A Survey on Opensource Private Cloud Platforms

Mayur S. Patil
Dept. of Computer Engineering, MAEER’s MIT College of Engineering, Pune, India

Abstract
Unleashing and making the cloud platform as candid, open source private cloud is playing crucial role in this development of academic as well as industrial needs. Their significant focus is on sandbox environment for IaaS platform. But behind the scenes, presence of the powerful architecture has trivial role to handle bucket loads of data. The Open Stack, Nimbus, Eucalyptus and Open Nebula are deriving the power of integration successor with deploying environment making Private cloud trustworthy for user as well as developers. This survey paper endeavors readers with the current trends and collective overview of qualitative features of these platforms. It is also included OpenStack discussion with other respective cloud architectures. Alongside with this, architectural capabilities with oblations & the goals for Standard Cloud Architecture as well as future scope for further improvements are also discussed.

Keywords
Eucalyptus, Nimbus, Open Stack, Open Nebula, Survey, Private Cloud, Opensource

I. Introduction
What does an average user require? Mostly two things large storage and access speed with data security. Cloud computing is the paradigm brings user computing with single node to the worldwide distributed node to access resources along with HPC. Good example is social network i.e. data from node to cluster approach. It supports variety of applications and abstractions for SaaS on IaaS platform [3]. The era high speed of internet makes hardware resources to access the services at users’ fingertips by leveraging power of platform as well as hardware virtualization to manage scalability and optimality of resources. The sandbox approach [10] is becoming popular to provide VM on demand for the Users for direct use of current services without modifying the hardware. So, enterprises are offering their services through the cloud vendors for users. This is beneficial to Enterprises in terms of cost cutting, resource provisioning and risk management [4]. The main benefit in cost cutting is the Operational expense (OPEX) is always lesser compared to Capital expense (CAPEX). Larger the support for technologies, greater is the developers’ interaction, for example, OpenNebula. Lesser the support, restrain the developers capability such as OpenStack. Author has chosen four cloud platforms for survey as they are having great impact on Cloud computing field within short continuance. The Google trends result shows that search of these platforms over past 12 months. From the bar graph, we can see that there is small gap of research of these cloud platforms. The OpenNebula, brown bar, is at the Peak value while Nimbus, blue bar, and OpenStack, red bar, are the tantamount competent. Eucalyptus, green bar, has fallen to least searched term in the last year. These IaaS driven integration successor private cloud platforms has provided the users as well as industries with high level computing sources and hardware virtualization which makes them obligatory for research.

II. Related Work
While looking for previous forge, the most of the literature work is on either architectural aspects [1, 3-5] or feature sets [2,5] considering two clouds mostly Eucalyptus [10] and OpenNebula [12]. Few of these are focusing on the Scientific cloud Nimbus [11]. Author has included OpenStack considering factors such as free from Vendor Lock-in, Rapid pace of development, support for most of Linux Distros and Easy for learning to get results on the fly. All these projects are also attracting professional as well as student developers by means of events hackathon [13] or participating in the Google summer of code to take brilliant and hungry minds for code to get their hands dirty. The actual Cloud framework works, as a means of user environment, to handle critical components of the software stack which is helpful to understand flow of information in the system cloud system [1]. The abstract cloud architecture is made up of six components namely hardware, OS; network; VM hypervisor; archive of VM disk image; front end for users; cloud framework. It proposes architectural comparison but on the higher level with proposed challenges [4]. The inventors of Eucalyptus have proposed generic architecture of Eucalyptus with pros and cons such as VM instance scheduling, VM and user data storage, construction of virtual networks [3]. The feature in proposed version is referenced to the advantages and feature implementations with version specific structure [2]. Both architectural as well as feature set discussion on basis VM based implementations platform for above mentioned clouds helpful to provide quick understanding from user point of view [5].

III. Architectural Overview of Opensource Private Cloud Platforms
In this architectural overview, author is intentionally neglecting architectural workings. Aim of paper to get readers familiar with Private Cloud architectural components to understand what are the new & noteworthy features of recent Private Clouds because going deep in architecture might divert the reader’s attention on feature set.

Fig. 1: Open Source Private Cloud Platform in Google Trends (4th November 2011 to 4th November 2012)
A. Open Stack
OpenStack has dual platform for public and private clouds making cloud service omnipresent. Openstack has shown the speed of development in just two years is dazzling. It has API compatibility with Amazon EC2 with EBS service and S3 porting minimal efforts on cloud building.

