OpenStack Cloud Platform

Valentin Poirot

I. Introduction

For the past ten years, cloud system has become an emerging technology. While all cloud service providers offer a reduction of cost for companies and a access to their data everywhere, numerous different technologies are available on the market. OpenStack is one of them but, instead of reposing on proprietary solutions, it is completely free and open source.
We can ask ourselves what are the specificities of this solution. First, we will retrace its history, from 2010 to its current version, and enumerate its principal characteristics. Then, we will take a more technical look at its composition and its architecture.

II. the OpenStack cloud platform

1. The history of the project

In March 2010, Rackspace started working on a new file system technology. They wanted to create a new open source cloud platform. The NASA, for their own work, were developing Nebula, a cloud computing platform for federal agencies in the US. In July 2010, they decided to collaborate and merge their two projects to create *OpenStack*, an open-source cloud-software initiative. The first version, named Austin, was published the 21st of October of the same year. It was composed of Nova, from NASA's Nebula project, and Swift from Rackspace. 4 months later, the second version, Bexar, is published, the 3rd of February, 2011. The Glance module joined the two precedents.
Rapidly, the community and companies start taking interest in the project and collaborate. In 2011, Ubuntu community adopts OpenStack. In 2012, Debian and Red Hat incorporate OpenStack. The same year, the OpenStack Foundation is founded. This non-profit organization will take care of the governance and future of the project.
Big companies also starts joining the development. In 2013, Oracle became a sponsor and unveiled Oracle OpenStack distribution for Solaris. In 2014, HP unveiled HP Helion, using OpenStack. Soon, VMWare, IBM, Cisco and others formalized their participation as well.
The current version, Mitaka, was launched the 7th of April 2016.

2. Principal characteristics

OpenStack is defined as "a cloud operating system that controls large pools of compute, storage and networking resources". That means OpenStack will work as an abstraction layer above the hardware, and will manage the integrality of resources available.
As such, OpenStack is considered as an Infrastructure as a Service (IaaS). Contrary to other technologies or cloud providers, the user has the control from the application layer until the (virtual) servers. It can create new virtual machines, databases or storage cluster by himself

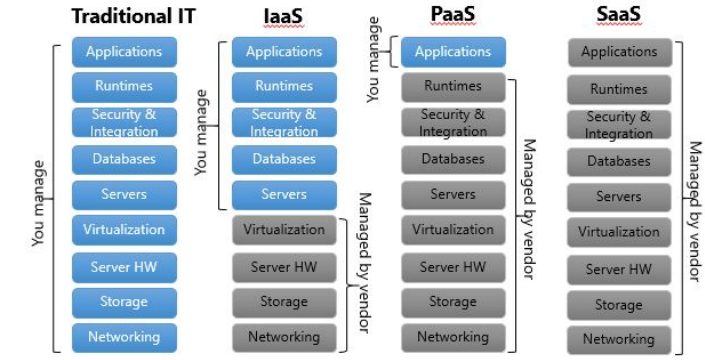

Figure 1 - … as a service comparison

The project aim to make cloud easy and scalable. Adding or removing server or storage should be intuitive and the system should support any amount of machines. Also, every machines or operating system should work. Even if OpenStack has been historically developed with KMV, it can work with other supervisors such as VMWare ESXi, Hyper-V and others.

OpenStack is open source, which means everyone can contribute, and is developed in Python. It can be controlled via the console (directly on the machines or via ssh) but also provides a RESTful API, offering a way to manage the cloud with a simple web interface.

It was designed with a modular architecture, in such a way that every process is clearly separated from the others and adding new functionality is easy.

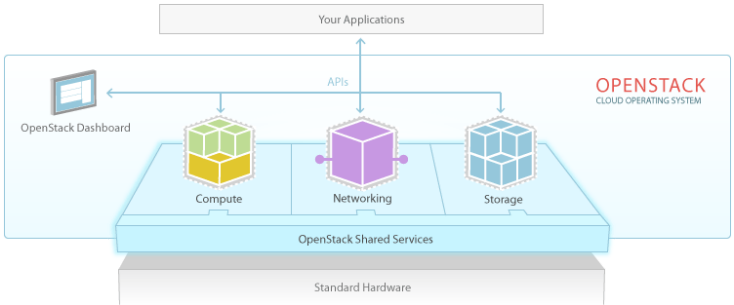

Figure 2 - OpenStack basic infrastructure

III. OpenStack components and architecture

1. Modularity of OpenStack

OpenStack was designed as a combination of modules working together. With such a design, the different functions and processes are clearly separated and it is easier to personalise your own instance of OpenStack with personalised modules. However, some modules are recognised as main parts of OpenStack.

