Bachelor's Thesis (UAS)

Degree Program in Information Technology

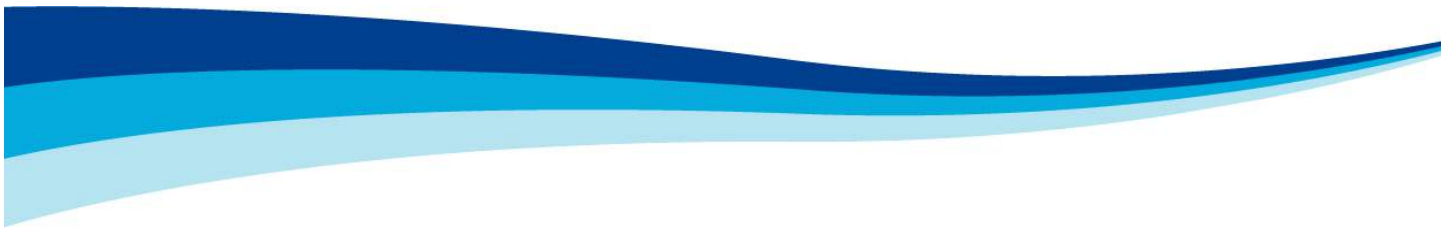
Internet Technology

2013

Odunayo O. Owopetu

Private Cloud Implementation and Security

Using EUCALYPTUS and XEN Frameworks



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Information Technology

March 2013 | 59

Instructor: Ossi Väänänen

Odunayo O. Owopetu

Abstract

Private Cloud Implementation and Security

Private cloud is a new improved way to organize and manage IT resources and services within an enterprise. This can be achieved by establishing a private cloud framework behind the corporate firewall of an enterprise. This is done in order to promote better efficiency in determining workload, usage priorities, and security threats in an enterprise as well as secure sensitive company information, which other cloud offerings may not always effectively protect.

The first part of this work is basically theoretical and explains the cloud computing concept as well as relay general information about the cloud architecture, deployment and delivery models of the cloud which was well linked to give a good understanding of the thesis work while the second part is mainly practical. The practical part describes the laboratory implementation of a private cloud.

Linux-Ubuntu Enterprise Cloud powered by Eucalyptus was used in this project as the preferred underlying technology upon which a private cloud infrastructure was built. Eucalyptus (management software) is used to create a scalable, secure web service layer that abstracts and stores data. These technologies are critically discussed and their implementation demonstrated in the later parts of this thesis. This thesis also explains step-by-step, the easiest way to deploy a private cloud in an enterprise and as such provides information on how to implement a private cloud and possible ways of effectively securing it.

KEYWORDS:

Cloud Computing, Eucalyptus, Firewall, Linux- Ubuntu, Private Cloud

ACKNOWLEDGEMENTS

I have given myself the opportunity to wish and dream throughout my life. I have reached some of my professional goals, not only because of my innate abilities, but because I have had the opportunity of meeting wonderful human beings that have contributed to my life with knowledge, words of support, and motivation.

First of all I thank you God, for life, health, and the energy that you have given me to reach some of my professional goals. My sincere gratitude goes to Kalliopi Skarli for being there for me through thick and thin and for being more than just a teacher to me all through the years. Special thank you also goes to my supervisor, Ossi Väänänen, for his patience and understanding and for his words of encouragement all through this thesis work. I would also like to appreciate Tero Virtanen for his assistance and guidance with the practical aspect of this work as well as Vesa Slotte for always encouraging me to work hard and never to give up.

To all other lectures that have taken the time and patience to impact knowledge on me in one way or another, I say a very big thank you. I also appreciate my parents back home for their kindness and support as well as my family members for always being there for me, and to my friends and colleagues, I say thanks and God bless.

TABLE OF CONTENTS

Abstract

Acknowledgment

Acronyms and Abbreviations

1 **Introduction to Cloud Computing** 9

Aim and Objective 9

Organization and Structure 10

2 **Cloud Definition** 11

2.1 Background 12

2.2 Cloud Evolution 13

2.3 How the Cloud works 15

3 **Cloud Architecture** 17

3.1 Generic Cloud Architecture 17

3.2 Private Cloud Architecture 20

3.3 Eucalyptus Private Cloud 21

4 **Cloud Deployment Models** 23

4.1 Public Cloud 24

4.2 Private Cloud 25

4.3 Community Cloud 26

4.4 Hybrid Cloud 28

5 **Cloud Delivery Models** 29

5.1 Cloud Infrastructure Systems 30

5.2 Cloud Platform Systems 31

5.3 Cloud Software Systems 32

6 **Securing a Private Cloud** 34

7 **Private Cloud Implementation** 39

7.1 Front End Setup 39

7.2 Back End Setup 41

7.3 Eucalyptus Tools Setup 42

7.4 Troubleshooting 43

8 **Conclusion** 45

References 46

Appendices

10.1 APPENDIX 1: FRONT END SETUP 50

10.2 APPENDIX 2: BACK END SETUP 51

10.3 APPENDIX 3: EUCALYPTUS TOOLS SETUP 53

10.4 APPENDIX 4: TROUBLESHOOT 54

Figures

Figure 1.0: Evolution of Cloud Computing (from the Indian Khaleej times). 15

Figure 2.0: Cloud Computing Architecture (from Jim Kaskade, January 24, 2009). 19

Figure 3.1: On/Offsite Private Cloud Architecture (Synergy Global Solutions). 20

Figure 3.2: Eucalyptus Cloud Components (from OmniGroup's OmniGraffle and inkscape). 22

Figure 4.0: Cloud Deployment Model (from the Slovak Bucov blog). 23

Figure 4.1: Community Cloud Architecture (from the Zurich IT Wissen on line blog). 27

Figure 5.0: Delivery Model Types (from Sam Johnston, 3rd march, 2009). 29

Figure 6.0: Software as a Service End User Application (from the big foot retail solutions). 33

Figure 7.0: Ubuntu Cloud Installation (from information week, 1st December, 2012). 40

Figure 7.1: Ubuntu Management Software (from information week, 1st December, 2012). 43

Figure 7.2: Ubuntu List of Available Images (from information week, 1st December, 2012). 44

Screenshots and Images

Image 1 Ubuntu Installation Screen 55
Image 2 Ubuntu language selection page 55
Image 3 Ubuntu Network Configuration 56
Image 4 Disk Partition Prompt 56
Image 5 User Account Creation 57
Image 6 Password Creation 57
Image 7 Proxy server Installation 58
Image 8 Software Installation Selection 58
Image 9 Grub Boot Loader Installation 59
Image 10 Installation Complete Page 59

ACRONYMS AND ABBREVIATIONS

AWS:	Amazon Web Service
AMD-V / VT-X:	Advanced Micro Dynamics Virtualization / Virtualization Technology Intel
API:	Application Programming Interface
BIOS:	Basic Input/ Output Operating System
CC:	Cluster Controller
CLC:	Cloud Controller
CPU:	Central Processing Unit
DHCP:	Dynamic Host Configuration Protocol
EC2:	Elastic Computing Cloud
EMI:	Eucalyptus Machine Image(s)
EBS:	Elastic Block Storage
EUCALYPTUS:	Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems
IAAS:	Infrastructure as a Service
ISCSI:	Internet Small Computer System Interface
KVM:	Kernel-based Virtual Machine
NIST:	National Institute of Standards and Technology
NFS:	Network File System, protocol
OS:	Operating System
PAAS:	Platform as a Service
S3:	Amazon Simple Storage Service
SAAS:	Software as a Service

SSH:	Secure Shell
TFTP:	Trivial File Transfer Protocol
VM:	Virtual Machine
UEC:	Ubuntu Enterprise Cloud
WS3:	Walrus Simple Storage Service

1. Introduction to Cloud Computing

Cloud computing is a modern day trend in the computing industry in which elastic and scalable IT-enabled proficiencies are conveyed as a service to customers via the internet. It is a computing paradigm, where a large pool of systems is connected in public or private networks, to provide infrastructures that are dynamically scalable, for applications, data and file storage systems. With the emergence of this technology, the computation cost, content storage, application hosting, and delivery is significantly reduced. Cloud computing is a practical way to undergo direct cost benefits and it has the potential of transforming an enterprise data centre from a capital intensive setup to a capital saving environment. The ideology behind cloud computing is based on the fundamental principle of virtualization ("*reusability of IT capabilities*").