Fig. 2: OpenStack Conceptual Architecture [9]

Up to the year 2011, the architecture of Openstack has only 3 components [6, 8] (talking in terms of Codenames) that are Nova (Compute): for virtual servers on demand, Glance (Image): VM image disk storage and Swift (Object storage): file distributed service. But in year 2012, there is dynamic increase in the cloud development by adding four more components namely Horizon (Dashboard): The UI is provided for developers as well as the end users, Quantum (Network): Interface Openstack users with external network, Cinder (Block storage): It manages data with Guest VM building on top of the Nova and Keystone (Identity): Authorization & authentication component in the Openstack [9]. Rackspace as well as NASA have collaborated to make the mono technical implementation which will help end user as well as developers to learn and deploy the Cloud infrastructures on their fingertips. Openstack has achieved the scalability as well as elasticity for big data users [6, 9]. The recent news is that Ubuntu 12.10 has support for Folsom (current 2012.2) release for ease of implementation to academic as well as industrial users.

B. Nimbus
It has main aim is availability of IaaS platform to scientific community. Its feature is to enable users as well as developers having the own implementations with IaaS private and community IaaS clouds [11].

Fig. 3: Architecture of Nimbus [11]

The main components are discussed as follows: The Workspace Service site manager is used to invoke different VM instance by using remote protocols. An EC2 front end is used to communicate with Nimbus clouds directly through API interface. Cumulus, successor over the GridFTP, is an open source implementation of the Amazon S3 [2, 4]. The RM API, deployed in Spring framework, useful to provide insight of remote and secure protocol of guest site. The cloud client uses the end-user interaction tool and manages their VM instances while reference client works as a means of command line tools. The VM manager is handled by using the workspace-control agent which based on Xen, KVM like implementations. The Context Broker helps to monitor Virtual clusters for ease of access to end users. To support and interact at boot time with this, the Context Agent lies on top of VMs [11].

C. Eucalyptus
The Eucalyptus framework for building the private Cloud has become the apple of developers as well as academician’s eye. Simplified architecture, popular interface like Amazon EC2 [15] for API communication, Walrus, an Amazon S3 like, storage performs handling of VM and user data effectively. Eucalyptus has spread like Wildfire at the time of its release due to its “only support” to Amazon Web Services (AWS) to build Private and Hybrid cloud. Then, canonical implementation, Ubuntu Enterprise Cloud (UEC), for Eucalyptus gives built-in support to it, making even newbies to install and utilize Eucalyptus in several steps [1, 5]. It tries to use existing virtual infrastructural by pulling services to manage the VM and related resources.

Fig. 4: Architecture of Eucalyptus [10]

The Components of Eucalyptus are Cloud Controller (CLC), Storage Controller (SC), Walrus, Cluster Controller (CC) and Node Controller (NC) [2, 10]. Isolated Web service is the aim of each component.

1. Cloud Controller
CLC is the heart of Cloud platform where user data and operations are performed over it. It has known to be the Entry-gate in the Cloud Service for developers, admin and users. CLC is useful for scheduling of VM and cluster controlling through Web services.
2. Walrus

The storage database of Eucalyptus saves data in terms of buckets and objects. It has two-way connections with user credentials database forwards to the Cloud controller. It is also useful to grant privileges to users either inside or outside of Cloud environment.

3. Cluster Controller

CC is the like manager access to several nodes i.e. users of particular region. It is midst way of CLC and NC for handling operation. Also handles load balancing on the node as well as the data belongs to each VM on separate node by managing the VM network.

4. Storage Controller

SC is, like Elastic Block Storage interface of Amazon, helpful to communicate with Walrus and CLC in either way. The concept of VM Snapshot is available in Eucalyptus with SC. Users are capable to create VM snapshot and store them in Walrus along the way. SC has capability to cope with various storage systems as well as network protocols but has access to particular data availability zone.

5. Node Controller

NC is the basic module of the Eucalyptus Cloud platform. It performs smart under VT support. It helps to manage VM instances, help to manage OS as well as Xen, KVM like hypervisors, which are continuously CC monitored [1, 3, 5, 10].

