack directly from source code an arduous and engineering-intensive task compared to deploying a packaged product from a distribution vendor or consuming OpenStack as a service. In fact the most recent OpenStack Foundation Survey revealed that "complexity to deploy and operate" presented a challenge for many users.[3]

Deployment Models for OpenStack

There are various ways to deploy OpenStack in your data center, each of which has its own strengths and drawbacks. The following architectural diagrams highlight several of these approaches: on-premises distribution, private cloud in a box, hosted private cloud, and OpenStack-as-a-service.

An important factor in each of these models is the question of who is responsible for the OpenStack lifecycle. Beyond the initial setup of the OpenStack environment, private cloud management includes:

- **Monitoring:** Who will monitor the OpenStack software (control plane and data plane) to ensure that it is healthy and operating as planned.
- **Troubleshooting:** Who will take care of problems - be it an OpenStack bug, a misconfiguration, etc. - to resolve these issues quickly when they arise.
- **Upgrades:** When it's time to upgrade to the next OpenStack release, who will backup the controllers, roll out the upgrade, and validate it prior to committing.



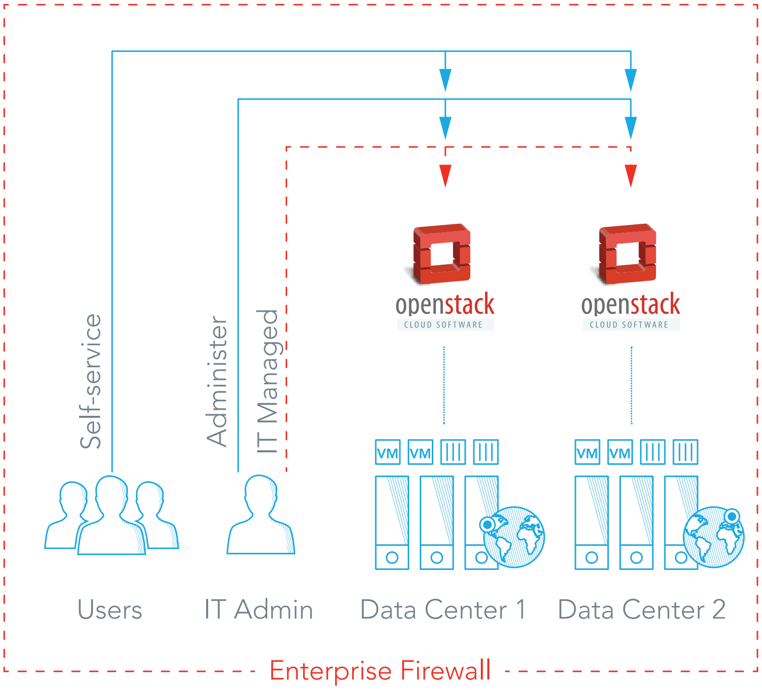


Figure 3: An on-premises OpenStack distribution uses in-house secure infrastructure

On-Premises Distribution

On-premises is the most commonly adopted deployment model, and can be implemented in a do-it-yourself (DIY) manner using Homebrew, or utilizing a particular vendor's OpenStack distribution.

With this model, OpenStack controllers run on-premises, as does the underlying compute, networking and story infrastructure. IT is responsible for installing, configuring, monitoring, and troubleshooting their OpenStack private cloud. In-house personnel customize OpenStack to meet organizational requirements. This deployment model can be cost effective for organizations but only when implemented successfully.

Typical vendors for this model are:
Mirantis, Rackspace, Red Hat.

Who is responsible for the OpenStack lifecycle:
Internal IT team

On-Premises Distribution Pros:

- Security: No workload data leaves your network perimeter.
- Customizable: OpenStack can be customized to suit an organization's requirements.
- Hardware independence: There is no requirement to utilize hardware from a particular vendor.

On-Premises Distribution Cons:

- High operational costs: There are high costs associated with configuring, monitoring, backing up, and upgrading OpenStack, exacerbated by the cost associated with the scarcity of skilled OpenStack personnel available on the market.
- Time to value: Homebrew projects are prone to stall due to technical challenges and lack of skilled staff; implementation with a vendor-supported distribution have the same challenges, but a packaged distribution can help somewhat.
- Creates silos: Having different OpenStack deployments across multiple geographies can create infrastructure silos. It's possible but challenging to avoid with this model, and there is likelihood of creating redundancy and complexity that is difficult to manage.

Private Cloud in a Box

This model of OpenStack deployment involves a vendor-provided hardware appliance that is installed on premises, and ships with a vendor-supported OpenStack distribution tailored to run on that hardware. Your compute, networking and storage infrastructure also runs on premises.

Several hardware vendors have shipped products based on this model, since it increases their margins (and differentiation from commodity, off the shelf hardware) and more customer "stickiness" (difficult to change hardware providers if your cloud platform locks you in).