"*There are no rules of architecture for a castle in the clouds.*" —G.K. Chesterton

Cloud computing services can be delivered by an internal IT organization (in-sourced) or by an external service provider (outsourced). The basic infrastructure can be hosted within an enterprise's data centre or in an external data centre. That basic infrastructure can be dedicated to a single customer (private cloud), jointly used between many customers (community cloud) or shared with a service provider's customer base in general (public cloud). The deployment of a private cloud in an enterprise is essential since it is setup behind the corporate firewall of an enterprise and thus ensures improved security, low end-user support costs, and more efficient management can all help improve the effectiveness and value of the enterprise.

This technology does not only reduce IT costs, it also accelerates innovation within enterprises. This is the impetus or drive behind the implementation of a Private cloud. Network virtualization is the bedrock for this solution because it can consolidate diverse networks into a single virtual entity, the first step in creating a service-oriented infrastructure. Based on this, IT resources can be scaled up and down virtually to provision on-demand services (a.k.a. private cloud services) without the addition of new physical devices or entities through server virtualization. This enables more agile balancing of expense reduction with business growth initiatives congestion, and bandwidth limitations.

Aim and Objective

The aim of this thesis is to demonstrate a simple way to build a private cloud for an enterprise using open source software. This is accomplished using the Ubuntu Enterprise Cloud (UEC) (which has the same machine image as Amazon EC2) which is highly scalable, readily deployable, easily customizable, lean, fast, and is implemented using Eucalyptus.

Organization and Structure

This thesis is divided into two sections: the first part is the theoretical part which deals with the definitions, structure and architectures involved in cloud computing as well as the delivery and deployment models while the second part is the practical part which illustrates the deployment of Eucalyptus and the implementation of the private cloud. The chapters are summarized as follows:

1.2.1 Chapter One – Introduction to Cloud Computing: This is the beginning of the theoretical aspect of this work and this first chapter introduces the cloud computing concept and also explains the motive behind doing this work. It answers the question: Why use the cloud? It also gives a brief description of the entire thesis work.

1.2.2 Chapter Two – Cloud Definition: This chapter gives hindsight into the evolution of the cloud and further explains how the cloud works.

1.2.3 Chapter Three -- Cloud Architecture: This chapter describes a generic cloud design and its architecture. It also justifies the reasons for implementing a private cloud using Linux Ubuntu powered by Eucalyptus. In addition, this chapter analyzes the components of a eucalyptus private cloud.

1.2.4 Chapter Four – Delivery Models: This chapter explains the kinds of services that the cloud offers which are referred to as the SPI model. SPI stands for Software, Platform, and Infrastructure as a service. This project will be explaining the Infrastructure as a service delivery model.

1.2.5 Chapter Five – Securing a Private Cloud: This chapter introduces security concerns that arises with implementing a private cloud, suggests ways to reduce these security threats to the barest minimum, and discusses best practices to be undertaken by enterprises to face these challenges

1.2.6 Chapter Six – Deployment Models: This chapter explains the various types of clouds that exists including Public, Private, Community and Hybrid clouds and also explains which type best fits an enterprise as well as features of the clouds and the advantages of using them.

1.2.7 Chapter Seven – Private Cloud Implementation: This chapter illustrates the practical section of this work and it describes the process involved in building a private cloud and shows in detail the front end, back end and the Eucalyptus tools setup as well as a few troubleshooting tips.

1.2.8 Chapter Eight and Nine – Conclusion and References: This is the concluding chapter of this work and it summarizes the whole thesis and also lists used references as well as a list of appendices.

2. **Cloud Definition**

The task of defining cloud computing is unfortunately not easy as everyone has a different definition of cloud computing. Virtually anyone with an interest in information technology has a one. In “A Break in the Clouds: Towards a Cloud Definition”, a white paper published for the ACM Computer Communication Reviews, the authors found over twenty distinct definitions of cloud computing in their research (“Vaquero et al., 2009”).

They assembled some of the main notions into: “A large pool of easily usable and accessible virtualized resources (hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the infrastructure provider by means of customized SLAs” (“Vaquero et al., 2009”).

Most definitions of cloud computing include elements of the complete description and yet typically do not address every single aspect that anyone has associated with cloud computing. The most important consideration for an enterprise is not whether a potential solution satisfies the definition of cloud computing but rather whether it adds value to business. A cloud based solution that does not increase revenue or decrease costs is of little interest. And a completely non-cloud oriented solution that does unambiguously improve the bottom line should be implemented regardless of the name by which it is called.

The key to understanding common interpretations of the term cloud computing is to examine the assortment of attributes of typical cloud solutions. This does not mean that every cloud attribute is essential to cloud computing or even that there is necessarily any which qualifies a given approach as fitting the cloud paradigm. On their own, cloud solutions are neither necessary nor sufficient prerequisites to the motion of cloud computing. But typically the more of these attributes apply the more likely others will accept it as a cloud solution.

The definition of cloud computing is based on five distinct attributes which includes: multi-tenancy (shared resources), massive scalability, elasticity, pay as you go, and self-provisioning of resources. These attributes of cloud computing will be further explained as follows:

Multi tenancy (shared resources): Unlike previous computing models, which assumed dedicated resources (i.e., computing facilities dedicated to a single user or owner), cloud computing is based on a business model in which resources are shared (i.e., multiple users use the same resources) at the network level, host level, and application level at the same time.

Massive scalability: Although enterprises might have hundreds or thousands of systems, cloud computing provides the ability to scale to tens of thousands of systems, as well as the ability to massively scale bandwidth and storage space.

Elasticity: Users can rapidly increase and decrease their computing resources as needed, as well as release resources for other uses when they are no longer required.

Pay as you go: Users pay for only the resources they actually use and for only the time they require them.

Self provisioning of resources: Users self provision resources, such as additional systems (processing capability, software, storage) and network resources.

One of the attributes of cloud computing is elasticity of resources. This cloud capability allows users to increase and decrease their computing resources as needed. There is always an awareness of the baseline of computing resources, but predicting future needs is difficult, especially when demands are constantly changing. Cloud computing can offer a means to provide IT resources on demand and address spikes in usage.

Interest in the cloud is growing because cloud solutions provide users with access to supercomputer-like power at a fraction of the cost of buying such is a solution outright. More importantly, these solutions can be acquired on demand; the network becomes the supercomputer in the cloud where users can buy what they need when they need it. Cloud computing identifies where scalable IT-enabled capabilities are delivered as a service to customers using internet technologies.

2.1 **Background**

The pursuit of science has evolved over hundreds of years from the development of the scientific method to the use of empirical methods. This evolution continues today at an increasingly rapid pace. Scientific pursuit has always been marked by advances in technology. The evolution of cloud computing continues today as well. The 1970s and 1980s saw the significant development of computational power. Computer simulations were considered a third pillar of science in addition to theory and experiment.

