

Deploying an Open Source Cloud Computing Infrastructure for Academic Research

C. N. Udanor¹, F. I. Akaneme², C. H Ugwuishiwu¹, S. O Aneke¹,
O.A. Ezugwu¹, O. E. Nweke¹, M. E. Ezema¹, I. A. Ogbodo¹

¹(Department of Computer Science/ University of Nigeria Nsukka, Nigeria)

²(Department of Plant Science & Biotechnology/ University of Nigeria Nsukka, Nigeria)

Corresponding Author: C. N. Udanor

Abstract: Research outputs of academics from the developing world, Africa, in particular, have been lowly rated as a result of doubts placed on works from this region because of unavailability of state of the art research tools. This has made it extremely difficult for such researchers to publish their research in reputable journals. Some researchers desire to use HPCs to simulate, analyze or predict with their experimental data, but unfortunately, it is either their laptops do not have such powerful processing abilities or they do not have the necessary software to perform such tasks. This research work is aimed at ameliorating the challenges faced by researchers in higher education institutions, who would want to perform experiments on powerful virtual e-infrastructures like the cloud computing platforms and leverage on the computing power of high-performance computers (HPC). After comparing the most popular open source cloud computing frameworks, this research developed a public cloud computing infrastructure built on Synnefo and Ganeti open source infrastructure as a service (IaaS) for academic research in University of Nigeria Nsukka. The Lion Cloud IaaS makes available to researchers powerful multi-core processors, storage repository, as well as open source software tools to perform big data analytics, machine learning, etc.

Date of Submission: 02-06-2019

Date of acceptance: 17-06-2019

I. Introduction

One of the major challenges faced by academics and lecturers in tertiary institutions in developing countries could be lack of access to relevant research infrastructure and scientific equipment. Sometimes it could be as simple as getting access to electricity, or internet connectivity, which is taken for granted in many countries and other regions of the world. Or it could be the absence of state of the art research instruments, devices, or laboratory chemicals. In the information age, where research is conducted via virtual resources (VRs) like virtual laboratories, virtual infrastructures (VIs), using powerful multi-core processors or GPUs on high-speed broadband networks. The challenge is even greater for the academic based in developing regions of the world. This is because it is either the infrastructure is not available, or the requisite technical know-how is missing. A solution could be in having access to computing resources that the researchers do not have, such as of low-cost, on-demand VIs like the cloud.

Cloud computing is a model for enabling network users' on-demand access to a shared pool of configurable computing resources that can be rapidly provisioned and released to the client without direct service provider interaction. Cloud computing plays a major role in improving the quality of education to attain high performance by offering many benefits for education such as flexibility, scalability, providing infrastructure with low cost, collaboration, flexibility, and ease-of-use (Udanor, et. al, 2018). The following section provides a brief overview of cloud computing definition and terminologies, while the other parts of the paper analyses existing open source cloud computing infrastructures describe the processes of setting up a new cloud platform and discusses the results of the experiments.

Li Cloud Computing

The National Institute of Standards and Technology (NIST) defines Cloud Computing as “a model forenabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” (Lynn et al., 2015; Caron et al. 2013; Forer et al. 2012).

The NIST definition of cloud computing outlines four deployment models: private cloud, community cloud, public cloud and hybrid cloud (Caron et al., 2013; Mell and Grance, 2011). There are four types of cloud orchestration such as public, private, community, and a hybrid.

The Public cloud model is defined as a cloud computing infrastructure used by a third-party provider to provide services to different categories of clients via the internet.

Private Cloud is a cloud infrastructure made available only to organizational users or explicit clients and the infrastructure is managed by the organization themselves or third party providers.

Community Cloud is a cloud Infrastructure model particularly used for providing shared infrastructure between organizations and is managed by the organization itself or a third party.

Hybrid Cloud is a cloud infrastructure consisting of two or more cloud deployment types using special techniques that allow data and application transplant between them.

The cloud has three service models, as illustrated in fig 1:

Software-as-a-Service (SaaS): SaaS is a model of software deployment whereby one or more applications and the computational resources to run them are provided for use on demand as a turnkey service. (Obunadike et. al., 2014)

Platform as a Service (PaaS): PaaS provides an environment for easy development and deployment of applications. (Labourey, 2014).

Infrastructure as a Service (IaaS): In IaaS the capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. (Chase, 2014; Krishna, et al. 2016; Kurup, et al. 2015).

Features of IaaS include Ease of use, fault tolerance, low-cost of entry/maintenance, Performance, ease of expansion, API provisioning, compatibility with other platforms and Agility / Fast provisioning.

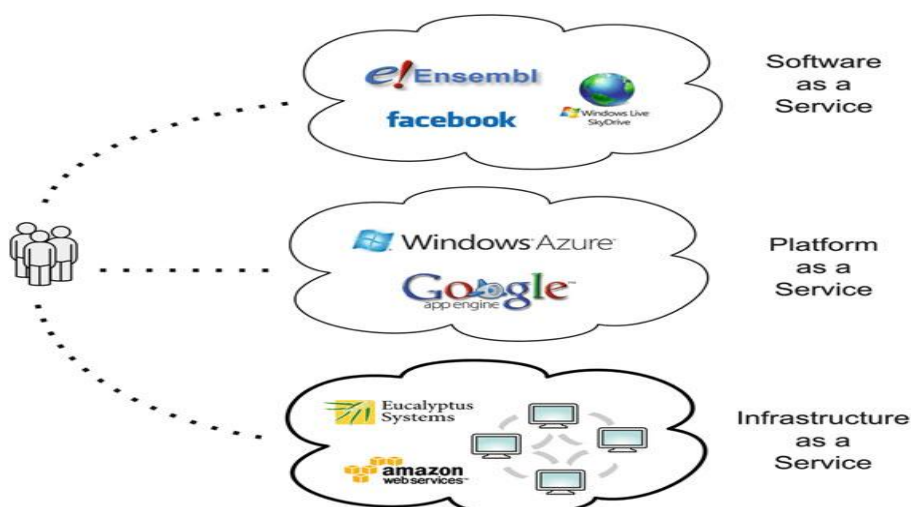


Fig. 1 Cloud computing service models. (Forer et al. 2012; Krishna, et al. 2016).

I.ii OPEN SOURCE IaaS FRAMEWORKS

Cloud Computing has become an important tool in delivering Infrastructure as a Service (IaaS) for users that require a high level of flexibility and management of their software stack. The cloud IaaS provides access to servers and storage possibilities (Forer et al. 2012).

The last decade has seen the rapid proliferation of vendors in the IaaS market through a public cloud development, major enterprise players include Amazon Web Services, with growing competition from competitors including Microsoft, Google, and IBM. While these services may offer flexibility to the user, commentators cite cost, lack of portability and lack of interoperability as drawbacks to enterprise IaaS. With each vendor promoting its own infrastructure and incompatible standards, users can suffer from vendor lock-in (Lynn et al., 2015). In some environments, it may be preferable for an organization to develop its own cloud, based on an open source framework. Open source frameworks offer an alternative to enterprise clouds by offering the freedom to modify the source code and build a cloud that is pluggable and open to extensions while reducing costs and avoiding vendor lock-in.