Typical vendors for this model:
Cisco Metapod, Stratoscale, ZeroStack

Who is responsible for the OpenStack lifecycle:
Internal IT team

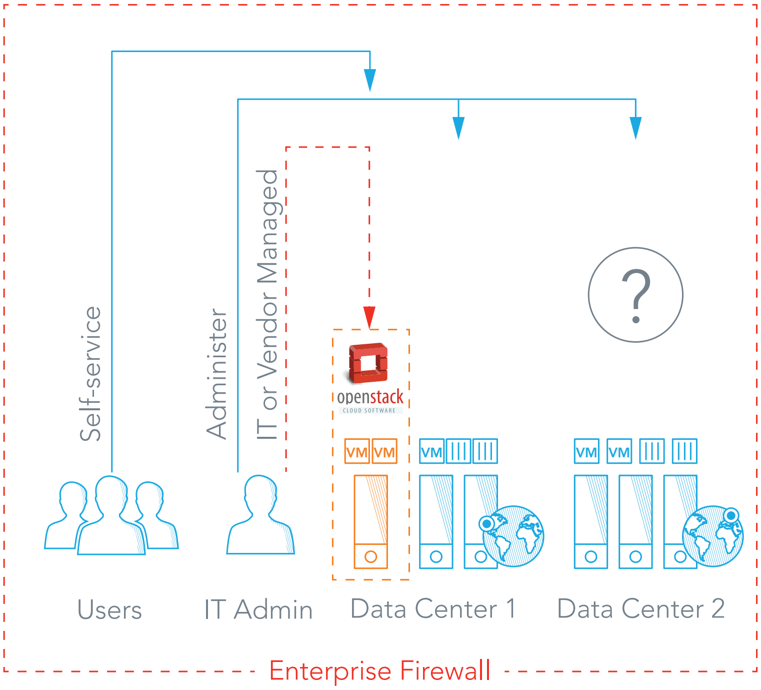


Figure 4: With private cloud in a box, a vendor-provided hardware appliance is installed on premises, tying OpenStack to specific hardware nodes.

Private Cloud in a Box Pros:

- Time to value: Since vendors package hardware and OpenStack into a tightly integrated bundle, this model could offer very quick time to value.
- Potential reduction in operational risk: Operational risk reduction in this model is somewhat variable and vendor dependent. It is possible for this model to substantially reduce the operational risks in running OpenStack, provided the vendor has the right backend automation and support infrastructure.

Private Cloud in a Box Cons:

- Creates silos: Since OpenStack is now tied to specific hardware nodes, this model creates new infrastructure silos and doesn't integrate with existing or other hardware deployments; or other virtualization deployments in the enterprise.
- Hardware dependence: This model is restricted to the vendor-provided hardware, which may be suboptimal for enterprises that prefer a particular hardware provider, or can get preferential pricing on other solutions such as commodity-off-the-shelf-solutions. Often, this hardware dependence creates near- and longer-term pricing disadvantages for customers.
- Vendor lock-in: If the vendor were to go out of business, get acquired or have a change in strategy, customers risk ending up with a substantial investment that may no longer be supportable. For instance, early customers for some first-generation OpenStack providers had to write off their investments following the corporate developments at those vendors.

Hosted Private Cloud

This model of deploying OpenStack involves using a service provider to deploy a hosted private cloud in the service provider's data centers. The service provider is responsible for installing, configuring, monitoring, upgrading and troubleshooting OpenStack as well as the customer's compute, networking and storage infrastructure. This contractual model is usually based on service level agreements (SLAs) and makes it simpler for organizations because they are able to leverage the service provider's data centers, hardware, and OpenStack expertise.

