5. Implementation

Implementation of the trust model requires first preparing a test bed. It is a cloud computing environment that is required as the first step towards the implementation. Various tools are used as a part of implementation to check their feasibility. A comparative study shows the various parameters are drawn and analyzed during the implementation that can be used as the basis for development of cloud by using any of these tools. Based on the tools for cloud environment trust model is implemented as a .Net service and observations are recorded.

5.1 Creation of Test Bed

Testing before actual implementation of cloud is becoming must for any organization providing cloud services. It reduces the risks associated with the actual deployment. Various cloud computing tools and simulation environment is available to give hands on experience. Cloud setup is implemented by using some of them. A comparative analysis of these tools is presented. Various parameters are identified to perform the comparisons. Any user can make use of this study to select a particular tool depending on his needs and type of experiment.

1) CloudSim

CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments as given in [Rodrigo N et al, 2011]. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It is framework for modeling and simulation of cloud computing infrastructure and services. The main advantage of using CloudSim for initial performance testing includes:

1. Time effectiveness: It requires very less effort and time to implement Cloud-based application provisioning test environment and

2. Flexibility and applicability: Developers can model and test the performance of their application services in heterogeneous Cloud environments (Amazon EC2, Microsoft Azure) with little programming and deployment effort.
2) Xen Hypervisor

The Xen hypervisor is a layer of software running directly on computer hardware replacing the operating system thereby allowing the computer hardware to run multiple guest operating systems concurrently. It’s support a variety of platform and other common operating systems as guests running on the hypervisor [Xen, 2011]. By using this virtual environment a cloud setup can be configured.

3) UEC

Ubuntu Enterprise Cloud (UEC) is a stack of applications from Canonical included with Ubuntu Server Edition. UEC includes Eucalyptus along with a number of other open source software. UEC makes it very easy to install and configure the cloud. Canonical also provides commercial technical support for UEC.

UEC consist of various components: Node Controller, Cloud Controller, Cluster Controller, Walrus Storage Controller and Storage Controller. The details of the individual components [UEC, 2011] are summarized as follows:

1. Node Controller (NC): A UEC node is a VT (Virtual Technology) enabled server capable of running KVM as the hypervisor. UEC automatically installs KVM when the user chooses to install the UEC node. The VMs running on the hypervisor and controlled by UEC are called instances. Node Controller runs on each node and controls the life cycle of instances running on the node. The NC interacts with the OS and the hypervisor running on the node on one side and the Cluster Controller (CC) on the other side. NC queries the Operating System running on the node to discover the node's physical resources like number of cores, the size of memory, the available disk space and also to learn about the state of VM instances running on the node and propagates this data up to the CC.

2. Cluster Controller (CC): CC manages one or more Node Controllers and deploys/manages instances on them. CC also manages the networking for the instances running on the Nodes under certain types of networking modes of Eucalyptus. CC communicates with Cloud Controller (CLC) on one side and NCs on the other side.
3. **Walrus Storage Controller (WSC):** WSC provides a persistent simple storage service using REST and SOAP APIs compatible with S3 APIs. WSC should be considered as a simple file storage system.

4. **Storage Controller (SC):** SC provides persistent block storage for use by the instances.

5. **Cloud Controller (CLC):** The Cloud Controller (CLC) is the front end to the entire cloud infrastructure. CLC provides an EC2/S3 compliant web services interface to the client tools on one side and interacts with the rest of the components of the Eucalyptus infrastructure on the other side. CLC also provides a web interface to users for managing certain aspects of the UEC infrastructure. CLC has a comprehensive knowledge of the availability and usage of resources in the cloud and the state of the cloud.

A private cloud setup can be configured and instances are registered and made available for use by various cloud users as per their requirements.