The development of open source solutions is of particular importance for the further proliferation of private, public and hybrid cloud development models (Claybrook, 2010). The fact is that there's no single, one-size-fits-all solution. One needs to do some homework to know what is most suitable for use. Specifically, one needs to understand the needed requirements in detail before approaching the problem.

As of 2014, five major open source frameworks led the market, these are OpenStack, Apache CloudStack, OpenNebula, Eucalyptus and Nimbus (Lynn et. al., 2015)

I.iii Review and Comparison of Open Source cloud IaaS

In this section, we review seven popular open source IaaS such as OpenStack, Eucalyptus, OpenNebula, CloudStack, Nimbus, Synnefo, and Abicloud.

1. OpenStack

OpenStack is a collection of open source software projects that enterprises or cloud providers can use to set up and run their cloud compute and storage infrastructure (Wen, et al. 2012; *openstack.org*, 2016). It can be used for private and public clouds as announced in 2010 (*openstack.org*, 2016). It was initially developed by Rackspace Hosting and NASA, it has since developed into a global collaboration of developers and cloud computing technologists. With over 4500 members, 850 companies, and support from major tech industry companies, OpenStack has a very powerful foundation. The Mission is “to produce the global Open SourceCloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable (Yunxia et al. 2014).

OpenStack has three core official software programs, Compute (Nova), Object Storage (Swift), and Image Service (Glance) (Kurup, et al. 2015).

Nova is the computing engine behind OpenStack. It is the cloud computing fabric controller and the central element of the IaaS system (Wang and Zhang, 2017). It manages all the compute resources of the OpenStack cloud. It includes an API (nova-api) server that accepts requests for creating virtual machines (VM), and their local or remote disks (Girishand Guruprasad, 2014), and associated metadata in a VM. Other processes within Nova include:

Nova-schedule – takes a VM request and determines where it should be placed.

Nova-compute – creates and terminates VM instances through a hypervisors’ API.

It includes support for Xen, XenServer/XCP, KVM, VMware vSphere, LXC, QEMU, UML, Hyer-V, etc. A **hypervisor** or **virtual machine monitor (VMM)** is computer software, firmware or hardware that creates and runs virtual machines. The Hypervisor runs on the host and acts as a bridge between the host and the guests.

Swift offers cloud storage software enabling the storage and access of objects across different nodes using a simple API. It is designed to be extremely scalable in size and capacity.

Glance provides image services for OpenStack. It provides a catalogue and repository of VM images, also allowing them to be used as templates when deploying new VM instances (Wen, et al., 2012).

Other official OpenStack programs include:

Cinder, Neutron, Horizon, Keystone, Ceilometer, Heat, and Trove.

Cinder is the block storage component of OpenStack. It provides persistent block level storage devices for use with OpenStack compute instances.

Neutron is OpenStack Networking; it provides networking capabilities to OpenStack and manages networks and IP addresses (source)

Horizon is the dashboard behind OpenStack. It provides a graphical interface for users and administrators.

Keystone provides identity services to OpenStack. It is a means of access to the OpenStack services available to different users.

Ceilometer provides billing services to individual users of the cloud. It also keeps an account of the users’ usage of different components of OpenStack.

Heat uses a template based orchestration mechanism to launch multiple composite cloud applications.

Trove is a database as a service for OpenStack. It provides functionality for both relational and non-relational database engines.

2. Eucalyptus

Eucalyptus (Elastic Utility Computing Architecture for Linking Your Program to Useful Systems) was developed by the University of California-Santa Barbara as an open source (Kurup, et al., 2015) infrastructure as a service and was released in 2008. It was transformed and run by a Eucalyptus company (Nwobodo, 2015). It allows for the installation of private and hybrid clouds. It is a private cloud and hybrid platform that implements the Amazon Web Service AWS (e.g. Amazon EC2 & S3) as Infrastructure as a Service (IaaS). Eucalyptus APIs with minimum modification or extension can also support other clients. Eucalyptus is a designed platform that fulfills the commercial need of EC2 cloud. It consists of a program called euca2ools, a front end for users, similar to EC2 program front end. Eucalyptus’ scalability is limited in comparison to some of the alternative open source IaaS solutions and, as such, has lost some of the popularity it had in the earlier days of its release (Lynn et al., 2015). The introduction of Scalable Object Storage has helped improve the scalability of Eucalyptus. Eucalyptus conforms to both the syntax and the semantic definition of the Amazon API and tool suite, with few exceptions. Eucalyptus targets Linux systems that use KVM and Xen for virtualization. It has been packaged for inclusion in the 9.04 release of Ubuntu, and other Eucalyptus packages exist for CentOS, Debian, openSUSE, and Red Hat Enterprise Linux 5.x.

Eucalyptus has six components (one being an optional component) (Claybrook, 2010): **Cloud Controller (CLC)** which manages the virtualization resources and APIs. It also provides a web interface.

Cluster Controller (CC) which controls the execution of VMs and their networking.

Storage Controller (SC) provides block-level storage to VMs (EBS)

Node Controller (NC) controls VMs via hypervisors (Albertson, 2013).

VMware Broker components and **Scalable Object Storage (SOS)**.

These components are divided into three layers: Cloud Layer/controller, Cluster Layer, and Cloud Controller.

Eucalyptus users have the capacity to monitor and execute all virtual machine deployed in their environment in an easy, flexible modular fashion. The platform is designed to enable flexibility in moving applications to the platform as desired on the public cloud. Moreover, it enables easy deployment on a hybrid cloud using any resources available in either private or public cloud to its advantage. Part of its disadvantage is the lack of interface for virtual machine management and advanced monitoring (Lynn et al., 2015)

3. OpenNebula

OpenNebula is a cloud technology initially released in 2008 by the European Union and now operates as an open source project. OpenNebula is an open source cloud computing tool specifically designed as modular to allow easy integration with different environment and hypervisors, further to manage the heterogeneity and complexity of distributed large infrastructure (Yunxia, et al. 2014). While it is most commonly used for private cloud, it also supports public and hybrid clouds. The OpenNebula mission is "to become the simplest cloud enabling a platform for the enterprise" and their purpose is "to bring simplicity to the private and hybrid enterprise cloud". OpenNebula has a large user base with telecommunication companies (e.g. Telefonica), system integrators (e.g. Logica), supercomputing centres (e.g. SARA), etc. (Yunxia, et al., 2014).

The OpenNebula architecture is a classical cluster with a front end and a set of cluster nodes to run the VMs. At least one physical network is needed for connecting all cluster nodes with the front end. The platform is mainly used for management of data centre private cloud and cluster of infrastructures, including supports in connecting the hybrid cloud to local and public infrastructure. OpenNebula can be used to build a cloud computing environment that is highly scalable. Moreover, public cloud is fully supported by the provision of interfaces and vital functions to VMs, network and storage management. The key difference of this platform with nimbus is that Nimbus uses remote interface based on EC2 and WSRF in which users can process entire security-related cases while OpenNebula lacks such capability.

The platform consists of three main components as shown in figure4 below such as Core Virtual Infrastructure Manager, Capacity Manager, and drivers.

Core Virtual Infrastructure Manager: The lifecycle of the virtual machine is managed with the Core Virtual Infrastructure Manager by ensuring a smooth run of basic operations including migration, monitoring, and deployment.

Capacity Manager: The second component Capacity Manager (scheduler) module is principally responsible for managing all functionality from the Open Nebula core such as balancing workloads in virtual machines.