D. Open Nebula

It is the IaaS opensource standard by developing solutions for virtualized data centers and wide range of development based on Private, Public & Hybrid Cloud computing platform[1, 12].

Fig 5: OpenNebula Core Architecture [12]

This layer deals with the three components namely the command line interfaces, scheduler and other tools. The Command line interface is used to provide the operational interface for admin and end users for interacting Cloud services. Scheduler is independent entity to manage and monitor VM instances, of XML-RPC interface, load balancing and failure handling [4-5]. Other tools discuss the Haizea lease Manager [18] for additional scheduling activity. Next, the OpenNebula Core consists of authority of Network level as well as the Database level components. Host Manager has control and management over host machine to monitor guest machine (client) in the cloud. The VM manager used for process migration between two VM’s, aborting the schedule of VM etc. The Virtual Network manager use to grant/deny permissions of the virtual machines across the network-wide idea. SQL pool, to establish and fetch connection in the pool of databases without recreating each time [13], is used to access information. It is also free from any extra setup with add on of continued administration of ACID properties having storage in one Cross-platform disk file. OpenNebula manages the SQL pool having none of external dependencies. OpenNebula comes with drivers transfer i.e. to manage VM and its instances w.r.t. Cloud hosting [5, 12]. The VM drivers are required for efficient & smooth working of VM. Information driver used to transfer clients data (command) into actionable form on system level. This is the modular interface to interact with Middle level.

IV. Features

The features of the Private clouds make it tunable with user functionality. In previous papers [1-5], the features are discussed along with their downsides. Eucalyptus has more privileges to Admin by neglecting most of the User rights, making the Eucalyptus suitable for Enterprise environment than the Community while in OpenNebula, conversely, fighting with shared files and NFS issues it has pure private cloud implementation. It has provided users with great privileges making system easier to administer; assuming he is having technical emphasizing but lack of technical understanding might put user as well as his system into trouble. Next, it has attracting researchers to experimenting and fusing with other technologies, for ex, Nimbula [24]. In field of scientific computing, Nimbus is the prominent platform. Managing private as well as community platform, it is flexible to manage VM and related network. It also makes capacity allocation along with dealing of lease limits. Its main advantage is the ability to manage heavy data as well as security due to Globus credentials. Unlike OpenNebula, it specially built for purpose of scientific computing. In this study, OpenStack is now included.

The following features are based on improvements over disadvantages of predecessors that we have just seen.

A. Open Stack

In this discussion, author will present version table for descriptions illustrating the version development. Openstack has made plus point over others. The feature description is in accordance with components, in terms of code-names, of architecture. It has added several new features in Folsom release.

1. Nova (Compute)

Standing on 6th major release, it helps operators to configure VM pool and users with performance by CPU powering clouds, security using Trusted module to verify from existing cloud environment. Two new features are added; first in “config drive” the need of Globus credentials. Unlike Eucalyptus, it has pure private cloud implementation. It has provided users with great privileges making system easier to administer; assuming he is having technical emphasizing but lack of technical understanding might put user as well as his system into trouble. Next, it has attracting researchers to experimenting and fusing with other technologies, for ex, Nimbula [24]. In field of scientific computing, Nimbus is the prominent platform. Managing private as well as community platform, it is flexible to manage VM and related network. It also makes capacity allocation along with dealing of lease limits. Its main advantage is the ability to manage heavy data as well as security due to Globus credentials. Unlike OpenNebula, it specially built for purpose of scientific computing. In this study, OpenStack is now included.

2. Swift (Object Storage)

It is the 6th major release. Swift has achieved the Cluster management feature for operators using statsd server, a network daemon to gather metrics over UDP, with unique storage for user clusters makes it easier to tune cluster as well as hardware failures. For boosting the performance & high level access to metadata by cutting overhead of disk space, high write access clusters uses Solid-State Drives (SSDs), device uses solid-state memory for permanent data storage.
3. Glance (Image Service)
Standing on 5th major release, with Gear up performance & security, it has included new API, client library and replication options from client to his storage functionality.

4. Keystone (Identity)
The 2nd major release of OpenStack. Identity service makes the authentication process stronger such that none of the single component of Openstack architecture can commit operation without authorization of Identity services such as PKI.