4) **OpenStack**

It is a collection of open source technology that provides massively scalable open source cloud computing software as mentioned in [Anuj, 2012]. Currently OpenStack develops two related projects: OpenStack Compute, which offers computing power through virtual machine and network management, and OpenStack Object Storage which is software for redundant, scalable object storage capacity. Closely related to the OpenStack Compute project is the Image Service project, named Glance. OpenStack can be used by corporations, service providers, SMBs (Small to Medium Business), researchers, and global data centers looking to deploy large-scale cloud deployments for private or public clouds. There are 3 main service families under OpenStack:

- Compute Infrastructure (Nova)
- Storage Infrastructure (Swift)
- Imaging Service (Glance)

A cloud setup by using openstack can be configured using single machine using its manual.
5.1.1 Comparative Analysis of Cloud Simulation Tools

Cloud computing simulation tools are available with various specifications. Some of them have used to implement private cloud setup for experimentations. Lists of parameters are identified based on which comparison between various tools can made as given in the following table 5.1. The parameters are described as follows.

1. **Time/popularity/newness:** A cloud software will goes for modifications if the user needs and demands increases. Timely updation of the versions is based on its usage and hence popularity. More flexible software, more its demands and therefore accordingly requires modifications or up-gradations.

2. **Version:** Version indicates various copies of the same tool with additional features. Any cloud tool comes with basic version first. It includes limited functionalities that should increase or upgraded with time. The latest version gives the major functionality and features that are needed by the user.

3. **API:** A cloud tools comes with set of API to provide functionalities. More number of APIs is desirable to have large number of operations that are supported by a cloud tools needed by the user.

4. **Main Support:** Mostly cloud tools come with support for specific cloud provider services. So depending on the user requirements a particular cloud tool is selected.

5. **Scalability:** It is the ability of a system, network, or process to handle a growing amount of work load. A cloud tool used for experimentation should be scalable to support large number of applications and user.

6. **Openness:** This property is useful to enhance or update the existing tools as per the research demand by a user. So a tool that has freely available APIs is desirable to perform research and experiments.

7. **Implementation:** A cloud computing environment provided by tools should be easy to configure and implement. Sufficient documentation along with commands and forum support is expected by various user who want to start experimentation with cloud computing.
8. **Resources:** A cloud computing setup may require high end machines for their installation. Therefore user can select particular tools depending on the availability of resource.

9. **Community:** There are three types of community that participate any application development. Users, Developer and Deployers are the communities that get opportunity in a cloud application development. A cloud tool is also classified based on the type of community for which it is useful and creates opportunity.

10. **Lock-in:** Vendor lock-in issues makes restriction with respect to data or application. Lock-in issues should also be considered while selection a cloud tool for experimentation.

11. **Web Service Support:** Many cloud computing tools supported by web services and selected accordingly.

Following table illustrates various tools that are compared based on parameters listed.