Drivers: The third component ("drivers") supports all platforms under it because every necessary driver to regulate data transfer and to manage VM on all host, irrespective of the hypervisors are contained (Claybrook, 2010).

OpenNebula does not support Graphic User Interface (GUI) which is the main drawback to the platform. The platform is a preferable choice in the data centre because it is an open source, extensible interface, and flexible structure cloud solution.

OpenNebula compared with Eucalyptus clouds has more power for private cloud support and dynamic, scalable, management of virtual machines on clusters. The platform in regard to the hybrid cloud provides an elastic mechanism and on-demand resource access like Amazon EC2.

This architecture is flexible and modular, making it easier to integrate multiple storages, network infrastructures and hypervisor technologies than using OpenStack.

The OpenNebula software consists of three layers: tools, drivers, and core. An example of an architecture reference model for IaaS clouds is shown in Figure 3.

The Tools layer provides interfaces to communicate with users and allows users to manage VMs through the interfaces, which include command line interface (CLI) and libvirt API. This layer also contains a scheduler that manages the functionality of the core layer (Wen, et al. 2012).

The Drivers layer contains components that communicate directly with the underlying operating system and capture the underlying infrastructure as an abstract service.

The Core layer contains the components used to perform user requests and control resources.

4. CloudStack

CloudStack was developed by Cloud.com and in 2010 released the majority of the project's source code. In 2011 CloudStack was acquired by Citrix who subsequently released the remainder of CloudStack's code. In April 2012 Citrix donated CloudStack to the Apache Software Foundation (ASF). It was accepted into the Apache Incubator and then graduated from the Apache Incubator in March 2013 to become a Top-Level Project of ASF (Nwobodo, 2015). CloudStack doesn't have quite the same foundation of users as OpenStack but a number of large companies do use Apache CloudStack including Apple, British Telecom, Dell, Nokia, and Fujitsu. CloudStack supports VMware ESX, Xen and KVM (and eventually Hyper-V). The management server also manages the assignment of guest VMs, assignment of IP addresses, allocates storage during VM instantiation and manages snapshots, disk images, and ISO images. It also supports Windows and Linux guest operating systems. It provides a web interface for administrators and the end-users, as well as API interfaces for CloudStack API and also the Amazon EC2 interface. It is a single point of configuration for the IaaS cloud and during deployment, the user informs it of the resources to be managed. The CloudStack deployment is organized with hosts contained within clusters, clusters contained within pods, pods contained within zones, and zone contained within regions. Regions are the largest available units in the CloudStack deployment. Regions essentially consist of a number of available zones, with zones roughly being the equivalent of a data centre. CloudStack components are Management Server, Hypervisor Nodes, Storage Nodes, and Layers: Zone, Pod, Cluster, Host, Primary Storage, and Secondary Storage.

5. Nimbus

Nimbus is an open source IaaS project that was initially released in 2009 (Kurup, et al. 2015). Nimbus' mission is "to evolve the infrastructure with an emphasis on the needs of science". Nimbus is built with three goals in mind:

- Enable resource owners to provide their resources as an infrastructure cloud.
- Enable cloud users to access infrastructure cloud resources more easily.
- Enable scientists and developers to extend and experiment with both sets of capabilities.

Nimbus cloud platform is an integrated open set tool, used to provide infrastructure as a service. It is supported by the University of Florida and Chicago (Zeng et al., 2012). Nimbus platform is specifically designed based on science community interest such as batch schedulers, proxy credentials and best efforts allocation, etc., although it has recently supported non-scientific applications. It enables users to provide, build various remote computing on-demand resources via the deployment of virtual machines. Nimbus permits a combination of Amazon, OpenStack, and many other clouds.

According to Wind (2011), Nimbus cloud seems more complicated than many cloud platforms as command and results are inputted in command lines, making nimbus appear extremely complex due to specific applications. Nevertheless, the software is particularly unique based on a greater level of flexibility as it supports almost every hypervisor. The provisions on the consistent and regular backup of platform workload help prevent issues of data loss in the event of a malfunction. Nimbus is supplied in Apache version 2 licenses like OpenNebula. Nimbus platform supports XEN and KVM virtualization technologies except for VMware (Kurup, et al. 2015).

Nimbus platform functional components can be grouped into three: client supported module, service supported module and resources management module.

The client supported modules is used to support users especially with quick instance launches, one-click clusters and consists of context client module, cloud client module, EC2 client module including reference client module.

Service supported module is primarily used in providing various cloud services and the modules include EC2 WSDL module, remote interface module, and web service resource framework module.

Background resource management module is principally used to manage a range of physical resources on a cloud computing platform such as IaaS gateway module, work service management module, EC2 plus other supported platform modules, workspace controller, workspace resource management module and workspace pilot module

The main infrastructure components are the Workspace Service site manager, a web services resource framework (WSRF) based remote protocol implementation, an EC2 based remote protocol implementation of their SOAP and Query APIs, Cumulus, RM API, the cloud client, the reference client, the Workspace Pilot, the workspace-control agent, and the metadata server.

6. AbiCloud Platform

AbiCloud computing platform is designed and developed by Abiquo, an established cloud computing company in Barcelona Spain specialized in the development of the cloud platform. Moreover, the platform can enable user's easy scaling, orchestration, deployment of a variety of cloud utilities. (Nwobodo, 2015). These

have the potentiality of accomplishing IaaS services within a short time, and further leveraging such capability in delivering managed hosting services or self-service. IaaS via simplified interface or the combination of both. AbiCloud platform can be used to manage EC2, facilitate deployment and implementation of hybrid or private cloud depending on user requirement, including easy deployment to another AbiCloud platform. This can support different kinds of virtual machines e.g., VMware, KVM, vBox and is developed based on Java AbiCloud VMS (Virtual Monitor System) is designed purposely to monitor virtual infrastructure primarily to study about status and events.

7. Synnefo

Synnefo is a complete open source cloud stack that provides Compute, Network, Image, Volume and Storage services, similar to the ones offered by AWS (Koukis, et al. 2013).

The Synnefo Open Cloud Computing Interface (OCCI) acts as an API middleware between the OCCI protocol and the Synnefo API. Synnefo cloud software utilizes Ganeti as a low-level virtualization layer to provide compute and storage cloud over OCCI 2016 - Special Session on Experiences with OCCI an extended OpenStack API. The ~okeanos IaaS, which provides compute and storage resources to the Greek and European academic communities, is the largest Synnefo deployment (Parák, et, al., 2016).

Synnefo (open source IaaS software) and ~okeanos (IaaS service powered by Synnefo) are maintained and provided by the Greek Research and Technology Network (GRNET), which is the National Research and Education Network (NREN) provider of Greece. The principal role of GRNET is to operate the Greek Academic network, connect it with global academic communities and institutions, and provide them with cutting edge IT services and technology. The Synnefo OCCI interface features a distributed design of separate components (e.g., snf-occi and astakos-vo-proxy) which are connected via RESTful APIs. Every bit of functionality provided by Synnefo is exported to end users via a clean, RESTful API called the Synnefo API. The Synnefo API is a superset of the OpenStack API. The rationale behind the aforementioned design choice is to ensure adaptability to the evolution of both OCCI and Synnefo, as well as robustness and security through the deployment on isolated nodes.

