MICROLAND®

Extraordinary. Everyday.

OPENSTACK DESIGN: BUILDING A ROBUST ARCHITECTURE FROM THE GROUND UP

SOLUTION PAPER

Abstract

Cloud offers wide-ranging benefits such as elasticity, flexibility, and efficiency. But the benefits come at a price – that of limited visibility and control. To bypass this roadblock, most organizations take the private cloud route - by leveraging OpenStack to build a private cloud infrastructure that supports enterprise applications as well as integrates with the existing datacenter infrastructure. A recent joint OpenStack survey shows that OpenStack adoption is increasing across the world as more and more enterprises move to a multi-cloud environment, supported by edge computing[1].

However, one common mistake most organizations make while building an OpenStack environment is that they start the exercise with a budget in mind. While budget is an important consideration, cloud investments are long-term in nature and should satisfy the current needs of developers and the business, while offering the flexibility to scale dynamically with future requirements. This makes OpenStack design a critical aspect of the overall deployment. Preliminary design should be developed based on what end users and business goals demand, while budget, resource, and other practical limitations can be added subsequently, as required, to refine it further.

This paper puts forth a robust approach for designing a flexible and scalable OpenStack environment that supports end-user requirements and is aligned with business goals.

BusinessWire, OpenStack adoption, deployments surge as enterprises increasingly favor multi-cloud strategies, **http://www.businesswire.com/news/home/20171105005088/en/OpenStack-Adoption-Deployments-Surge-Enterprises-Increasingly-Favor**

OpenStack popularity continues to grow

OpenStack, since its inception in 2010, has evolved from a cloud orchestration layer to a full-fledged cloud operating system. Given its unparalleled capabilities and strong support community, OpenStack is rapidly becoming a highly favored system, which could be a double-edged sword if inefficiently designed and deployed. Designed well, it can provide a powerful hyper scale, multi-featured, multi-tenanted ServiceDown cloud. On the flip side, poor OpenStack design can lead to undesirable outcomes. But it must be remembered that there is no one-size-fits-all approach for an OpenStack cloud deployment as the software can be used to support varied purposes, each with its own architectural requirements.

Enhancing developer team agility: An OpenStack use case

Let's begin with an illustrative business problem that we will try to address by implementing an OpenStack environment and choosing from the different options available. The goal of this exercise is to recommend an OpenStack deployment. To this end, we will need a sample requirement matrix that highlights the use cases for the cloud deployment in question, and the corresponding features that need to be enabled for the cloud to perform as per business requirements.

Problem statement

Assume that the business is grappling with the following problem.

Our competitors have started gaining on us in terms of agility by deploying cloud. To counter this and grow market share, we need to improve the agility of our development teams, by providing them with a self-service and programmable cloud infrastructure. However, since we have already heavily invested in upgrading the in-house infrastructure and would like to ensure stringent compliance, we would prefer to keep the cloud deployment in-house.

From the above problem statement, a few things can be ascertained:

- Type of cloud (based on location considerations): **Private** - As the customer wants the infrastructure to be in-house - for cost, compliance, as well as control purposes.
- Type of cloud (based on technical considerations): **ServiceDown** - Since the customer needs programmable infrastructure i.e. Infrastructure-as-a-Code, which is not a prominent offering of the Infrastructure Up clouds (for instance, VMware vRealize Suite based cloud).
- Type of cloud (based on financial considerations): **Private** - As customer has invested heavily in their on-premise infrastructure and moving everything to a public cloud is not likely to make economic sense.
- Cloud usage: **Dev-test cloud** - As developers and/or DevOps engineers are going to be the primary users of the cloud in their attempt to become more agile in their development efforts.

Based on the above, it is easy to zero in on OpenStack as the possible solution as it is the only ServiceDown orchestrator available in the market. Every other private cloud orchestrator is either Infrastructure-up or a hybrid - for instance, Open Nebula. At this point, we need to ask a few targeted questions to pinpoint which of the 19 OpenStack projects the business would need, what would be the mode of deployment and network configuration, and so on and so forth.

Let's take a look at a non-exhaustive list of questions for the business that can help with the formulation of the right solution.

