iCloudSpace: an Open Source Private Cloud Solution for College Campus

Barkha Pundir[1]
M. Tech. (MPC)
CSE Department
IGDTUW, Delhi
barkhapundir18@gmail.com

Monika[1]
M. Tech. (MPC)
CSE Department
IGDTUW, Delhi
monikamor04@gmail.com

Monika Choudhary[2]
Assistant Professor
CSE Department
IGDTUW, Delhi
monika.ch13@gmail.com

Dr. S.R.N. Reddy[2]
Associate Professor
CSE Department
IGDTUW, Delhi
rammallik@yahoo.com

Abstract—Storage, compute and network provisioning are the most necessary resources for computational purposes in academics. Further, availing resources easily and at a low cost is desired especially for students. In this paper, an open source cloud solution is proposed for use at the college campus. A private cloud deployment on college premise using OpenStack 3-node cloud architecture is discussed. Once the cloud is successfully deployed, users can provision storage, network, and compute resources after successful authentication. Both web interface and command line clients can be used to interact with the cloud. PaaS and SaaS can also be developed on top of OpenStack IaaS. As this cloud solution is open source, it can be used by different institutions for meeting their needs.

Keywords—Cloud computing, open-source, OpenStack, private cloud, IaaS.

I. INTRODUCTION

In the last few years, the field of cloud is gaining popularity and most organizations are adopting the technology of cloud computing for their businesses. Cloud computing has risen as a solution to fulfill the requirements for heavy computational tasks without acquiring expensive hardware and software to do the service. Cloud computing provides everything whether it is computing power or computing infrastructure. Applications, business processes to personal collaboration, all can be delivered as a service whenever required.

Cloud computing is enabled by two main technologies—virtualization and networking. Virtualization improves the server utilization by allowing several applications and operating systems to run on a single machine thus providing scalability and sharing of resources. Improving server utilization leads lesser servers needed in data center and decrease in generated carbon footprint. The networking in a cloud environment is very crucial as it plays an important role in data tunneling and providing user access to cloud resources.

A number of companies have been providing cloud platforms for various purposes. Amazon (Amazon EC2, S3), Microsoft (Windows Azure) and Google (Google App Engine) are amongst the top public cloud providers. These platforms provide closed source solutions for IaaS and PaaS. On the other hand, there are open source cloud solutions like OpenStack, Eucalyptus, OpenNebula which allow developers to use the source code and develop cloud platforms and applications. In academics, both closed source and open source cloud solutions have been used. Some institutions use closed source solutions of public clouds to lower their infrastructure and operating costs while big institutions deploy their own private clouds using open source platforms.

A cloud solution is discussed in this paper for accessing resources anytime, and across devices and networks. An open source cloud can be deployed to provide a low cost solution. Open source means open APIs, 100% open-source free of vendor lock-in and a solution for maximum flexibility and choice. This model of cloud computing is gaining momentum because of the fact that servers installed in most data centers are not operated at their full capacity which leads to less than full utilization of their computing capability. These resources can be used efficiently through virtualization and thus enables greater returns on data center investments. The available open source platforms for cloud make it possible for an organization to create its own private cloud utilizing its own infrastructure.

II. LITERATURE SURVEY

This section provides a review of work done on various open source and closed source cloud computing platforms. The literature review helps in determining the best platform according to our application and development environment.

The authors of [9] have discussed the challenges that may be faced by large, medium and small enterprises in the adoption of cloud services. A comparison of open source cloud platforms OpenStack and OpenNebula has been done in [10] to help in the selection of the best platform for any organization. The parameters for comparison are provenance, architecture, security, hypervisors, etc. In [11], a detailed comparison of open source and closed source cloud platforms has been presented focused on the aspects such as the architecture, characteristics, application and so on. To show the differences that exist...
between open source and close source approaches in cloud environment, the authors have mentioned some examples for Platform-as-a-Service, Software-as-a-Service and Infrastructure-as-a-Service.

In academia, cloud computing can prove to be a flexible tool for data access. Choosing a public cloud can lead to problems like data security and privacy, and vendor lock-in. On the other hand, a private cloud solution is suitable for having complete control over the data and an open source solution saves from vendor lock-in issues.

There are a number of open source cloud solutions available for deploying IaaS. OpenStack has been chosen as the cloud computing platform because it is a completely open source, modular (composed of a number of different projects) and widely supported platform. It has great community support and good documentation for installation and debugging errors. A lot of SDKs are also provided by OpenStack to create applications on top of private cloud. OpenStack is used by a lot of companies like IBM, Rackspace, RedHat, and many more to provide private cloud solutions.