In the 1990s, scientists were beginning to develop and rely heavily on the internet communication technologies. The National Centre for Biotechnology Information's BLAST server provided scientists with an early sequence alignment tool that made use of remote computing potentiality. Queries and results were exchanged through email. This service was later made available on the internet and continues to operate today.

Previously, supercomputers were accessed by a small number of users who were members of elite research groups. Tera Grid science gateway program was initiated in 2003 and they helped recognised that the impact of high-end resources could be greatly increased if they could be coupled onto back end existing web portals being developed prolifically by scientists.

Technology continues to evolve with increasing rapidity. Cloud computing and similar technologies are examples of technologies which have virtualized access to high-end resources that enable high throughput computing, which makes the rigorous computations possible. The high-end resource which is required by the back end systems is what the Internet is based upon. This sees to it that back end systems could perform multiple tasks as well as be used by multiple individuals all at the same time.

With the evolution of the internet comes the development of the operating system, isolated computers for the first time could simultaneously perform multiple functions. In earlier times, many computer users sharing a single computing device that is sometimes powerful were known as local area networks.

This enabled the first set of systems with multiple users at the same time. Under these conditions, a stand-alone central server or computing system device supported many dumb terminals (keyboards and computer screens) or personal computers contained in same physical location. The actual processing is done by the terminals after connecting to the central server.

Cloud Computing evolved from these previous shared computing efforts. As high-speed internet connections came into being, and the cost of processing power and storage have drastically reduced, cloud computing has gained the interests of many businesses and individuals. As at the end of September 2008, up to 69 percent of internet users were storing data online, making use of webmail services, or otherwise using some kind of software programs; which could be word processing applications; which have their functionality located on the web.

2.2 **Cloud Evolution**

The evolution of cloud computing is a major change in computing technology. One of the most important parts of that evolution is the advent of the first production platforms based on the cloud paradigm. Such platforms promise real gains in terms of performance scalability and agility to their users. By leveraging cloud computing, an enterprise can rapidly deploy applications where the underlying technological components can expand and contract with the natural ebb and flow of the involved business life cycle. Traditionally, once an application was deployed, it was bound to a particular infrastructure until the infrastructure was upgraded / improved.

The result was low efficiency, poor utilization, and limited flexibility. Cloud enablers such as virtualization and grid computing allow applications to be dynamically deployed onto the most suitable infrastructure at runtime. Cloud computing takes these concepts further, by allowing more automated resources and workload management practices. This elastic aspect of cloud computing allows applications to scale and grow without needing traditional upgrades. Like any new paradigm, cloud computing represents an architectural shift from the traditional distributed computing approaches. Such a shift is best described by the addition of a new and as transparent as possible middleware layer on top of the existing computer and device operating systems that we can call a cloud computing system.

It can be considered as a network operating system running atop a cloud, i.e., a hyper network of computers. As its name suggests, this kind of runtime platform lets users write applications that run in the cloud, or to use services provided by the cloud, or both. But the transformation that cloud computing makes possible goes beyond simply running applications on a virtualized platform built on someone else's hardware. It extends the computing model with the transparent utilization of a platform that the provider has created, and which, to some extent, abstracts the essence of scalability and distributed processing.

More generally, the concept of cloud computing can incorporate various computer technologies including Web 2.0, and many other emerging technologies. Users may have different perspectives and views about it. For example, from the perspective of an end user, the cloud computing service moves the application software and operating system from desktops to the cloud side, which enables users to plug-in anytime from anywhere and utilize large scale storage and computing resources. On the other hand, the cloud computing service provider may focus on how to distribute and schedule the computer resources. Nevertheless, storage and computing on massive data are the key issues for a cloud infrastructure.

Cloud computing dates back to a time when computer systems were remotely time-shared computing resources and applications. In modern times, cloud computing refers to the many different types of services and applications that are rendered over the cloud, and most times, the devices used to access these services do not require any special applications.

Cloud computing evolved from parallel, utility, autonomic, distributed, and grid computing. There is a slight similarity between all these technologies but they all work differently. Computing can be simply explained as the sharing and use of resources and applications of a networked environment to complete a task without concern about ownership or management of the network's resources and applications. The image below describes the evolution of cloud computing since the advent of the internet up till now. This is displayed in Fig. 1, below.

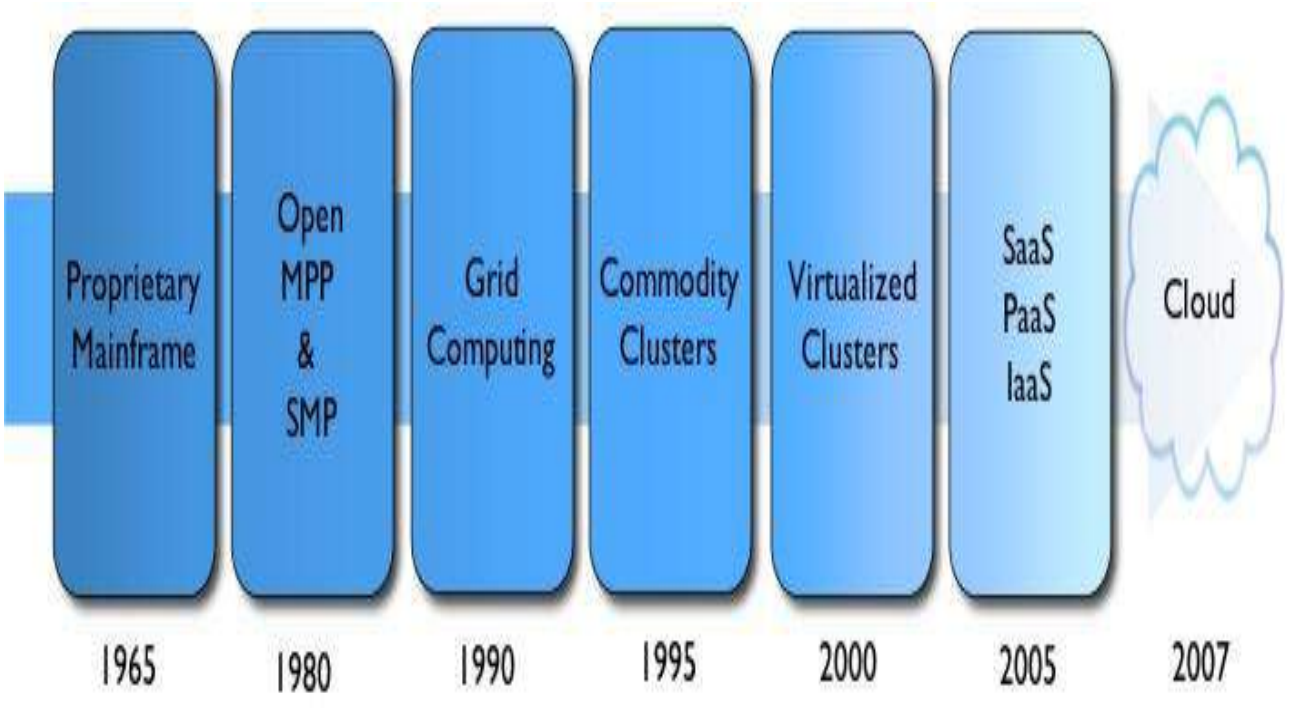


Figure 1. Evolution Of Cloud Computing

Based on scale, with Cloud computing, system resources for completing tasks and their data are no longer stored on personal computers; they are rather hosted elsewhere and are made accessible in any location and at any time to end users. Unlike cloud computing, Grid computing is basically a kind of a combination of distributed as well as parallel computing, in which a powerful computer that is virtualized consists of a network of clustered computers that are connected and are working together to perform difficult operations. Furthermore, autonomous computing systems are those which are capable of self-management.