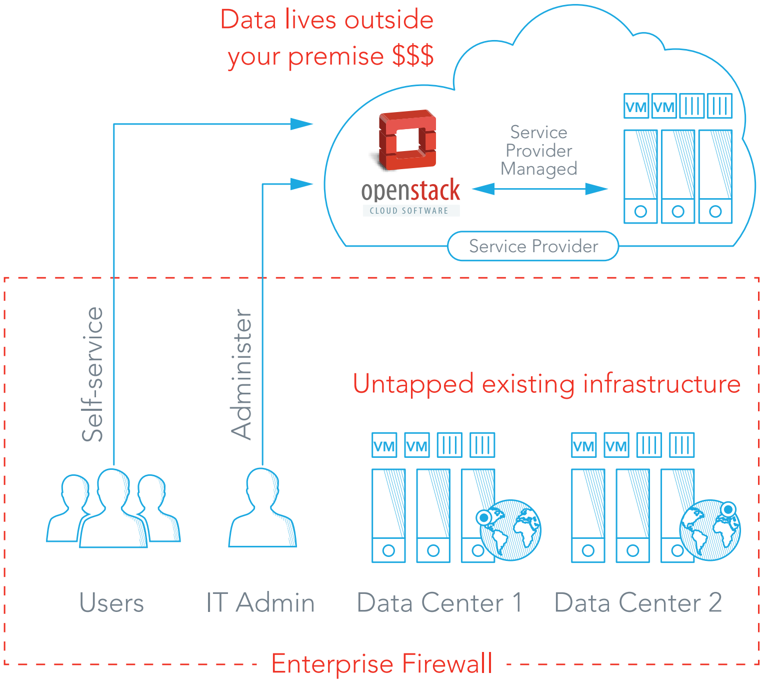


Figure 5: In a hosted private cloud model, data lives outside the organization's premises

Typical vendors for this model are:
IBM Bluebox, Rackspace

Who is responsible for the OpenStack lifecycle: Vendor

Hosted Private Cloud Pros:

- Time to value: Organizations who want a dedicated private cloud without owning the hardware can quickly and conveniently deploy one.
- Operational risk: This model saves organizations from the operational complexity of OpenStack, and vendor-provided SLAs reduce operational risk.

Hosted Private Cloud Cons:

- Hardware dependence: The service provider's choice of hardware dictates the hardware on which your private cloud will run.
- Creates silos: Existing infrastructure might be unused, leading to fragmentation and redundancy.
- Security: Workload data leaves your network perimeter; organizations must depend on the service provider to build a secure and operational OpenStack.
- Lock-in: There is 100% vendor lock-in with this model.

OpenStack-as-a-Service

The OpenStack-as-a-service approach combines the advantages of the on-premises OpenStack distribution and the hosted OpenStack models. Unique to Platform9, the OpenStack-as-a-service model provides the economics, data locality, and infrastructure choice of an on-premises deployment model with the convenience and operational efficiency of hosted private clouds. Organizations that use Platform9 Managed OpenStack host their data on their own infrastructure while Platform9 hosts and manages the OpenStack control plane.

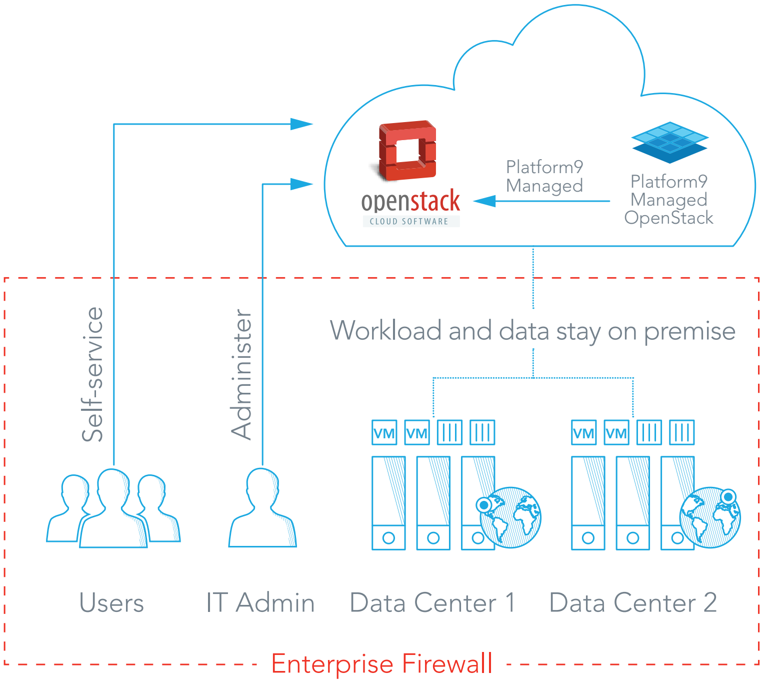


Figure 6: Platform9 manages the OpenStack control plane in the cloud while leveraging and organization's on-premises infrastructure

Typical vendors for this model: Platform9

Who is responsible for the OpenStack lifecycle: Vendor

OpenStack-as-a-Service Pros:

- Set up in minutes: IT team members can build an OpenStack private cloud in minutes.
- Eliminates silos: Platform9 provides a single pane of glass across geographies and different virtualization platforms such as KVM, VMware vSphere, and Docker.
- Secure: No workload data leaves the user's network perimeter.
- Greenfield and brownfield: Works seamlessly with existing or new infrastructure
- Vendor independence: Since customers own the hardware and data, they are not locked into the OpenStack provider.
- Low operational risk: A guaranteed SLA removes operational risk for the OpenStack control plane.

OpenStack-as-a-Service Cons:

- HTTPs access: This model requires outbound, secure HTTPS access from the organization's servers to the Platform9 OpenStack controller.

Comparison Matrix

For easy reference, the different models and their pros and cons are highlighted in the model below.