5. Horizon (Dashboard)
The 2nd major release of OpenStack. Dashboard has improved Nova instances working, enhancing swift resources and controlling Openstack end users with cross-browser support like Firefox, error handling, support for private and public image uploading and advanced networking capabilities.

6. Quantum (Networking)
It is 1st major release. This helps users to manage their backend technology using plug-in architecture support. In added to this, this module is designed by taking into consideration for managing IP address, API and secured NAT management. API conjunction with policy framework gives the control of network on rent level to Enterprises & only admin level to service providers with aid of Identity service.

7. Cinder (Block Storage)
The 1st major release of OpenStack Block Storage, a subpart of Nova, now supports extensible range of volume continuing support for previously deployed volumes.

Table 1: Version Evolution Overview

<table>
<thead>
<tr>
<th>Version</th>
<th>Life starts</th>
<th>Introduction of New component</th>
<th>Minor release</th>
<th>Full (Major) Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin and Above</td>
<td>2010</td>
<td>Nova Swift</td>
<td>r10 r6</td>
<td>R6 R6</td>
</tr>
<tr>
<td>Essex</td>
<td>2011</td>
<td>Glance</td>
<td>r8</td>
<td>R5</td>
</tr>
<tr>
<td>Essex</td>
<td>2012</td>
<td>Keystone</td>
<td>r5</td>
<td>R2</td>
</tr>
<tr>
<td>Essex</td>
<td>2012</td>
<td>Horizon</td>
<td>r4</td>
<td>R2</td>
</tr>
<tr>
<td>Folsom</td>
<td>2012</td>
<td>Quantum</td>
<td>r1</td>
<td>R1</td>
</tr>
<tr>
<td>Folsom</td>
<td>2012</td>
<td>Cinder</td>
<td>r1</td>
<td>R1</td>
</tr>
</tbody>
</table>

B. Nimbus
The features included are VM enhancements, Security enhancement, Workspace services and component enhancements introduced in the recent version; which are seen as follows:

1. Storage Service Enhancement
   • Cumulus, an VM image storage of Nimbus for effective storage & transfer service, is compatible with S3 REST API clients, for managing VM storage, and against other AWS API such as boto (Python interface), s3cmd (command line tool), jets3t (JMX implementation). It has other implementation to calculate per user storage in terms of memory. Its nature is to constrain resources than cost.
   • CloudClient provides user friendly experience, even to newbie, Storage and IaaS cloud for authenticate users to access the workspace services through workspace client having Java reference implementation. It also tracks deployment time for each client, in which client can manage resource allocation for VM & used it during deployment, (used and remaining) which is helpful in authorization decisions for usage tracking. Similarly, clients can cross check for it.

2. Workspace Service Roles
   • Improved VM lifecycle and remote deployment (more similar in OpenNebula) - Nimbus clients can control and manage VM. On deployment, these features are included for users; VM metadata configuration, resource allocation and inspection of VM characteristic through WSRF polling [13].
   • Workspace service used to support two remote protocols; WSRF based for LTS (Long term Support) compatible with previous version as well as cloud-client and EC2 based for EC2 clients. Web services use Axis framework of Java. Main advantage of Workspace service is nothing constrained to web services, just like dependencies of Java applications, for remote protocols; only communicating system should have Java libraries.
   • Workspace service having network configurations, bridging exist address; allocate new network address from connection pool etc., provides client for VM accommodation of their taste with different NIC allocate with different pools from different addresses.
   • There are also provisions of mechanisms for operating deployment time configurations such as selection of physical host, network address management which include DHCP delivery network and specification on arbitrary file. It provides plugin to local resource manager for managing the pool of node with workspace-control, a lightweight Python script, also interacts with Xen. VMs used it as deployment model. Workspace-control maps bridging interface for networking requests, managing workspace instances with file isolations etc.

3. Security Enhancement
   • Supports X509 Credentials for purpose of strong security.
   • Users can assign identities to logical group includes, networking, VM images, resources, reserved time by the client, and write authorization policies for them. Additionally, it puts reservation limit, reservation, path & node checking considering precious usage of workspace having flexible authorization and authentication.

4. Other Enhancements
   • The One-click clusters is the functionality in which nodes of cluster are auto-configured themselves to adapt with new network environment. Nimbus allows cloud client to launch “one-click”clusters for managing entire cluster [1, 4].
   • Workspace-pilot, an alternative program to workspace service used to integrate set of VMM resources with site scheduler using local resource manager plugin, makes dual use of grid cluster; VMM node with no guest VM then regular jobs can be performed or VMM node when assigned to workspace service, Virtual Machine can be used.