The main component of the interface is called **snf-occi**, a service that maps OCCI v1.1 requests to OpenStack/Synnefo. It is designed to run as a stand-alone service, connected to Synnefo through its RESTful API. (Parák, et, al., 2016).

Astakos is the authentication and policy (e.g., user quota) enforcement component of Synnefo. To facilitate the mapping between OCCI and Synnefo users, the proxy maintains an LDAP directory with the minimum user information needed for the mapping. Possible changes in the status of the supported user pool are reflected by frequently updating the directory with the assistance of a human operator.

Synnefo offers a no-hassle, intuitive Web UI for all custom architectures. Synnefo supports KVM-based VMs, managed by Google Ganeti. Synnefo uses Google Ganeti for VM management at the backend (Koukis, et al., 2013). The Ganeti-based control plane ensures stability, redundancy, manageability, upgradability and above all production readiness. Ganeti employs a simple, clean design with readable, well-commented and well-maintained code. KVM does full system virtualization and supports Microsoft Windows, Linux, and BSD deployments inside its VMs. So far, server Images for Red Hat Enterprise Linux/ CentOS, Fedora, Debian Linux, Ubuntu/Kubuntu, Arch Linux, Gentoo Linux, Microsoft Windows Server 2008R2/2012 and FreeBSD have been tested extensively inside Synnefo deployments, using virtio-based storage and network drivers for minimal virtualization overhead.

The API is a superset of the OpenStack API and allows the use of 3rd party tools with Synnefo. Synnefo transforms all existing hardware resources to commodity virtual components (Synnefo.org).

Other Key Features of Synnefo are:

Intuitive Web UI: Synnefo comes with a clear, simple Web UI through which the user may quickly manage existing compute, network and storage resources.

Public Networking: Synnefo supports full IPv4 and IPv6 connectivity to the public Internet for its VMs.

Private Networking: Synnefo provides virtual Ethernets as a separate resource, giving the user freedom to create arbitrary network topologies of interconnected VMs, e.g., for multi-tiered deployments of enterprise software.

Out-of-the-box Firewalling: The user may protect each public IPv4/IPv6 interface with a virtual firewall, choosing from a number of predefined firewall configurations.

Self-healing distributed storage backend: Files, Images and VM storage volumes are physically stored as objects in a distributed, redundant, object-based storage backend (RADOS).

Intelligent Storage with snapshots and clones: Efficient VM provisioning is key to delivering elastic scaling of cloud resources. A custom volume composition layer in Synnefo provides for low-overhead snapshots of VM storage volumes and the creation of clones from them (Albertson, 2013).

COMPARISON OF OPEN SOURCE SOLUTIONS

Table 1 provides a summary and comparison of the features of all seven open source cloud IaaS solutions.

Table 1: Comparison of Open source IaaS Solution

| Features | OpenStack | Eucalyptus | OpenNebula | CloudStack | Nimbus | Synnefo | Abicloud |
|-----------------------------|---|------------------------|---------------------------------|---|------------------|---|-------------------------------|
| Year Developed | 2010 | 2008 | 2008 | 2010 | 2009 | 2011 | |
| Cloud type | public | Public, private | Private | Public | private | Public | Public/private |
| Compatibility | Supports EC2, S3 | Supports EC2, S3 | Open, multi-platform | Supports EC2, S3 | Supports EC2, S3 | | EC2 Not supported |
| Deployment Manner | Web interface/ command line | Command line interface | Command line/ web interface | Web interface/ CLI | Command line | Web UI, Command line or native syncing client | Web interface |
| Hypervisor support | KVM, Xen, VMWare, ESX, Hyper-VLXC | VMWare, Xen, KVM | Xen, VMWare, ESX, KVM, ESXi | VMWare, Oracle VM, KVM, XEN, Citrix | Xen, KVM | KVM | Virtual box, Xen, VMWare, KVM |
| Web Interface | Native API, Amazon EC2 API | Web Service | LibVirt, EC2, OCCI, API | | EC2, WSDL, WSRF | Open Cloud Computing Interface (OCCI) | LibVirt |
| OS Support | Linux, Windows X86 Server | Linux | Linux | Mac OS x, Asianux, Cent OS, Debian, DOS | Linux | Debian Linux | Linux, windows |
| Development language | Python | Java, C, Python | Java, C++, rUBY | Java | Java, python | Python | Ruby, C++, Python |
| Security | Protection against DOS/ faulty clients. Keystone for identity mgt | Public and private key | Authen password, RSA, SSH, LDAP | Cloudsstack security groups | PKI | Authentication tokens | Code Access System |
| OS License type | Apache license version 2 | BSD License | Apache version 2 | Apache license | Apache version 2 | GNU License | Common public attribution |

After comparing the cloud IaaS infrastructures we decided to use the Synnefo cloud stack. The reasons are explained in the next section.

II. Material And Methods

During the period of this work, the researchers deployed two numbers of HP PROLIANT GL360, G5 servers, each with 2.4GHZ XEON CPUs, 4GB RAM, 1TB HDD (Udanor et. al, 2015), with Debian Linux 9 (Debian Jessie) installations. Each machine has a published public IP address. On these machines, we deployed and configured the Synnefo cloud stack version 0.19. Synnefo was developed by the Greek Research and Education Network (GRNET). (ref: <https://www.synnefo.org/>). Synnefo is an open source cloud framework used for creating a massively scalable IaaS cloud. It uses Google's Ganeti for the low-level Virtual Machine (VM) management.

The choice of Synnefo is based principally on two reasons:

- i. Compared to OpenStack, which is adjudged to be the best open source cloud infrastructure (Lynn et. al., 2015), Synnefo requires minimal hardware. While OpenStack requires not less than five (5) physical machines for full production cloud, Synnefo requires only two (2). This makes the cost of implementing Synnefo based cloud much lower, especially for infrastructure-challenged institutions in developing countries.
- ii. While OpenStack requires highly trained technical personnel to handle the installation processes because of its rigorous and complex architecture, Synnefo requires very little time of training and learning to set up.

Other reasons for using Synnefo includes the following:

Synnefo is perfect for non-commercial clouds, like educational and institutional or research centre clouds (Parák, et, al., 2016).

II.1 Synnefo Cloud framework

Synnefo is a complete open source IaaS cloud stack written in Python that provides Compute, Network, Image, Volume and Object Storage services, similar to the ones offered by AWS. Synnefo manages multiple Ganeti clusters at the backend for the handling of low-level (VM) operations and uses Archipelago to unify cloud storage. To boost 3rd-party compatibility, Synnefo exposes the OpenStack APIs to users. (ref: <https://www.synnefo.org/docs/synnefo/latest/index.html>)

Synnefo has three main components which provide the corresponding services:

- Astakos: Identity/Account/Quota services
- Pithos: Object Storage service
- Cyclades: Compute/Network/Image/Volume services

It unifies storage resources (Objects/Volumes/Images/Snapshots) using Archipelago as the common storage substrate for all services.