FOLLOW UP QUESTIONS

Ideally, the follow-up questions (see Table 1) should be rated on a scale of 1- 10. This allows the answers to be grouped into categories based on their criticality such as – 'must have', 'good to have', or 'can be ignored'. It is best that this exercise be carried out with both the end customer and the supporting IT team as some questions might have an impact on the ongoing support of the system.

Topic	Question	Rating (1-10) 10-Strongly Agree 1-Strongly Disagree
Usage	Is the system not going to host any production workloads?	
	Will the system be used by diverse groups of users?	
	Will the system be an isolated environment with very little communication with the outside world?	
Authentication	Should the authentication to the OpenStack system be the same as the organizational authentication process?	
Updates	Is it okay to update the system frequently with bug fixes and new features?	
Capability	Does the system need to provision to the bare metal servers?	
	Does the system need to provide Databases-as-a-Service (DBaas)?	
	Is the system expected to provide Containers-as-a-Service?	
	Is the system expected to provide Object Storage as a Service?	
	Will no persistent data be stored in the system?	
	Will custom images be created by the development teams?	
	Is the system expected to be scaled up to 10000's of physical servers?	
	Is the system expected to provide Load Balancer-as-a-Service (LBaaS)?	
	Is the system expected to provide VPN-as-a-Service (VPNaaS)?	
	Is the system expected to provide Firewall-as-a-Service (FWaaS)?	
	Is the system expected to provide billing data?	
	Is the system expected to provide Hadoop (Big Data)-as-a-service?	

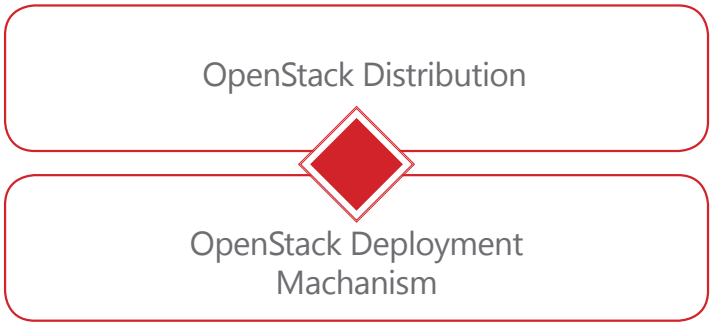
	Is the system expected to provide the DNS-as-a-Service?	
Support	Is the management system expected to have a 99.99% uptime?	
	Does the organization dictate that support needs to be purchased for the system?	

Table 1: Follow-up questions that determine OpenStack architecture

The answers to the next set of questions should be quantifiable to help determine the size of the deployment. These include questions such as:

- How many users are expected to use the system?
- How many virtual servers are expected to be running on the system?
- What will be the access mechanism to the system?
- What is the current inventory of network, compute, and storage systems?

Once these questions are answered, the following design decisions need to be taken.



CHOOSING THE RIGHT DISTRIBUTION AND DEPLOYMENT OPTION

OpenStack distributions vary ever so slightly, but the differences can alter the impact in terms of supporting different hypervisors. Table 2 assists in selecting the Distribution - based on the support needed for different hypervisors and the Guest Operating systems. Please note that this is a non-exhaustive list that has not been updated with the latest release.

Distribution Name	Hypervisor Support	OS Support
Ubuntu OpenStack	Hyper-V, QEMU, KVM, ESXi, LXC, LxD	Linux, Windows
Red Hat OpenStack	ESXi, KVM	Linux, Windows
VMWare Integrated OpenStack	ESXi	Linux, Windows
Mirantis OpenStack	Xen, Docker, Hyper-V, ESXi, LXC, QEMU, KVM	Linux, Windows
Oracle OpenStack for Solaris	Solaris	Solaris
Oracle OpenStack	Xen, KVM	Linux, Windows, Solaris
Cisco OpenStack	QEMU, KVM	Linux, Windows
IBM Cloud Manager	z/VM, PowerVM, ESXi, Hyper-V, KVM	Linux, Windows
Suse Cloud	Xen, Hyper-V, ESXi, KVM	Linux, Windows
Dell Red Hat Cloud	KVM	Linux, Windows
HP Helion OpenStack	ESXi, KVM	Linux, Windows

Table 2: OpenStack distribution options

Once the right OpenStack has been chosen, the deployment method can be easily finalized, depending on the services that need to be installed. For example, some of the most used deployment tools are:

- Red Hat OpenStack: RDO Deployment, OpenStack Ansible (OSA)
- Ubunutu: Metal as a Service (MaaS) and Juju
- VIO: VMware vSphere
- Mirantis OpenStack: Fuel

Businesses also have the option of simply building the OpenStack from the source and installing it. Lately, the OpenStack Control plane needs to be installed on containers like Docker (with or without Kubernetes). In order to do so, one could use Stackanetes or Kolla to deploy the OpenStack environment.