2.3 **How the Cloud Works**

This could be explained by considering conventional servers years ago (before the advent of virtualization), an enterprise installs an operating system (OS) on its hardware and has an email exchange client running on its operating system suffers the problem of dependency. Dependency here refers to the fact that the email exchange server depends on everything below it. For instance, if the OS gets a virus, or the hard disk crashes, or the CPU fan stops functioning, this will consequently shut down the email exchange services from the email server. In other words, since

the application is installed on the OS and the OS is installed on the hardware, so if either the OS or the hardware fails, the application also consequently fails.

The idea of cloud computing is to try and disconnect these dependencies between the application, the OS and the hardware. This is made possible through the virtualization technology which enables the OS to be placed in a container-like structure (cluster) running on the hardware with a virtualization software running on it. In case the hardware of this server fails, the cluster can be taken out of the failed server and placed in another hardware that is functioning properly or configured to automatically complete this function. A cluster can also determine if one of its servers is down and it redirects traffic of requests away from it and it also performs load balancing if need arises.

Because the computers are set up to work together, the applications can take advantage of all that computing power as if they were running on one particular machine. Cloud computing also allows for a lot of flexibility. Depending on the demand, the user can increase how much of the cloud resources accordingly without the need for assigning specific hardware for the job, or just reduce the amount of resources assigned to the user when they are not necessary.

From a small enterprise's point of view, the idea really is that information can be kept in a cloud or on the Internet and as such it is not compulsory to store important data on company computers. The beauty of this is that if the computer fails, the data is not lost because sensitive data would have been replicated and stored elsewhere prior to systems failure. So, all that is required of the enterprise is to replace the broken computer systems, get access to the internet and retrieve all the information stored in the cloud. For example, some applications (like Google Docs), allow the creation of documents online.

An enterprise can create a report or maybe use a spread sheet, and if the computer's hard disk, or any of their external storage devices breaks, there will be no cause for alarm because the next time the enterprise's system is able to access the internet, the same document will be available. This is because the document was stored on applications servers on the cloud. Another great advantage of cloud computing is that these documents can be shared with other users in the enterprise, and even more users can be invited to collaborate on it and any one of these users can edit it in real time. The saved document could be easily published as a web page or can also be made accessible to the rest of the organization or the wider world via the internet.

3. **Cloud Architecture**

Generic Cloud Design: An internet cloud is envisioned as a public cluster of servers provisioned on demand to perform collective web services or distributed applications using data-centre resources. The following are cloud design objectives as well as an insight into a basic cloud architecture design.

Cloud Platform Design Goals: Scalability, virtualization, efficiency, and reliability are four major design goals of a cloud computing platform. Clouds support Web 2.0 applications. Cloud management receives the user request, finds the correct resources, and then calls the provisioning services which invoke the resources in the cloud. The cloud management software needs to support both physical and virtual machines. Security in shared resources and shared access of data centres also pose another design challenge.

The platform needs to establish a very large-scale high performance computing (HPC) infrastructure. The hardware and software systems are combined to make it easy and efficient to operate. System scalability can benefit from cluster architecture. If one service takes a lot of processing power, storage capacity, or network traffic, it is simple to add more servers and bandwidth. System reliability can benefit from this architecture.

Data can be put into multiple locations. For example, user e-mail can be put in three disks which expand to different geographically separate data centres. In such a situation, even if one of the data centres crashes, the user data is still accessible. The scale of the cloud architecture can be easily expanded by adding more servers and enlarging the network connectivity accordingly.

3.1 **Generic Cloud Architecture**

The internet cloud is envisioned as a massive cluster of servers. These servers are provisioned on demand to perform collective web services or distributed applications using data-centre resources. The cloud platform is formed dynamically by provisioning or de-provisioning servers, software, and database resources. Servers in the cloud can be physical machines or virtual machines. User interfaces are applied to request services. The provisioning tool carves out the cloud system to deliver the requested service.

In addition to building the server cluster, the cloud platform demands distributed storage and accompanying services. The cloud computing resources are built into the data centres, which are typically owned and operated by a third-party provider. Consumers do not need to know the underlying technologies. In a cloud, software becomes a service. The cloud demands a high degree of trust of massive amounts of data retrieved from large data centres. This requires the building of a framework to process large-scale data stored in the storage system. The cloud platform demands a distributed file system over the database system. Other cloud resources are added into a cloud platform, including storage area networks (SANs), database systems, firewalls, and security devices. Web service providers offer special APIs that enable developers to exploit internet clouds. Monitoring and metering units are used to track the usage and performance of provisioned resources.

The software infrastructure of a cloud platform must handle all resource management and do most of the maintenance automatically. Software must detect the status of each node server joining and leaving, and perform relevant tasks accordingly. Cloud computing providers such as Google and Microsoft have built a large number of data centres all over the world. Each data centre may have thousands of servers. The location of the data centre is chosen to reduce power and cooling costs. Thus, the data centres are often built around hydroelectric power plants to get power from it and use the water for cooling.

In general private clouds are easier to manage, and public clouds are easier to access. The trends in cloud development are that more and more clouds will be hybrid. This is because many cloud applications must go beyond the boundary of an intranet. A company's IT personnel must learn how to create a private cloud and how to allow it interact with public clouds in the open internet. Security becomes a critical issue in safeguarding the operation of all cloud types, but the easiest cloud to secure is the private cloud type.

The architecture of a cloud is developed at three layers: infrastructure, platform, and application. These three development layers are implemented with virtualization and standardization of hardware and software resources provisioned in the cloud. The services to public, private, and hybrid clouds are conveyed to users through networking support over the internet and intranets involved. It is clear that the infrastructure layer is developed first to support IAAS services. This infrastructure layer serves as the foundation for building the platform layer of the cloud for supporting PAAS services. In turn, the platform layer is a foundation for implementing the application layer for SAAS applications. Different types of cloud services demand applications of these resources separately.

The infrastructure layer is built with virtualized compute, storage, and network resources. The abstraction of these hardware resources is meant to provide the flexibility demanded by users. Internally, virtualization realizes automated provisioning of resources and optimizes the infrastructure management process. The platform layer is for general-purpose and repeated usage of the collection of software resources. This layer provides users with an environment to develop their applications, to test operation flows, and to monitor execution results and performance. The platform should be able to assure users that they have scalability, dependability, and security protection. In a way, the virtualized cloud platform serves as a "system middleware" between the infrastructure and application layers of the cloud. The layers of the cloud architecture are described below in Fig. 2.