This is an overview of the Synnefo services is shown in fig 2:

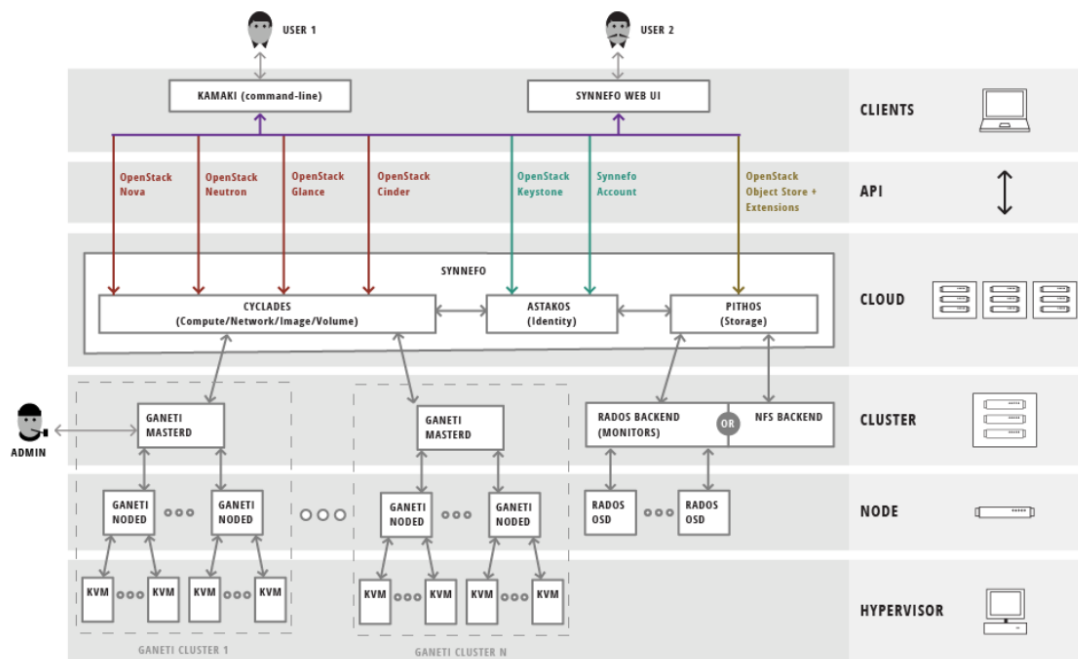


Fig. 2: SynnefoArchitecture (Albertson, 2013).

Ganeti

Ganeti started as an internal Google project but was made open source in August 2007. It was used primarily for back-office servers for Google and focuses on hardware fault-tolerance. It is characterized by local block-level storage and cheap commodity hardware.

Ganeti Components

- Master daemon which Controls overall cluster coordination
- Node daemon, Controls node functions (storage, VMs, etc)
- Conf daemon, Provides a fast way to query configuration
- API daemon - Provide a remote API
- Htools - Auto-allocation & rebalancing tools

Ganeti has the following third-party add-ons:

Ganeti Web Manager

- o Simple IaaS management
- o Permissions, groups
- o Just the UI

Synnefo

- o Ganeti as a cloud service
- o Cloud, API, UI

- DRBD, LVM, Shared File, RBD

Setting up Synnefo

Synnefo has a well-documented Administrative Guide for setting up the cloud IaaS. To install Synnefo several service components are required, and they must be installed one after the other in the right sequence. Such services include:

- Identity Management (Astakos)
- Object Storage Service (Pithos)
- Compute Service (Cyclades)
- Image Service (part of Cyclades)
- Network Service (part of Cyclades)

The above services were installed in two server machines after the Debian Linux OS has been installed. Each machine has an assigned public IP address. The machines and their IPs are (last part of the address is masked out):

196.222.12x.xx astakos.unn.edu.ng – node 1
196.222.12x.xx pithos.unn.edu.ng – node 2

To be able to download all the installation components needed we had to edit the sources.list file under the /etc/apt folder in each machine. We added the repository sources in the sources.list file as follows:

Deb <http://apt.dev.grnet.gr> Jessie/
Deb-src <http://apt.dev.grnet.gr> Jessie/

Using the curl command we imported the repo's GPG key to enabling us to have access to the installation files. Like:

```
curl https://dev.grnet.gr/files/apt-grnetdev.pub | apt-key add -
```

The following general Synnefo dependencies were installed in Node 1 by running the commands in the guide:

- apache (http server)
- public certificate
- gunicorn (WSGI http server)
- postgresql (database)
- rabbitmq (message queue)
- ntp (NTP daemon)
- gevent
- dnsmasq (DNS server)
- Archipelago

While the following general Synnefo dependencies were installed in Node 2:

- apache (http server)
- gunicorn (WSGI http server)
- postgresql-client (database)
- ntp (NTP daemon)
- gevent
- certificates
- dnsmasq (DNS server)
- NFS directory mount
- Archipelago

After installing and editing all the configuration files, the Astakos and Pithos were both installed and tested on the web browser using the urls <https://astakos.unn.edu.ng> and <https://pithos.unn.edu.ng>, respectively. Fig 3 and 4 show the respective outputs.