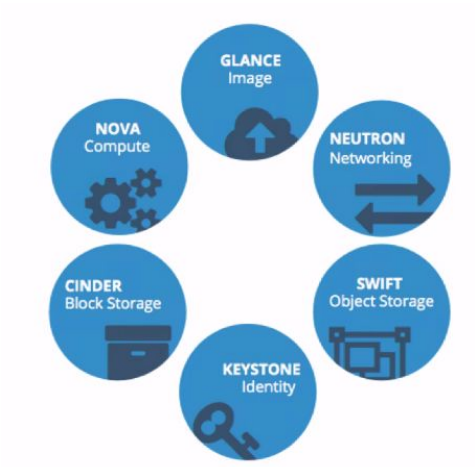

Figure 3 - OpenStack primary modules

a. Nova

Nova is the computing module. It manages pool of computing resources. The user can control and communicate to the instances and hypervisors through the API. It is scalable and on demand, which means the computing resources can be increased or decreased depending on the current needs.

b. Cinder

Cinder is the block storage module for OpenStack. A block storage works by defining blocks with a maximal length and containing a collection of records. It can be used to store virtual machines, since the format is more adapted than object storage. Cinder storage blocks will appear as iSCSI volumes for the user.

c. Swift

Swift is the object storage module. It is the most simple and efficient way of storing data with OpenStack. It is reliable by taking care of redundancy without needing RAID technology. By default, Swift will create three instances on different physical locations when a new object to be stored is created.

d. Glance, Neutron, Keystone & Horizon

Glance module manages images. It is used for the discovery, registration and management of disks and server images. It can also be used for backup by creating images of machines.
Neutron is the networking module of OpenStack. It can be used as a DHCP server to provide control and provide pools of dynamic IPs, but can also provide static IPs. It is also compatible with the Software Defined Network paradigm.
Keystone is a central authentication system that can be used for the entire cloud. It also supports LDAP.
Finally, Horizon provides a simple web user interface to manage your instance of OpenStack.

e. Secondary modules

In addition to these primary modules, the community offers a variety of other modules, called secondary modules. Although all of their development may not be mature enough, they can

be easily tested or use in a production installation. We can, for example, cite Ceilometer for telemetry, Congress for governance, Murano for an application catalog, Trove for databases, Manila for shared filesystems, Heat for orchestration, and many more...

2. An overview of its architecture

From its modularity, OpenStack is highly scalable. Adding new computing power will not impact the storage module or images. Moreover, due to its high personalisation, OpenStack modules must have a loose-coupling between them. With such architecture, there is not hard dependencies between modules.

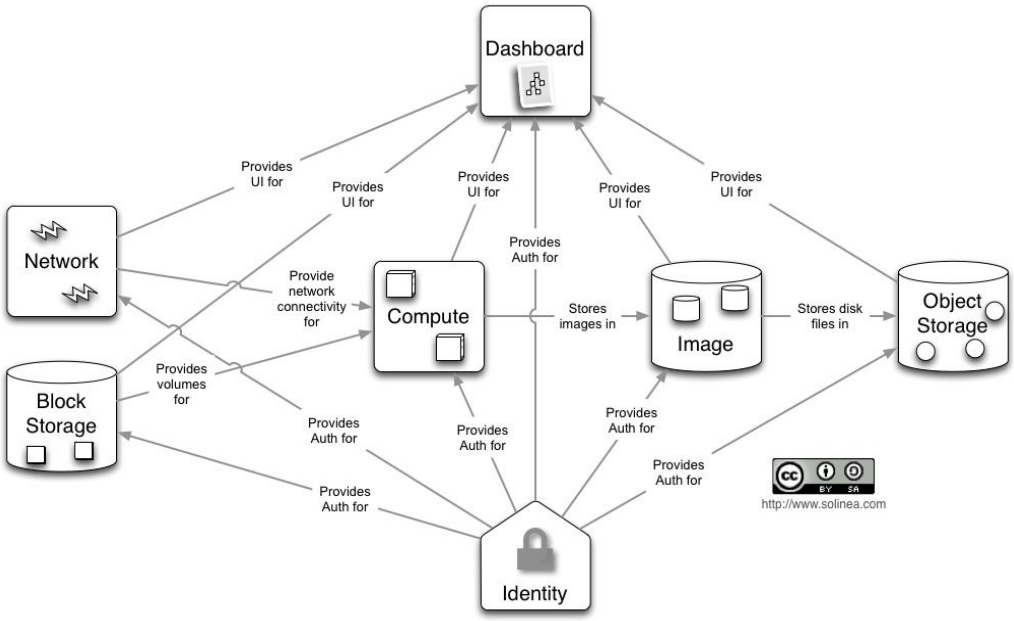

Figure 4 - OpenStack architecture

IV. Conclusion

In conclusion, since OpenStack is open-source, the absence of vendor lock-in allow combination of heterogeneous devices from different companies in the same cloud. In addition, migration between two OpenStack instances will be easier (more compatibility) than between two different instances. OpenStack makes cloud management easy and data migration between providers easier. OpenStack even provides plugin to migrate data from Amazon cloud to an OpenStack instance.
Moreover, the industry is highly involved in the creation and the development of the project, such as Red Hat, IBM, Oracle, Dell, HP, Cisco, etc.
We can then conclude that OpenStack is a huge collaborative work between companies and individuals that provide a free, open source and easy way to manage your own Infrastructure as a Service cloud.

V. References

- OpenStack, http://www.openstack.org

- OpenStack presentation, Hitesh Wadekar, http://fr.slideshare.net/openstackindia/openstack-introduction-14761434
- Nova Developer Documentation, http://docs.openstack.org/developer/nova/
- Swift Developer Documentation, http://docs.openstack.org/developer/swift/
- Chris Evans, OpenStack storage explained, http://www.computerweekly.com/feature/OpenStack-storage-Cinder-and-Swift-explained