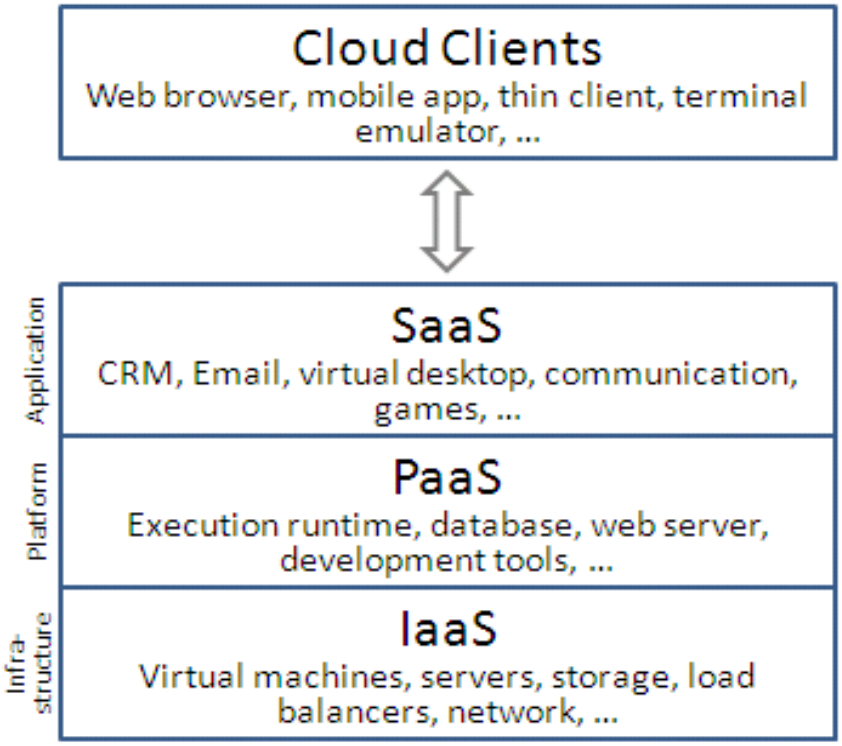


Figure 2. Cloud Computing Architecture

The application layer is formed with a collection of all needed software modules for SAAS applications. Service applications in this layer include daily office management work, such as information retrieval, document processing, and calendar and authentication services. The application layer is also heavily used by enterprises in business marketing and sales, consumer relationship management (CRM), financial transactions, and supply chain management. It should be noted that not all cloud services are restricted to a single layer. Many applications may apply resources at mixed layers. After all, the three layers are built from the bottom up with a dependence relationship.

3.2 **Private Cloud Architecture**

A private cloud possesses four main architectural elements; managing accesses, managing service, managing resource and pooled resources. A self-service portal may give access management and a relatively easy interface, but the automation of service behind that portal is also important.

A virtualized infrastructure that can have rapid re-provisioning as a virtualization tool can give a fluid and flexible resource pool, but unless the allocation of the virtualized resources to meet service agreements is automated, it is not a private cloud architecture. A private cloud will usually be in-sourced and run on-premises using equipment owned by an enterprise, but this is not always true. Private cloud infrastructure can be outsourced to a third party and externally hosted (this is usually referred to as virtual private cloud but that term is used describe a broad spectrum from dedicated equipment to virtual private networks).

A private cloud can have its data centre on or off the enterprise's premises for performance and security reasons. A private cloud that is IAAS will usually leverage virtual machines, but not always. Private cloud solutions that leverage rapid provisioning (e.g. IBM cloud burst) will be available to support physical and virtual devices. The diagram below (Fig. 3.1) shows private onsite and offsite private clouds.

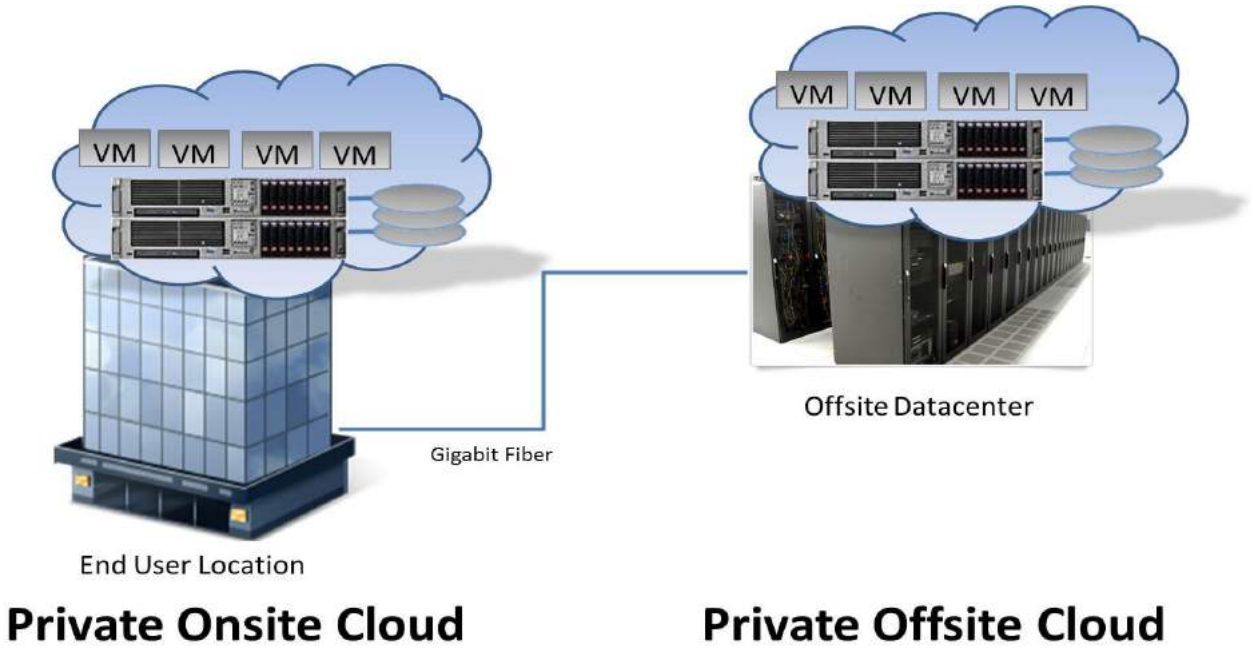


Figure 3.1. Onsite / Offsite Private Cloud Architecture

3.3 **Eucalyptus Private Cloud**

Eucalyptus is an open source infrastructure developed in the Computer Science Department of the University of California, Santa Barbara that implements IaaS style cloud computing using local machine and cluster resources. The software components that make up a Eucalyptus cloud are based on web services and are implemented using only free available software packages, many of which are part of common Linux distributions.

In addition, while the infrastructure itself is modularized so that a variety of interfaces can be supported, an interface module has been initially included to Eucalyptus that conforms to the Amazon AWS interface specification. Thus, once installed and running, a Eucalyptus cloud supports the same programmatic and user interfaces that AWS does with respect to IaaS provisioning. For the purpose of this thesis, AWS was chosen as an initial interface for several reasons.

Firstly, Eucalyptus is the product of a research effort in which the researchers were interested in the viability of AWS as a scientific computing platform. Thus, as a local cloud infrastructure compatible with AWS, Eucalyptus is designed to function as instrument-able development platform that provides transparent and controlled experimentation prior to AWS deployment.

Secondly, the AWS compatibility allows researchers to leverage the rich ecosystem of tools and services that is emerging from the AWS community.

Thirdly, Eucalyptus is designed to foster greater usage of cloud computing in general, and AWS in particular, as a way of simulating and accelerating the paradigm.

Open source projects use AWS as a high performance cloud platform, both for their own development and for application support. Thus, AWS interface compatibility has proved essential in promoting greater AWS usage and thus greater cloud computing uptake.

Eucalyptus is designed to function as a collection of cooperating web services that can be deployed in environments where network connectivity is not necessarily symmetric. Academic research works have access to small clusters, pools of workstations, and various server/desktop machines. Public IP addresses, however, are usually scarce and the security ramifications of allowing complete access from the public internet can be daunting so system administrators commonly deploy clusters of pool of client machines on private, un-routable networks with a single head node responsible for routing traffic between the client pool and a public network.