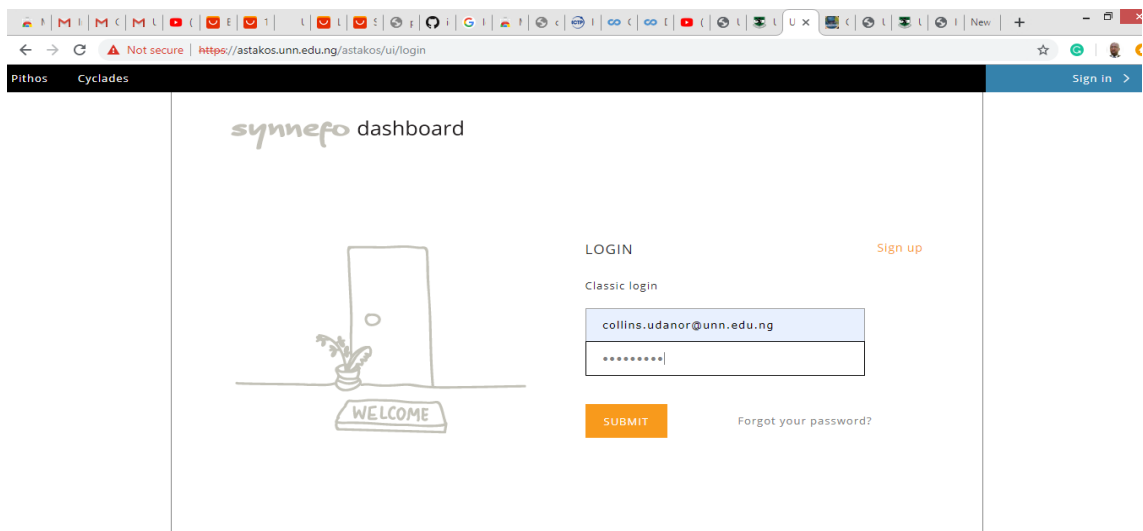


Fig. 3: Astakos (Node 1) – the Dashboard interface

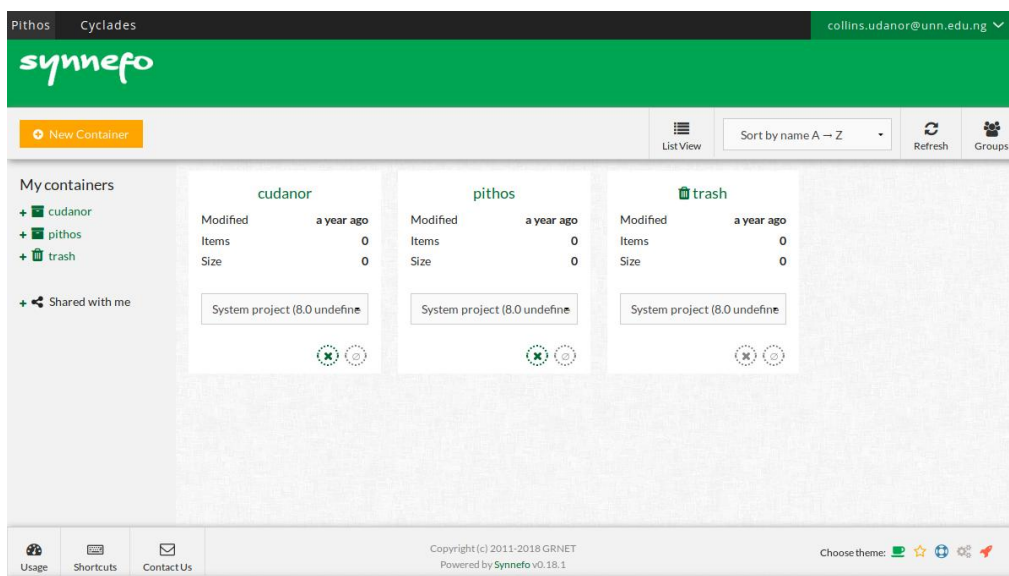


Fig. 4: Pithos (Node 2) – the Storage interface

After successfully setting up the two nodes, using apt-get install the command line interface known as Kamaki is installed. Kamaki is an OpenStack API client library that comes with custom extensions that is specific to Synnefo.

Installing Cyclades

Cyclades is the service that enables the Synnefo cloud to provision virtual machines. Before going to this stage we made sure that nodes 1 and 2 were fully functional since Cyclades depends on them. Again, Cyclades uses the Ganeti cluster manager to provide low-level VM management at the back-end. Ganeti uses the native (kernel VM) KVM hypervisor in Linux to create VMs, so we ensured that the qemu-kvm was installed. A successful Ganeti installation concludes with a working ‘GANETI-MASTER’ and a number of ‘GANETI-NODE’s installed on both Node 1 and Node 2, respectively. We also configured the network bridges and set up private networks and their IP ranges for the VMs. Finally, Cyclades was installed on Node 1. The Cyclades web interface runs on <https://astakos.unn.edu.ng/cyclades/ui>.

III. Result

After 3 weeks of rigorous installations, consulting, and following up on forums we concluded the installation. Users were invited to register on the infrastructure by creating their user accounts from the astakos dashboard. Astakos provided the authentication also. Fig 5 shows what a successful login looks like on astakos, while fig 6 shows the user's detailed information and fig 7 shows how a user can create a project or join an existing one.