Table 1 lists the differences between various open source cloud computing platforms based on the review done of the papers [7-15].

Apart from selection of cloud platform, the cloud team needs to select the appropriate hardware (servers, switches and routers), and software (hypervisor, operating system, etc.).

### III. PROPOSED CLOUD ARCHITECTURE

In this paper, a cloud solution is proposed in accordance with the college campus computing needs. In the proposed cloud deployment, services can be provided to students and faculty for storing data and developing applications. To configure the private cloud IaaS, OpenStack is used. OpenStack provides an Infrastructure-as-a-Service model with a variety of other services where each service has an application programming interface (API) that is used to support the integration. It provides all the compute, storage and network resources. PaaS and SaaS solutions can then be developed on the top of IaaS. The proposed cloud architecture is as shown in Fig. 1.
Different cloud services will be offered to university faculty and students.

i. Teachers can upload students’ attendance and marks using the online portal.

ii. Assignments can be given to students online and practice tests can be uploaded for students.

iii. Students doing research can upload their data to the cloud for data storage. PaaS layer can be used to develop applications online.

A suitable open source platform (PaaS solutions such as Cloud Foundry or OpenShift) can be configured on top of IaaS on which applications can be built and run. Joomla and Drupal are open source frameworks for developing SaaS applications.

The flow of deploying the proposed cloud is shown in figure 2 below.

For implementing IaaS, OpenStack is chosen as the platform. It is an open source cloud IaaS solution which is a collection of various projects to provide different services. In the current work scenario, certain services that have been configured include keystone, glance, nova, neutron, horizon, cinder, and ceilometer. Figure 3 below shows the interaction of the various OpenStack services.

Nova component or compute Service is responsible for the life cycle of instances within the OpenStack cloud and supports them. It is also used to manage and the computer resources which can be used to work with different virtualization technologies or hypervisors.

Glance component or OpenStack Imaging Service is responsible for the cloud disk and server images along with providing lookup and retrieval system for the virtual machine images. Keystone component or Identity Service is responsible for the authentication across the cloud infrastructure. It provides the authorization and authentication services for all components in either username/password format or token based. Horizon component or Administrative Web Interface is used to manage or administer all OpenStack services. It is also used to manage the instances and Images, creating new key-pairs, defining security groups, uploading images, console, manipulating the volumes to instances etc. Swift component or Storage Service provides virtual object store which is distributed. It has great capability to store large size files or large number of files among servers in an organized way along with redundancy and failover management.

Three-node architecture of OpenStack Juno version is deployed at the university. The nodes are configured as controller node, network node and compute node. Different services are installed on the three nodes as per the Juno installation document [1]. The systems are pre-installed with Ubuntu 14.04 LTS.

For the networking between the nodes and the external network, three network interfaces have been created (Fig 4(a)). The interfaces are:
1. Management interface - This interface is for communication between the services installed on the three nodes.

2. Instance tunnels interface - This interface is for virtual machine data to flow between network node and compute node.

3. External interface - This interface is to provide communication with the internet.

For realizing these networks, hardware requirements are as under:

1. Three NICs on network node
2. Two NICs on compute node
3. One NIC on controller node
4. A Layer 3 hub/switch to connect the three nodes on the management interface.

NIC stands for Network Interface card.

The estimated bill of materials (hardware) for this work is given in Table 2. There is no cost of the software as it is open-source and freely available on OpenStack site.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Specification</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 server units (controller, network and compute nodes)</td>
<td>HP Business Desktop Elite 8300 Desktop Computer Core i7 - Small Form Factor</td>
<td>3× Rs. 39,000</td>
</tr>
<tr>
<td>3 Additional NICs</td>
<td>Technotech PCI express LAN card network adapter</td>
<td>3 NICs× Rs. 700</td>
</tr>
<tr>
<td>1 hub/switch</td>
<td>D-link DE 809TC</td>
<td>Rs. 1000</td>
</tr>
</tbody>
</table>

The networks and the nodes in the deployed architecture are shown in Fig.4.2.

V. RESULTS

The successful implementation of IaaS results in a graphical user interface called a dashboard which is used to manage images, networks, virtual machines, storage, hypervisors, etc. As shown in Fig.5(a) & (b), the user can authenticate himself via the login page of the dashboard that has been customized by editing certain files of horizon according to the requirement of the university.
Further, Fig. 6 shows all the instances, VCPUs, RAM, Volumes currently used in the cloud. The administrator can easily view all the cloud resources at any time via this page. All the authenticated users of the cloud are shown in Fig. 7.