Although this configuration provides security while using a minimum of publicly routable addresses, it also means that worker machines can initiate connections to external hosts but external hosts cannot typically connect to worker machines running within each cluster. Thus, Eucalyptus adopts a hierarchical design logically; there are four service components within a functioning Eucalyptus installation: the client API translator, the cloud controller, one or more cluster controllers, and one or more node controllers as described in Fig. 3.2 below.

Figure 3.2. Eucalyptus Cloud Components

The interfaces between these components are described by individual WSDL specifications so that any functional component may be replaced or modified. Client requests are translated to a canonical Eucalyptus internal protocol before they are passed to the cloud controller acting as message proxies between the publically routed networks to which each head node is attached and the internal private networks that worker nodes can access. Cluster controllers also implement a scalable scheduling protocol for VM assignment although this scheduling mode can be replaced as a plug in. Finally, each machine within a cloud that is expected to contribute resources (CPU, memory, or disk) to user allocations must run a node controller.

4. **Deployment Models**

Deployment can be described as the launch of an application or platform into the cloud by an organization. Since most organizations are focusing on leveraging the cloud in order to reduce capital expenditure and costs of operations there is a need for organizations to understand the necessary requirements before opting for the various deployment models available on the cloud. Not understanding these requirements could bring security risks and challenges for the IT management of the organization which could in turn be more expensive for them to deal with. These models include the public, the private, the community, and the hybrid clouds. As shown below, the three deployment models are as follows in Fig.4.

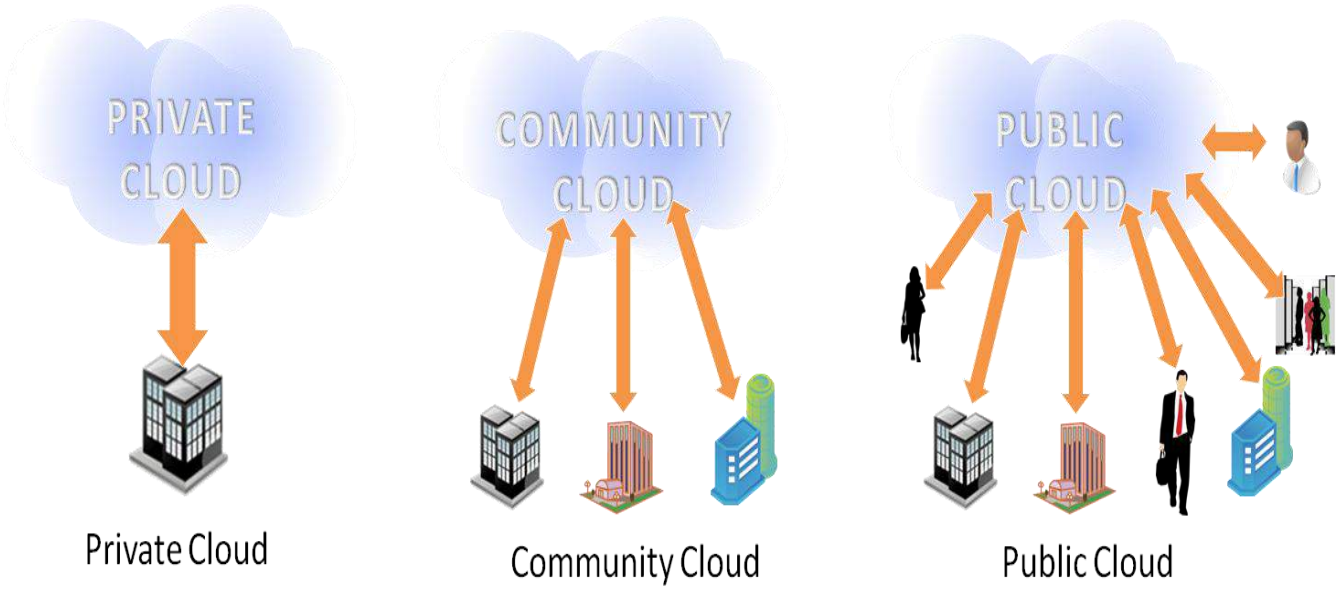


Figure 4. The Three Cloud Deployment Models

In order to determine which deployment model best suits a particular organization, the question of which cloud type to deploy must be answered. The answer to this differs due to the changing needs of different organizations. The public cloud is extremely popular and is best suited for situations like the initial launch of a new internet website because it is exceptionally scalable. It can take in basically any type of setup from single ones to those that are much complicated. It is also the most cost effective option that is available in more recent times. If the price of setting up a cloud is the top priority, the public cloud wins unopposed. Additional situations in which it pays to use the public cloud could be in project collaborations. This is obtainable because the public cloud makes it easy for every participant to view the project and participate simultaneously.

The private cloud is ideal in situations when an enterprise has a product that is one of a kind or service that requires strict monitoring and control. In such situations, there is a frequent need to fiddle with the infrastructure since this requires some management skills on the part of the company. Private cloud is also the best choice when a company must conform to strict security and privacy regulations. Doing this is not very easy in the public cloud. If it is critical to maintain complete control of the cloud at all times, then a private cloud is the answer although it does not come cheap.

The hybrid cloud is ideal for companies that can benefit from both public and private clouds. The hybrid cloud combines the best advantages of the public and private clouds in a very attractive way. A hybrid cloud could be considered in a situation where it does not make sense to keep all the required data in a private cloud. Since the private cloud is costly, by moving some data to the public cloud the company can save some money and this is a typical reason to consider a hybrid cloud. Another reason to consider a hybrid cloud would be security concerns. Exceptionally sensitive information can be kept in the private cloud and others in the public cloud. This is a best of both worlds approach and it works for many companies.

4.1 **Public Cloud**

Public clouds (or external clouds) describe cloud computing in the traditional mainstream sense, whereby resources are dynamically provisioned on a fine-grained, self-service basis over the internet, via web applications or web services, from an offsite, third party provider who shares resources and bills on a fine grained, utility computing basis.

A public cloud is hosted, operated, and managed by a third-party vendor from one or more data centres. The service is offered to multiple customers (the cloud is offered to multiple tenants) over a common infrastructure.

Depending on an organization’s specific needs, such as customized configuration requirements and service level agreements, regarding uptime, an enterprise must carefully consider moving critical applications to a public cloud vendor.

The most important of these requirements to consider is security. The public cloud configuration offloads the most management duties from the client, or user organization, to the third party cloud service vendor.

In a public cloud, security management and day to day operations are relegated to the third party vendor, who is responsible for the public cloud service offering. Hence, the customer of the public cloud service offering has a low degree of control and oversight of the physical and logical security aspects of its local cloud.

Benefits of a Public Cloud: Public clouds such as, Google, Drop-box, and Amazon EC2 offer their users highly flexible cloud environment. They enable users to share data as well as store it as per their personal capabilities. They can decide what to and what not to share with their prospective clients. Public clouds allows scaling up or down at whatever capacity that is required by the enterprise.

Public clouds like these encourage users to create a cloud on their own without necessarily requiring professional help. These are pre-configured clouds existing on the Internet. Organizations that wish to opt for public clouds only need to do is to visit a public cloud website or service portal to get started with it. Third party help is not required to create or run this type of cloud. It will be managed and handled by the client and the client will be the prime proprietor of it.

This particular characteristic enables the cloud technology to be more accessible for organizations to work in a synchronized fashion. The more the cloud services are being used, the brighter the future prospects of the organization becomes. However, there is a flip side to this; payment is charged on the basis of cloud services used by the users.

Another feature of a public cloud is its availability to all and its agility. Anyone who has an Internet connection can access the public cloud from anywhere on the globe at any time. The public cloud enables clients to become more independent and enables the implementation of important business task thus strengthening customer relations and bringing businesses closer to their clients across the globe.