Each of the deployment methods has its pros and cons. Choosing the right method is therefore critical to implementation success as well as superior business outcomes.

IDENTIFYING THE BEST-SUITED NETWORK

The network is the next important aspect for any cloud solution. Depending on the services (VPNaaS, FWaaS, LBaaS) and the types of interfaces needed with hardware network devices, one or more neutron plugins may be used and configured.

In case, cloud is an extension and only connects to enterprise network with no tenant networking (an extension of a flat network), a simple Linux bridge is recommended. Table 3 provides a comparative view of the features of the Linux Bridge and Open vSwitch.

Feature	Linux Bridge	Open vSwitch
Layer 2	Yes	Yes
Layer 3 Routing	Yes	Yes
Layer 3 HA	Yes	Yes
Layer 3 DVR	No	Yes
DPDK	No	Yes
Floating IP	Yes	Yes
LBaaS	Yes	Yes
VPNaaS	Yes	Yes
Isolated Tenant Networking	Yes	Yes

VLAN Overlays	Yes	Yes
VXLAN Overlays	No	Yes
GRE Overlays	Yes	Yes

Table 3: Feature comparison between Linux Bridge and Open vSwitch

So, unless Distributed Virtual Routing (DVR) is needed or Data Plane Development Kit (DPDK) needs to be enabled, one could simply choose Linux Bridge as the default virtual networking device.

If overlays are not being used or external network devices/SDN controllers need to be configured, then plugins from different vendors such as Midokura, Open Contrail, Cisco, and others can be used.

ENSURING OPTIMAL STORAGE

Choosing the storage connection is the next step in the process. While Cinder is used for creating volumes, it supports several back-ends right from EMC's VNX to Linux LVM Storage. If storage boxes are already present in the environment, which is more likely than not, Cinder will be able to orchestrate the create, read, update, and delete (CRUD) operations of the storage box. This link can be used to check if the storage is supported by the Cinder backend driver:

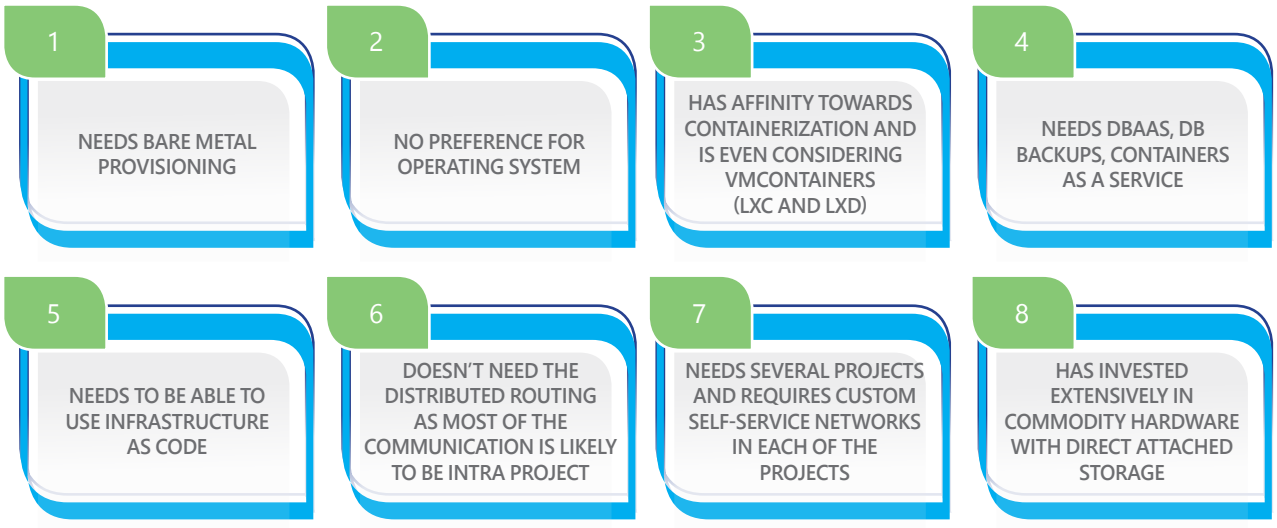
https://wiki.openstack.org/wiki/CinderSupportMatrix