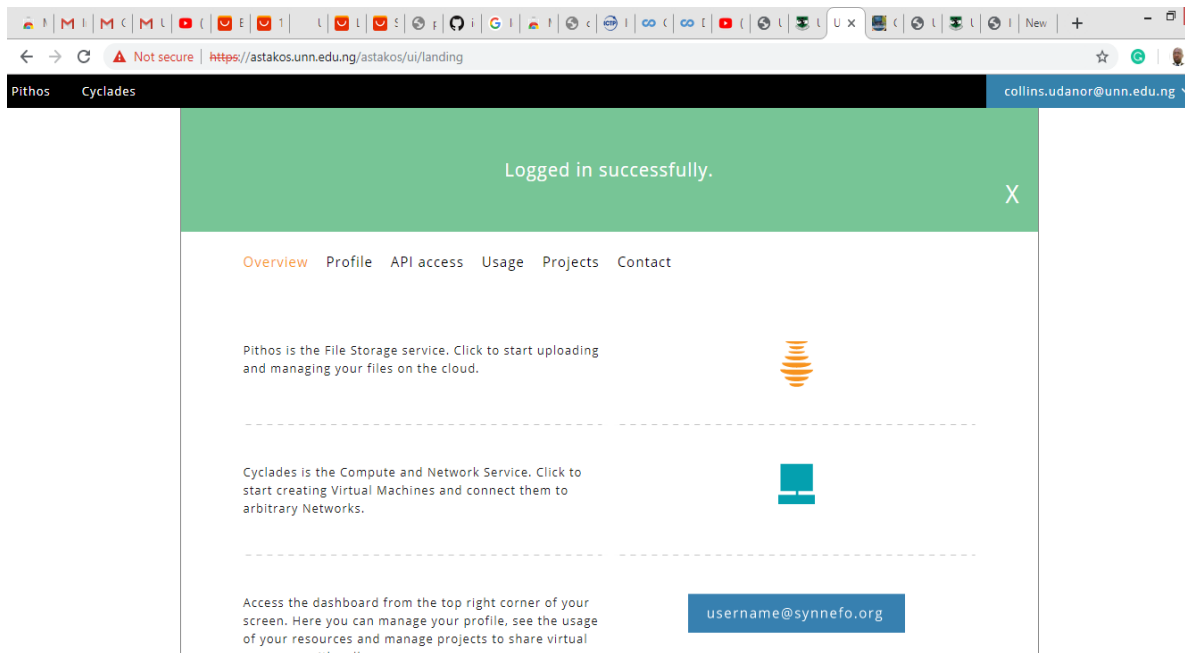


Fig 5: Successful user login on the Astakos Dashboard

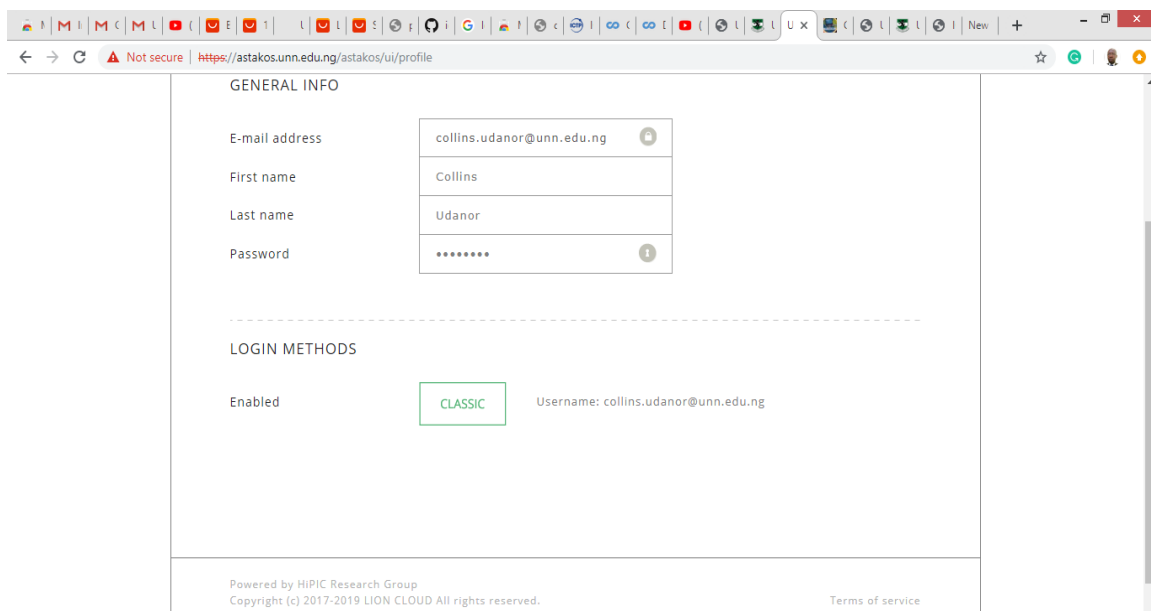


Fig 6: User detail general information

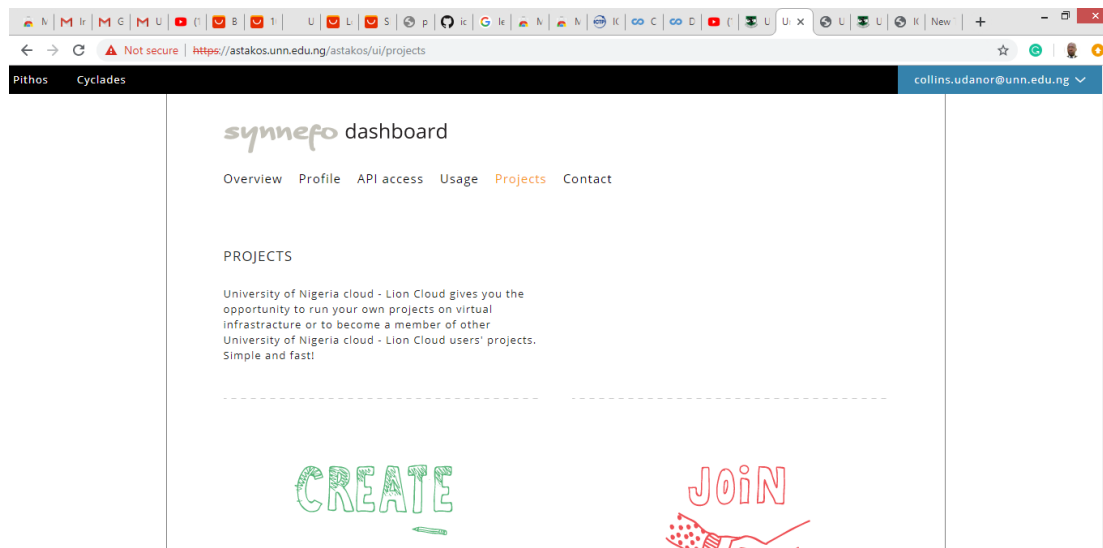


Fig 7: Selecting tasks to perform from the Dashboard.

IV. Discussion

Cloud computing has drastically reduced the cost of computing by making available to the user powerful processors, large storage disk spaces and powerful and up to date applications which were hitherto hard to come by to users in diverse areas of business. The academic industry is not left behind. Powerful virtual labs have been made available at little or no fees by the major players in the cloud industry, such as Google (the Colaboratory), Oracle (the Oracle Academy), etc.

We have been able to design and build a cloud computing infrastructure locally in our university that is able to allow a new user register, create VMs, upload, and store research data in his personal container. The platform also makes it possible for researchers to install their own application software on their VMs and enjoy the use of powerful multi-core processors on our HPC servers. This means they can upload jobs and allow the virtual machines to run the jobs on the Lion cloud. This takes the overhead out of their PCs. Unlike similar works reviewed below, our work provided the most extensive review of open source cloud solutions. While some of the works did not involve the design and implementation of cloud infrastructures, ours implemented an e-infrastructure that provides IaaS, PaaS, and Security as Service at no cost to the higher education community. Our work gained recognition beyond our university to the extent that we received requests from universities in the country to join the Lion Cloud.

Cordeiro et al.(2010) in the work titled *Open Source Cloud Computing Platforms* focused on giving the reader an understanding of some major existing open cloud computing solutions such as Xen Cloud Platform (XCP), Eucalyptus and OpenNebula. In this paper, the researchers did and presented a comparative description of the three popular cloud computing solutions – (XCL), Eucalyptus and OpenNebula. The work also considered the differences in each solution and described with examples of how each can be used. It was hoped that by giving these insights on some of the main differences between them, one may decide where and when each solution may be more appropriate for its use.

Niranjani et al. (2015), proposed an implementation of a prototype private cloud setup with open-source cloud operating environment on a Linux platform and measuring the performance of cloud by running the scientific applications. The study and evaluation of its performance of user codes in cloud platform with dynamic configuration of Infrastructure-as-service and in non-cloud hardware platform were done and results analyzed. To increase the utilization of resources, kernel-based virtual machine was used to provide virtualization infrastructure. The nodes such as compute node and cloud controller node were installed and configured. The nodes were connected internally to OpenStack dashboard with the internal network. Client users who are ready to use the cloud service were connected through the external network to the controller node of the private cloud. The scientific applications considered for benchmarking the prototype private cloud setups were Reactor Analysis code and Molecular dynamic calculations. The result of the experiment came by running scientific applications on virtual machines dynamically provisioned by the cloud infrastructure. The performance in terms of execution times of reactor shielding analysis code and molecular dynamic applications were measured in both cloud and non-cloud platforms. The outcome showed that the performance of these applications on a virtualized cloud environment is almost comparable with the performance of the same on non-cloud hardware.

Kumar and Gupta (2014) presented a research paper that explained the benefits, components, and architecture of Eucalyptus and gave a reason for choosing Eucalyptus, open source cloud platforms. Analysis as

well as summarization that would offer help to users in understanding the characteristics and provide an allowance for users to choose better services that fit their requirements. In this case, users were expected to be able to make a more informed decision on open source cloud platform they would use according to the cloud types, interfaces, compatibility, implementation, deployment requirement, as well as development support. The researchers were able to show how cloud computing has the potentials in transforming the IT industry, making software to be more attractive as a service, and Open Cloud platforms offer real alternatives to end user for improved flexibility as well as on-demand services easily.

Odun-Ayo et. al (2018) in their work have proved that Cloud computing has enabled massive development in hardware and software infrastructure. Some of the benefits of Open Source Software (OSS) which were done in terms of *price, flexibility, efficient system integration, cheap enhancement, mobility transparency with No Lock-In*. Cloud computing was also shown to offer elastic, scalable on-demand services to cloud users through the provision of Internet service. Such services helped to improve operations both for small and big enterprises alike. The developments in cloud computing reflect the advances being made in the area of OSS. OSS is available for the public to use and at the same time contribute to cloud developments. Several OSS are being used to enhance activities on the cloud. Such OSS like OpenStack are providing massive opportunities in cloud computing. However, security vulnerabilities in OSS must continue to receive greater attention.

V. Conclusion

Building a cloud infrastructure is quite expensive in terms of purchase of HPC servers, procuring and maintaining public IP addresses, and also time-consuming. Most importantly, it requires a steep learning curve. Patience is a virtue that one must have to succeed in the process. Synnefo has its challenges, it is not that easy to set up. The forum is not very active, so one may post a question, and for weeks no one may have responded to it. Again, unlike OpenStack, Synnefo does not have many fans. It has not been deployed in as many projects as OpenStack. It finds its clientele mostly among educational institutions. Deploying a local cloud infrastructure has advantages of quick response times, and control in terms of security.

Further work will focus on the development of a custom cloud portal for the Lion cloud and assessment of usage by the academic community.