5. To Return Nodes to Their Original State in Condition of Non-VM-Host
   • Protection against the workspace service not being available.
C. Eucalyptus

The Eucalyptus cloud author is considering here is the Community Edition. At the time of writing, latest version is in beta i.e. 3.1.2b. There are not any significant changes in feature as this version is a maintenance release. Support for Windows based VM instances. The only change is improvement to the NC, under heavy disk load, addresses various fault conditions of losetup, Libvrt, dmsetup and affect disk operation by external disk load [10].

D. Open Nebula

It has introduced the VM enhancements, Network Enhancement, Server enhancement and other components enhancements.

1. VM Enhancements

- Enhance VM development by Data and Transfer driver addition for storage solution for shared NFS without affecting disk space and VMFS (Virtual Machine File System) native support.
- It also provides solution to user unavailability, about misconfiguration in VMM or gateways of IP [1] as well as failure condition occurs, hence the Rollback mechanism is introduced. Whenever, VM state neither resumes nor stops, it returns to original state.
- The kernel cgroups on KVM used to restrict VM CPU usage as predefine in template while SPICE protocol support has also added.
- Poweroff functionality of VM has added so as to reboot VM in same host with new state.
- Command line support has added to create VM images.
- During VM transition, cleanup operation are supported to make resubmit of operations robust.
- From security point of view, the key pairs are included user template and injected same in VM instance through contextualization i.e. initialization phase to permit to set default value at time of template creation [13].

2. Network Enhancements

- Virtual Routers are used to cope with VN template to provide basic L3 services like DHCP, DNS.
- VNC (Virtual Network Computing) proxy system, previously Implemented in Essex version of OpenStack, uses to limit number of openports in OpenNebula. It has CLI administration capability, bridge between public-private network (where clients-vncserverlive respectively), token authentication and analyze hypervisor related connection details for enhancing client experience.

3. Server Enhancement

- Sunstone Custom routes can be added to the Sunstone server to ease the integration with third party tools.
- To communicate with EC2 Query Server[15], API for EC2 server has implemented with EBS interface to manage and EC2 instances. In addition to this, CLI tools are also improved named as ECONE.
- For OCCI Server, the OCCI API has improved implementation of new actions and hot plugging of volume to create volatile disk easily or import or an existing image from a DB to live VM.
- Purpose of this driver is to integrate newly created storage back-ends either after or before live migration.
- By setting data store on each different cluster having hypervisor specific deployment makes easier to work.

4. General Enhancements

- Without scan or parsing through user template, user related data (user name, group id and name) is easily available.
- Support for IE.

Table 2: Amazon AWS/API Compatibility w.r.t. Cloud Components

<table>
<thead>
<tr>
<th></th>
<th>OpenStack</th>
<th>Eucalyptus</th>
<th>Nimbus</th>
<th>OpenNebula</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC2</td>
<td>Nova</td>
<td>CLC</td>
<td>Workspace service</td>
<td>OpenNebula Core</td>
</tr>
<tr>
<td>S3</td>
<td>Glance</td>
<td>Walrus</td>
<td>Cumulus</td>
<td>-</td>
</tr>
<tr>
<td>EBS</td>
<td>Nova</td>
<td>SC</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

V. Mathematical Model

Mathematically, the problem can be expressed as follows:

Let \( P \) is a cloud instances with finite set of resources defined by the following relation

\[
P = (c_i, m, r, d, l)
\]

where \( c_i \)= CPU resources

\( m = Memory resources \)

\( r = MapReduce framework \)

\( d = Datasets \)

\( l = load on node of Cloud instance \)

Let \( d = (d_1, d_2, ..., d_k) \) be number of datasets (with definite capacity) handled by one Cloud instance \( P \) and \( r = (r_1, r_2, ..., r_m) \) be MapReduce framework implementation on one Cloud to manage datasets having CPU resources in terms of CPU capacity and the memory capacity.

\[
C = (C_{cpu}, C_{ram})
\]

By using mathematical induction,

\[
1+2+3+...+p = \frac{p(p+1)}{2}
\]