	OpenStack On-premises DIY Style	OpenStack On-premises Vendor Distro	OpenStack Appliances	OpenStack Hosted Cloud	OpenStack-as-a-Service
Major Players		Red Hat, Rackspace, Mirantis	Cisco Metapod, Stratoscale, ZeroStack	IBM BlueBox, Rackspace	Platform9
Where do OpenStack controllers run?	On-premises	On-premises	On-premises	**Vendor Hosted**	**Vendor Hosted**
Where does infrastructure run? (compute, storage, networks)	**On-premises**	**On-premises**	**On-premises**	Vendor Hosted	**On-premises**
Time to Value	Poor	Variable	Variable	**Good**	**Good**
Who owns OpenStack lifecycle management?	IT (higher risk)	IT (higher risk)	IT or vendor (variable risk)	**Vendor (low risk)**	**Vendor (low risk)**
Hardware Independence	**Good**	**Good**	Poor (Lock-in, Silos)	Poor (Lock-in, Silos)	**Good**
Mixed Environments (existing hardware, multi-hypervisor support)	Challenging, but possible	Challenging, but possible	No	No	**Yes**
Vendor Lock-in?	**No lock-in**	**No lock-in**	Lock-in	Lock-in	**No lock-in**

Key Takeaways

OpenStack is very powerful private cloud technology, but the complexity of deploying and operating OpenStack deters some organizations from taking advantage of its power as a private cloud solution. There are various deployment models for OpenStack, that were discussed in this guide. As you consider your deployment options, ensure that you are selecting a model that removes complexity and risk while lowering operating expenses associated with headcount. An OpenStack-as-a-service model such as Platform9 Managed OpenStack helps organizations realize significant cost and time savings when deploying their private cloud.

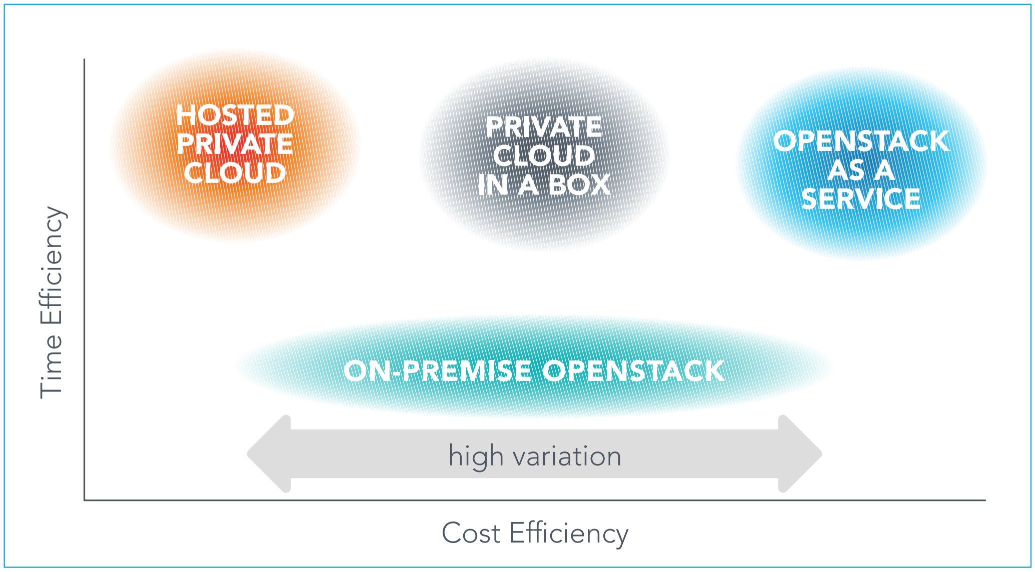


Figure 7: Platform9's deployment approach saves money and time for organizations deploying an OpenStack private cloud.

About Platform9

Platform9 transforms an organization's existing servers into an AWS-like agile and efficient self-service private cloud at any scale within minutes, while leveraging the latest open source innovations. Powered by OpenStack, Platform9 is the first 100% cloud-managed platform for KVM, VMware vSphere, and Docker. Founded in 2013 by a team of early VMware engineers, Platform9 is situated in Silicon Valley. Platform9 is backed by leading Venture investors, and has dozens of customers in production including Box and PubMatic. For more information on Platform9 Managed OpenStack, please contact us at customer-success@platform9.com or visit www.platform9.com.

Footnotes

1. http://blogs.barrons.com/techtraderdaily/2014/04/03/vmware-price-tag-versus-openstack-immaterial-says-isi/?mod=yahoobarrons&ru=yahoo
2. http://www.openstack.org/assets/survey/April-2016-User-Survey-Report.pdf page 13
3. http://www.openstack.org/assets/survey/April-2016-User-Survey-Report.pdf page 12