4.2 **Private Cloud**

Private clouds and internal clouds are terms used to describe offerings that emulate cloud computing on private networks. These (typically virtualization automation) products claim to deliver some benefits of cloud computing without the pitfalls, capitalizing on data security, corporate governance, and reliability concerns. Enterprises most buy, build and manage them and, as such, do not benefit from lower upfront capital costs and less hands-on management. The customer for a private cloud is responsible for the operation of his own cloud. Private clouds differ from public clouds in that the network, computing and storage infrastructure associated with private clouds is

dedicated to a single enterprise and it is not shared with any other enterprise (i.e., the cloud is dedicated to a single tenant in the enterprise).

A private cloud offers the potential to achieve greater security over cloud-based assets. However, between the potential for better security and the achievement of better security lie many ongoing activities. The true advantage of a private cloud is that the provider has a vested interest in making the services interface more perfectly matched to the tenant needs. But, it should also be noted that many of the shortcomings of enterprise security have to do with the fact that the enterprise itself implements and manages its own IT security which would be perfectly fine except security is generally not a core investment nor is it measured as though it were.

Benefits of a Private Cloud: The main advantage of private cloud is speed. Implementing a service through a self-service interface, and automating the delivery of those offerings, can increase the speed of delivery automatically. A private cloud by itself has special abilities such as automation, and some form of resource pooling with virtualization which will reduce costs but these can be done without building a complete private cloud.

Private cloud computing restructures IT use and changes the relationship between the customer and the private enterprise to a business relationship, based on service delivery and usage metrics. For most enterprises, this is a difficult change. Variation in processes, management adjustments, changes in funding, service standardization, culture and politics consistently arises in polls as more difficult challenges than technology itself.

However, considering these changes for private cloud computing also lays the path for the enterprise to consider substitutes to the public cloud in the future, if and when they begin to meet enterprise requirements for security, service levels, compliance, etc. Private cloud computing is a stepping stone to public cloud computing and it should be designed to facilitate future sourcing choices. This is because the private cloud can serve as a foundation on which to build robust approaches to cloud computing that can then be used to support a transition to the community, hybrid or public clouds.

4.3 **Community Cloud**

A community cloud allows multiple independent entities to gain the cost benefits of a shared non-public cloud while avoiding security and regulatory concerns that might be associated with using a generic public cloud that did not address such concerns in its SLA. This model has tremendous potential for entities or companies that are subject to identical regulatory, compliance, or legal

restrictions. Different kinds of community clouds are being used in the United States and the European Union by governments at the national and local levels.

This makes great sense since there are multiple benefits to both the individual entities as well as collectively. For instance, when multiple government agencies that transact business with each other have their processing collocated in a single facility, they can achieve both savings and increased security in terms of reducing the amount of traffic that would otherwise need to traverse the internet. Continuity of operations can also be enhanced at a lower overall cost to all parties when multiple data centres are used to implement such a community cloud and this can be seen in Fig. 4.1 below using “Kunde A” and “Kunde B” as two different individuals accessing the same community cloud.

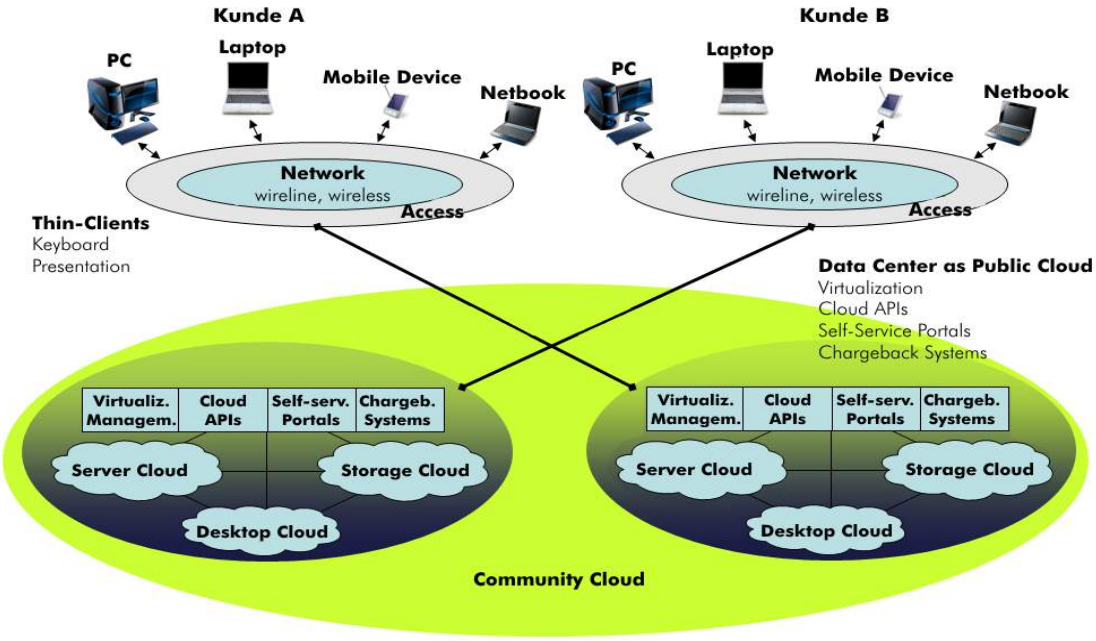


Figure 4.1. Community Cloud Architecture

Benefits of a Community Cloud: In most cases, the standard requirement to be allowed to be a part of a community is to take into consideration an enterprise’s security structure. This is because all community members must be cleared to meet certain security requirement standards which in turn would ensure data security sovereignty amongst community members. This ensures that the community cloud is secure and hard to get infiltrated by attackers.

The community cloud by all means would be larger than a traditional private cloud leading to a more effective utilization of cloud resources and better leverage for assets procurement within the

cloud environment. The main benefit of scale is reduced cost in cloud operations since a large cloud requires low operation cost.

The community cloud is not owned or controlled by any one organisation, and therefore not dependent on the life span or failure of any one organization. It is resilient and robust to failure, and immune to the system-wide cascade failures of vendor clouds, because of the diversity of its supporting nodes. When occasionally failing it will do so gracefully, non-destructively, and with minimal downtime, as the unaffected nodes compensates for the failure.

4.4 **Hybrid Cloud**

A hybrid cloud environment consisting of multiple internal and or external providers is a possible deployment for an enterprise. A hybrid cloud comes about when an organization leverages a public cloud or a community cloud to expand the capabilities of its private cloud thereby implementing a hybrid cloud model. In reality, a hybrid cloud could be formed by the combination of any of the public, private or community clouds. With a hybrid cloud, enterprises may run non-core applications in a public cloud, while maintaining core applications and sensitive data in-house in a private cloud. Service provided through the integration of cloud components are evolving, barriers are being overcome, and enablers are being developed. A major concern is to trust that an enterprise's information is both secure and private. Establishing this trust is a major milestone in the adoption of the full range of cloud computing.

Benefits of a Hybrid Cloud: The hybrid cloud provides quite a number of benefits. It capitalizes on the reduced cost and flexibility of the cloud of this cloud offering which is the amalgamation of two of any of the other cloud types. Hybrid cloud also improves provisioning and scalability at a reduced cost, allowing the allocation of cloud resources for small scale development projects at reduced cost and this usually help avoid making changes to pre-owned infrastructure of the enterprise. One often overlooked advantage of the hybrid cloud is that it allows reluctant enterprises which are too scared to migrate into the cloud fully the opportunity to try the cloud out systematically at a reduced cost and with much reduced risk. This can be an added advantage for company with conservative management teams towards technology and development.