It states that if MapReduce framework can handle one cloud instance then it can also handle \( \frac{p(p+1)}{2} \) clouds instances efficiently. Now, Dataset is handled by Cloud instances \( P \) allocating resources for managing datasets, so equation becomes as

\[
P = (\sum C \cup d)
\]

It tells that each instance of cloud with its dataset requires the specific amount of resources. But when load is increases on each instance, chances of failure might be there. So equation becomes unbalanced.

\[
P \neq (\sum C \cup d) + 1
\]

It makes exhausting for instance alone to store, transfer data as well as to manage load on it. So, if there we implemented map reduce framework, it will handle load as well as resources on the cloud instances. Now the equation is

\[
M(P) = (\sum C \cup d) + 1
\]

Where \( M(P) \) is MapReduce implementation for Private Cloud platform. This is useful to calculate load balancing and resource utilization on cloud instance of one cloud platform.
VI. Conclusion
While ascribing this survey on open source Private cloud frameworks and their features, we can see that their way of building describes the current trends of customers and developers implementation. Current models are more oriented towards the Virtualized enterprise data management. From the above survey, we can say Open Nebula is the most suitable for development purpose. While, OpenStack has shown rapid on fly development in just two years. By considering challenges, load balancing and data scaling are challenges for Private Cloud platform due to their constrained design. Web Application development and deployment is possible due to wide range of API. Programming environments of Cloud paradigm structure decides range of developers’ interaction; hence the development. On the other hand, it puts architecture Amalgamation, due to direct inclusion of technologies without deducing downside, which makes architecture complex and difficult chances of rising security issues. Author has presented tables for quick and easy understanding of these platforms as a summary at the end.

VII. Future Scope
As data on the Cloud services increasing day by day, it makes tiresome and clumsy job to manage and handle that data which is even in Petabytes to Service Providers. Though auto-scaling option is successful at somewhat extent but processing power consumed a lot. Map-Reduce [13], Google proposed technology, implementations in Cloud. It uses distributed computing approach to process large data sets. So frameworks, can be used to process, scale and manage the clustered data, like CloudMapReduce (AWS implementation) [19], Apache Hadoop [14], Disco [20], “Sector [21], DryadLINQ (Microsoft implementation) [22] and, are proved to be useful” it should be as ” Sector [21] and DryadLINQ (Microsoft implementation) [22] are proved to be useful”.

Table 3: Cloud Platforms Summary

<table>
<thead>
<tr>
<th>Focus</th>
<th>Openstack</th>
<th>Nimbus</th>
<th>Eucalyptus</th>
<th>OpenNebula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop Ubiquitous i.e. available at everywhere platform</td>
<td>To provide platform for Scientific computing</td>
<td>Develop AWS Compatible clouds</td>
<td>Develop solutions for Virtualized enterprise infrastructure</td>
<td></td>
</tr>
<tr>
<td>Technology</td>
<td>Python</td>
<td>Java(for Front end), Python (for Back end)</td>
<td>Java, C</td>
<td>Ruby:23%, C++:39%, javascript: 20% mainly.</td>
</tr>
<tr>
<td>Database</td>
<td>MySQL, Postgresql</td>
<td>Hsql</td>
<td>SQLite 3</td>
<td></td>
</tr>
<tr>
<td>Components</td>
<td>Nova, Glance, Keystone, Cinder, Horizon, Quantum, Swift.</td>
<td>CLC, SC, CC, Walrus, NC.</td>
<td>Frontend, host, Image Repository, Physical network</td>
<td></td>
</tr>
<tr>
<td>Suitability</td>
<td>Easy development and quick started for Python people</td>
<td>Great platform for scientific standards</td>
<td>Easier architecture for Academic knowledge of Cloud</td>
<td>Wide range of support of languages for developers and users.</td>
</tr>
<tr>
<td>Cloud Characters</td>
<td>Private and Public</td>
<td>Private And Community</td>
<td>Private and Hybrid</td>
<td>Private, Public and Hybrid</td>
</tr>
</tbody>
</table>

References
Mayur S. Patil has received B.E. in Computer Science and Engineering in year 2011. He is currently pursuing M.E. in Computer Engineering at MAEER's MIT College of Engineering, Pune University. His areas of interest are Cloud Computing, Data mining, Operating system and Programming. He is currently working on project of security aspects in Eucalyptus Private Cloud.