Acknowledgment

This research was made possible by the financial award from Nigeria's Tertiary Education Trust Fund's Institutional Based Research award (TETFund-IBR 2016).

References

- [1]. Collins N.Udanor, Stephen O Aneke & Cynthia U Obillo. Implementing Cloud Computing Storage In Nigerian Tertiary Institutions, Proceedings of the iSTEAMS Multidisciplinary Cross-Border Conference University of Ghana, Legon, Accra, Ghana - October, 2018
- [2]. Nwobodo I. (2015). A Comparison of Cloud Computing Platforms. International Conference on Circuits and Systems (CAS 2015). https://www.researchgate.net/publication/282323121_A_Comparison_of_Cloud_Computing_Platform.
- [3]. Lynn T., Hunt G., Corcoran D., Morrison J. and Healy (2015). A Comparative Study of Current Open-source Infrastructure as a Service Frameworks. 5th International Conference on Cloud Computing and Services Science (CLOSER 2015), pp 95-104 www.scitepress.org/Papers/2015/54233/54233.pdf.
- [4]. Caron E., Toch L., Rouzaud-Cornaba J. (2013). Comparison on OpenStack and Open-Nebula performance to improve multi-Cloud architecture on cosmological simulation use case. [Research Report] RR-8421, INRIA. pp.23. <hal-00916908>. <https://hal.inria.fr/hal-00916908/document>.
- [5]. Parák, B., Šustr, Z., Kimle, M., Fernández, P., García, Á., Sachtouris, S. and Muñoz, V. (2016). "Evolution of the Open Cloud Computing Interface. Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER 2016)" - Volume 2, pages 339-346 <http://www.scitepress.org/Papers/2016/59341/59341.pdf>.
- [6]. Bill Claybrook (2010). Five open source tools for building and managing clouds. <https://searchcloudcomputing.techtarget.com/tip/Five-open-source-tools-for-building-and-managing-clouds>.
- [7]. Yunxia J., Bowen Z., Shuqi W. and Dongnan S. (2014). Research of Enterprise Private Cloud Computing Platform. Based on OpenStack. *International journal of Grid distribution and computinf*. Vol.7, No.5, pp171-180. www.sersc.org/journals/IJGDC/vol7_no5/16.pdf.
- [8]. Wang L. and Zhang D. (2017). Research on OpenStack of open source cloud computing in colleges and universities' computer room. 3rd International Conference on Advances in Energy, Environment and Chemical Engineering. <http://www.iopscience.iop.org/article/10.1088/1755-1315/69/1/012140/pdf>.
- [9]. Girish L. S. and Guruprasad H. S. (2014). Building Private Cloud using OpenStack. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*. https://www.researchgate.net/publication/274388772_Building_Private_Cloud_using_OpenStack.
- [10]. C.N. Udanor, F.I. Akaneme, Okezie, CEA, Ogwo B, Aneke S, Ogbuokiri B.O, Ezugwu A.O, Ugwuishiwu C.H. Deployment of an e-Infrastructure for Academic Research. The 7th EAI International Conference on e-Infrastructure and e-Services for Developing Countries (AFRICOMM 2015). Springer Lecture Notes of the Institute of Computer Sciences, Social Informatics and Telecommunications Engineering. Volume 171. [10] Linticum D. (2018). 7 essential open-source tools for cloud management
- [11]. Synnefo. White Paper. Published in *Synnefo Cloud* <https://techbeacon.com/7-essential-open-source-tools-cloud-management> Platform. Available at: https://www.synnefo.org/downloads/synnefo_white_paper_Hi.pdf.

- [12]. Albertson L. (2013). Comparing Ganeti to other Private Cloud Platforms <http://www.2013.linuxfestnorthwest.org/.../Comparing%20Ganeti%20to%20other%20private%20>.
- [13]. Mell, P. and Grance T (2011). The NIST Definition of Cloud Computing. Available at: <https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>.
- [14]. Wen X., Gu. G., Li Q.,GaoY.,Zhang X. (2012). Comparison of open-source cloud management platforms: OpenStack and OpenNebula. 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD) <https://ieeexplore.ieee.org/document/6234218/citations#citations>
- [15]. Koukis V., Venetsanopoulos C., and Koziris N. (2013). Synnefo: A Complete Cloud Stack over Ganeti, VOL. 38 NO. 5. Available at: https://www.usenix.org/system/files/login/articles/02_koukis-online.pdf
- [16]. Forer L., Schönherr S., Weißensteiner H., Specht G., Kronenberg F., and Kloss-Brandstetter A. (2012). Bringing Computational Power to Medical Genetics. <https://www.springer.com/cda/content/document/cda.../9783709109465-c1.pdf?>.
- [17]. Krishna, M., Nandimandalam V. and Choi E (2016). COMPARISON OF OPEN-SOURCE PAAS ARCHITECTURAL COMPONENTS. pp. 53–62. Available at: <https://airccj.org/CSCP/vol6/csit64906.pdf>
- [18]. Obunadike, G., Tyokyaa, R. and Umeh, A., (2014). A Phase Approach for Adopting Private Clouds as a Collaborative Platform for Nigerian Universities. Information and Knowledge Management. www.iiste.org ISSN 2224-5758 (Paper) ISSN 2224-896X (Online) Vol.4, No.9.
- [19]. Labourey, S. (2014). CloudBees: 4 Key Things You Should Know About Platform-as-a-Service. Dzone Research - The 2014 Cloud Platform Research Report.
- [20]. Chase, N. (2014). Understanding the Cloud and Building Your Own Infrastructure-As-A-Service. Cloud Platform Research Report, DZone. 15, 16.
- [21]. Cordeiro T., Damalio D., Pereira N., Endo P., Palhares A., Gonçalves G., Sadok D., Kelner J. , Melander B., Souza V. & Mångs J. (2010) Open source cloud platforms , 2010 Ninth International Conference on Grid and Cloud Computing, 366 – 371, DOI 10.1109/GCC.2010.77. Available: https://www.researchgate.net/publication/221602268_Open_Source_Cloud_Computing_Platforms
- [22]. Kumar R. & Gupta S. (2014) Open Source Infrastructure for Cloud Computing Platform Using Eucalyptus, *Global Journal of Computers & Technology* 1(2) 44 – 50, ISSN: 2394-501X. Available: https://www.researchgate.net/publication/271531172_Open_Source_Infrastructure_for_Cloud_Computing_Platform_Using_Eucalyptus
- [23]. Niranjani S., Jehadeesan R., Vanishree S. & Karthick P. (2015) A Prototype For Private Cloud Implementation Using Open-Source Platform, *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)* ISSN: 0976-1353 13(1) 311 – 316 . Available : <https://pdfs.semanticscholar.org/5c6b/a905420cf7e0c7336b7a887b14e8cca0256c.pdf>
- [24]. Odun-Ayo I., Falade A., & Samuel V. (2018) Cloud Computing and Open Source Software: Issues and Developments, *Proceedings of the International MultiConference of Engineers and Computer Scientists Vol I* 2018, March 14-16, 2018, Hong Kong, ISSN: 2078-0966. Available: http://www.iaeng.org/publication/IMECS2018/IMECS2018_pp140-145.pdf

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

C. N. Udanor. "Deploying an Open Source Cloud Computing Infrastructure for Academic Research." *IOSR Journal of Computer Engineering (IOSR-JCE)* 21.3 (2019): 61-74.