DESIGNING THE COMPUTE

The best thing about OpenStack is that it will work with most x86_64 servers running on Linux. So, in most cases, the hardware is not a problem as long as it can run a supported version of Linux and KVM. It is also to be noted that different hypervisors may be supported either in different versions of the OpenStack or different distributions – a key aspect to be considered while designing the compute.

AN ILLUSTRATIVE EXAMPLE: SAMPLE DESIGN

For the purpose of illustration, we assumed certain typical requirements and answered the above questions, resulting in the following summary regarding the customers' needs and preferences:



With the help of the above summary, it is possible to start an effective design process.
Some non-exhaustive choices could include:

- Distribution: Ubuntu OpenStack
- Deployment Method: MaaS + Juju
- OpenStack Projects: Keystone, Nova, Glance, Swift, Cinder, Trove, Heat,Horizon, Magnum, Ironic
  - Trove (DBaaS) needs Nova, Glance, Swift and Cinder to function
  - Bare Metal requires Ironic
  - Magnum is necessary for Container Orchestration
- Virtual Switch: Linux Bridge
- Network Mode: Type -2 (With Tenant Networking)
- Storage Backend: Ceph
  - With DAS in place, the best option would be to run Ceph on top of that, and run Cinder to orchestrate the Ceph

While OpenStack deployment needs deeper thought and attention to detail, the steps discussed in this paper can help businesses get a head start.

OpenStack: The future of cloud computing

Private cloud adoption among enterprises is at an all-time high – it increased to 77% in 2016, in turn, driving hybrid cloud uptake from 58% to 71% "YoY". In the seven years since its inception, OpenStack has become the dominant private cloud platform globally. As cloud matures and its deployment barriers - most prominently security and cost - become less of a concern for enterprises, OpenStack software offers a compelling value proposition. This includes improved security, agility, compliance, and efficiency with greater control over the hardware.

Deriving the maximum value from OpenStack deployment however, is a matter of making the right choices right from the planning and design stage. A bottom-up approach that focuses on gathering end user requirements and developing a preliminary design that can be fine-tuned later, works better than a top-down approach that could be limiting. Capabilities, not constraints should be the emphasis for organizations looking to OpenStack and and leveraging the expertise of an experienced technology partner can help ensure the right design and direction from the start.

"SDX Central, Enterprises Increase Private Cloud Adoption to 77%,
https://www.sdxcentral.com/articles/news/enterprises-increase-private-cloud-adoption-by-14/2016/02/"

About the author

Alok Shrivastwa
Director – Special Projects

Alok Shrivastwa is the Director, Special Projects at Microland Ltd. He heads the design and implementation of several cloud management platforms and machine learning-based tools. With rich experience in design and implementation of robust solutions across network security, automation, databases, VoIP environments, datacenter designs, and public and private clouds, Alok helps build tools and intellectual properties for operationalizing emerging technologies. He has also authored two books - Learning OpenStack and OpenStack Trove Essentials - along with several whitepapers and blogs on technology and metaphysical topics.

For further information
Contact us at: **+1 646-254-3598** or Email us at : **cloud@microland.com**

About Microland

Microland accelerates the digital transformation journey for global enterprises enabling them to deliver high-value business outcomes and superior customer experience. Headquartered in Bangalore, India, Microland has more than 3,800 professionals across its offices in Australia, Europe, India, Middle East and North America. Microland partners with global enterprises to help them become more agile and innovative by integrating emerging technologies and applying automation, analytics and predictive intelligence to business processes. For more info: http://www.microland.com.

Learn more about us at:

www.microland.com

MICROLAND®