Any user after being logged in have the capability to use any image being uploaded already or to upload any new image. A list of images, example, Ubuntu, Cirros have been uploaded via the user interface as shown in Fig. 8.
Fig. 7. List of users of the OpenStack IaaS.

<table>
<thead>
<tr>
<th>User Name</th>
<th>Email</th>
<th>User ID</th>
<th>Enabled</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>demo</td>
<td><a href="mailto:monikamor04@gmail.com">monikamor04@gmail.com</a></td>
<td>08f1dc3e46034a0e2a5ba068200351ad1</td>
<td>True</td>
<td>Edit</td>
</tr>
<tr>
<td>nova</td>
<td></td>
<td>2749743a3b04b960b75c13660b77908d77</td>
<td>True</td>
<td>Edit</td>
</tr>
<tr>
<td>glance</td>
<td></td>
<td>474945b9d7420b8a7679680b60a0a3</td>
<td>True</td>
<td>Edit</td>
</tr>
<tr>
<td>ceilometer</td>
<td></td>
<td>babc16764ab04b960b77908d77</td>
<td>True</td>
<td>Edit</td>
</tr>
<tr>
<td>cinder</td>
<td></td>
<td>e06068a2b23a45d96e826d1e252513</td>
<td>True</td>
<td>Edit</td>
</tr>
<tr>
<td>admin</td>
<td><a href="mailto:barkhapundir16@gmail.com">barkhapundir16@gmail.com</a></td>
<td>e10f1126725445babb1ed2102944b2d</td>
<td>True</td>
<td>Edit</td>
</tr>
<tr>
<td>neutron</td>
<td></td>
<td>1c111f4f19f14b7ac65d3528c50288</td>
<td>True</td>
<td>Edit</td>
</tr>
</tbody>
</table>
VI. OPERATION & MAINTENANCE
There are different issues that arise on the basis of the degree of advancement of cloud architecture adoption. In the beginner scenario, closed source model might not be the right choice as the source code is not released to the public. Open source model is preferable due to its source code availability on a universal level. Open models are customizable, more secure, has extended community and developer support and so OpenStack was used here. However, during or after implementation, the organization needs to do the operation and maintenance part. In current scenario, there are various hardware sizing considerations for the controller node that should be checked. Some of them includes the following:

- Number of instances that will run at the same time. For this, the database server should be sized accordingly.
- Ensuring that the messaging queue handles requests successfully and size accordingly.
- If different users will make multiple requests, the CPU of cloud controller should be able to handle it.
- If the users are provided with dashboard as main interface, make sure that all requests will be completed by adding more CPU as the dashboard makes many requests, even more than the API access.
- Starting instances and deleting instances is demanded on the compute node but also on the controller node because of need of all the API queries and scheduling. So the duration of running an instance should be monitored.
- External systems require network connectivity between the cloud controller and them. So there should be an effective authentication method.

Monitoring and Logging is required in case of any error or unrequired functionality to find the root cause of problem. OpenStack services use the standard logging levels that includes DEBUG, INFO, AUDIT, WARNING, ERROR, CRITICAL, and TRACE. Messages appear in the logs only if they are more severe than the particular log level. For example, TRACE is logged in the file only if the software has a stack trace, while INFO is logged for every message and it includes those messages too that are only for information. Log for all the services is traced at /var/logs/service. Keystone and horizon are handled a little differently. For the modification in the logging level of keystone, one can edit the /etc/keystone/logging.conf file and check the logger_root and handler_file sections. Logging for horizon is configured in the python file /etc/openstack_dashboard/local_settings.py.

VII. CONCLUSION & FUTURE WORK
The proposed system is a suitable low-cost solution for meeting the cloud computing needs of universities. This cloud solution is open-source and thus can be customized by other institutions according to their requirements. It is a paradigm for cloud enthusiasts to learn new technologies. More authentication with registration and login procedure can be included in the graphical interface. Further, in the future, the PaaS and SaaS layers can be configured on the top of current infrastructure.

ACKNOWLEDGMENT
This work is supported by the project Mobile Education Kit-2 funded by Nokia. We would like to thank the OpenStack community forums in helping us resolve the issues that were being faced while deploying the OpenStack services.

REFERENCES