Table 5.1: Comparison of cloud implementation

<table>
<thead>
<tr>
<th>Sr.No.</th>
<th>Parameter</th>
<th>Eucalyptus</th>
<th>Openstack</th>
<th>Xen Cloud</th>
<th>CloudSim</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Time/popularity/</td>
<td>Basic version of Ubuntu eucalyptus</td>
<td>Advance version of Ubuntu comes</td>
<td>Very basic and exist before cloud also</td>
<td>The first version is of the year 2009 and upgrading versions are evolved further</td>
</tr>
<tr>
<td></td>
<td>newness</td>
<td>cloud is the first IaaS cloud</td>
<td>along with the openstack cloud</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Version</td>
<td>All the versions of Ubuntu possess</td>
<td>11.10 version of Ubuntu onwards</td>
<td>Xen comes along with the Linux</td>
<td>The latest version of it is 3.0.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>eucalyptus cloud features</td>
<td>openstack cloud option is made</td>
<td>distribution of which latest version is 4.2.2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>API</td>
<td>Small set of API</td>
<td>More API providing as a backbone</td>
<td>Sufficient set of API to perform basic</td>
<td>Large set of classes that can be modified to suit user requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>for building private and public</td>
<td>operation</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Main Support</td>
<td>Amazon EC2 and S3</td>
<td>Suitable for all cloud applications</td>
<td>Most preferred virtualized</td>
<td>Simple simulation that can used independently along with java and C programming</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>environment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Scalability</td>
<td>Used for small scale application</td>
<td>Scalability issue is solved</td>
<td>Scalable as per the resources</td>
<td>It’s a simulation kit give good performance with increased users also</td>
</tr>
<tr>
<td>---</td>
<td>---------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------</td>
<td>------------------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>6</td>
<td>Openness</td>
<td>Not completely open. Certain features like VMware etc cannot be modified</td>
<td>Fully open Source Software</td>
<td>Not much modifications can be possible</td>
<td>Fully open source and can be modified completely</td>
</tr>
<tr>
<td>7</td>
<td>Implementation</td>
<td>Simple and less number of commands and instructions set is required</td>
<td>Complex command and large set of instructions require for implementation</td>
<td>Setting of Xen Cloud Platform is easy and fast</td>
<td>Compare to others fewest number of instructions are required</td>
</tr>
<tr>
<td>8</td>
<td>Resources (high configuration systems)</td>
<td>Testing and implementation requires large set of resources</td>
<td>Small set of resources is sufficient to play with it</td>
<td>A dedicated xen server is must.</td>
<td>A simple desktop machine is sufficient</td>
</tr>
<tr>
<td>9</td>
<td>Community</td>
<td>Limited support for developer, deployer and user</td>
<td>Unlimited opportunity for community</td>
<td>Limited support</td>
<td>Unlimited support for testing</td>
</tr>
<tr>
<td>10</td>
<td>Lock-in</td>
<td>Cloud provider lock-in is not much prevented</td>
<td>Completely prevents the lock-in problem</td>
<td>Not much prevented</td>
<td>Prevents lock in to Some extent</td>
</tr>
<tr>
<td>11</td>
<td>Web Service Support</td>
<td>Not supported by any web services</td>
<td>Support provided by AWS (Amazon web services)</td>
<td>No support</td>
<td>Can support to some extent</td>
</tr>
</tbody>
</table>

Parameters can be act like a checklist to consider specific implementation tool for specific service. Based on the choice of the experiment needed for a particular service any of tools can be used. Cloud using openstack cloud architecture is implemented and configured the resources that are required for implementing trust model.

### 5.2 Trust Model Implementation

Implementing the trust model requires implementing the various modules. One of the modules is cloud service manager that acts a repository of trust values associated with a cloud service. Figure 5.1 indicate the snapshot of a cloud service manager.
Figure 5.1: Cloud Service Manager

It uses trust model to evaluate the trust at any time. The basic module which actually calculates the security strength is the trust model. Trust model consists of various parameters. Cloud service trust can be calculated using the Trust model. Trust model implementation requires implementing various parameters. Various modules and parameters are implemented. Various options can be used to give input for a cloud service whose trust value needs to be measured. Various cloud services with their features are observed and noted the results for various parameters. APIs are designed for measuring specific functions and to calculate the trust value. Parameters Identity management system and Authentication are discussed below.

1) **Identity Management Parameter**

Trust model with respect to identity management is formulated using the .net framework version 4 and trust values with respect to various service specifications are calculated. Following figure 5.2 and 5.3 shows the snapshot of the IDM parameter design page.
Figure 5.2: Identity Management Parameter – Web page

Figure 5.3: IDM Parameter web page-(continued)
2) **Authentication**

Another parameter that is implemented in .net framework is Authentication. Following figure 5.4 and 5.5 shows the output in the browser window.

![Authentication Parameter](image)

*Figure 5.4: Authentication Parameter - web page*
The code to perform calculation is written in .cs file of the .net framework. A web service is created that can take the values as the input for various sub parameters and functions value chosen. The code of web service file is also given below that calculated strength value of IDM and password given in appendix.

All the parameters and modules are implemented and values for the various services are recorded using private cloud setup.