5. **Cloud Delivery Models**

Cloud computing could be classified into one of three delivery models as speculated by national institute of science and technology, (an agency of United States department of commerce) and this is known as the SPI model. As the area of cloud computing was emerging in 2009, the systems developed for the cloud were quickly stratified into three main subsets of systems:

- **Software as a Service** (SaaS),
- **Platform as a Service** (PaaS), and
- **Infrastructure as a service** (IaaS).

Based on this general classification of cloud systems, the SPI model was formed (J. Rhoton, 2009) and denotes the Software, Platform, and Infrastructure systems of the cloud, respectively. The three cloud classification models present different levels of details of the cloud computing landscape, since they emerged at different times of evolution of this computing field. Although they have different objectives, they collectively expedite comprehending some of the interrelations between cloud computing systems. The SPI model is depicted in the diagram below, Fig.5 shows the separate levels of the three SPI models.

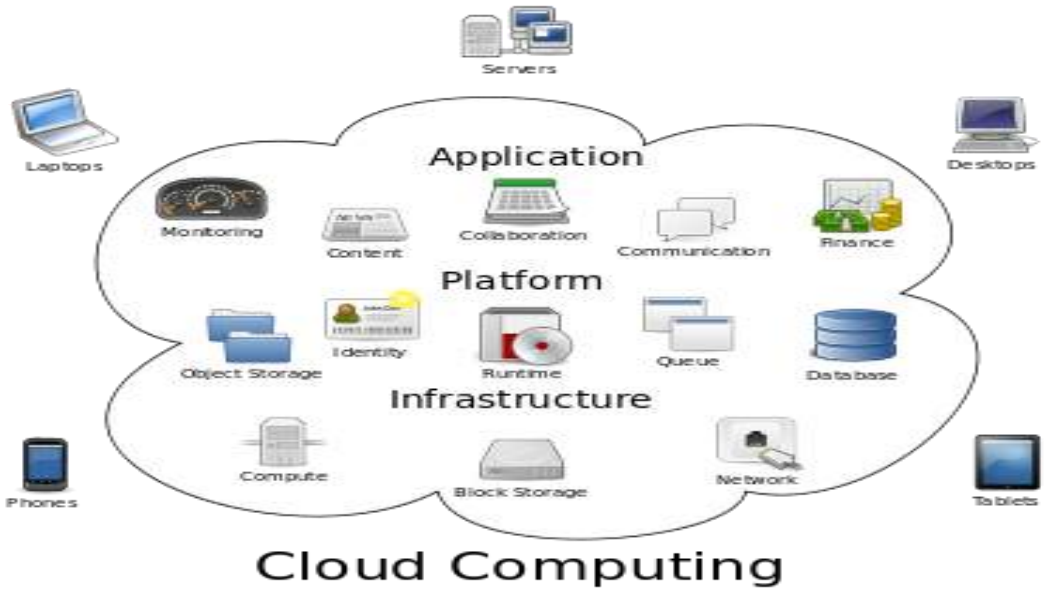


Figure 5. Delivery Model Types (SPI model)

5.1 **Cloud Infrastructure Systems**

Infrastructure as a Service (IAAS) is the simplest of cloud offerings. It is an evolution of virtual private server offerings and basically provides a mechanism to take advantage of hardware and other physical resources without any administrative requirements or capital investment. The benefit of services at this level is that there are very few limitations on the consumer. There may be challenges including (or interfacing with) dedicated hardware but almost any software application can run in an IAAS context.

At this junction, it would be good to further divide IAAS into three categories: Servers, Storage and Connectivity. Providers may offer virtual server instances on which the customer can install a custom image and run. Persistent storage is a separate service which the customer can purchase. Finally, there are several offerings for extending connectivity options. A comparable standard of infrastructure service is Amazon. While they are not unique in their offerings, virtually all IAAS services are either complements to Amazon Web-Services or else considered competitors to them. It is therefore useful to structure the analysis of IAAS along the lines of Amazon's offerings. It could also be noted that there is an open-source equivalent to Amazon Web-Services that is roughly comparable to its interface. Eucalyptus "Elastic Utility Computing Architecture for Linking Useful Systems", which is available as a source package, RPM, and as a Rocks disk image. It comes with Ubuntu starting from version 9.04 as its default cloud computing tool.

SERVERS: Servers represent the allocation of computational resources along with minimal storage and input/output channels. Cloud computing can be seen as the evolution of managed hosting provides such as Terre mark or Savvies. They offer co-location capabilities as well as dedicated pools of rack-mounted servers. Their managed hosting capabilities often include virtualized resources on dedicated infrastructure with console-based provisioning.

The server outsourcing model can be divided into three allocation options: Physical, Dedicated Virtual, and Shared Virtual. Physical allocation means that a specific hardware is allocated to the customers. Dedicated virtualized servers offer dedicated hardware but provide a hypervisor on the physical machine so that customer can run multiple operating systems and maximize server utilization. Shared virtual servers are exposed to the customers as pools of virtual machines. It is not discernible on which physical equipment a particular instance is running or what applications may be co-resident on the same machine.

STORAGE: Storage services, such as Storage Networks, have been around since the late 90s. Similar to the first application hosting providers the initial offerings were not very profitable ("Ruth, et al., 2009"). More recent on-demand offerings have changed the game, however, and made Storage as a Service one of the most promising areas of cloud computing.

The offerings are typically characterized by a location agnostic, virtualized data store that promotes the illusion of infinite capacity in a resilient manner while their level of automation makes them very easy to use. One of the most common applications is an online backup using a SAAS delivery. Storage services are also useful for archiving, content delivery, disaster recovery and web application development.

NETWORK: The notion of cloud computing would be very dull without connectivity. But merely having a network is not sufficient. There are many variations in the kind of capabilities that the connections can have. For instance, by default Amazon EC2 instances will receive a single dynamic (DHCP) address. If they require additional addresses, static addresses, or persistent domains then they need to request these separately.

There are two other network-related functions that cloud providers may offer. There may be two options for network segmentation and mechanisms to bridge the segments. There may also be performance related functionality such as load balancing.

Many cloud server providers, such as Amazon EC2, allow the customer to define firewalls which restrict the inbound and outbound traffic to specific IP ranges and port numbers. Additionally the guest Operating systems may apply further personal firewall settings.

5.2 **Cloud Platform Systems**

Cloud platform system otherwise known as Platform as a Service, originated from a convergence of two trends compared with the suboptimal nature of IAAS for cloud computing and the evolution of web applications. Infrastructure as a service offers many benefits to customers who wish to move their applications into a cloud-based environment. However, infrastructure services tend to run on platforms that were designed for desktops and traditional client-server environments. They may now be virtualized but they have not been optimized for cloud.

Cloud platforms act as run-time environments which support a set of (compiled or interpreted) programming languages. They may offer additional services such as reusable components and libraries that are available as objects and application programming interfaces. Ideally the platform will offer plug-ins into common development environments, such as Eclipse, to facilitate development, testing and development.

Nonetheless, there has been a distinct increase in the number of web hosting services that support a variety of active server-side components ranging from Microsoft ASP.NET and Java to scripts

such as PHP, Python and Ruby on Rails. Compared to infrastructure as a service, these platforms reduce the storage requirements of each application and simplify development. Rather than moving virtual machines with entire operating systems, the application only requires the code written by the developer. An additional benefit is the increased ease for the service provider to sandbox such application by only providing functions that cannot disrupt other tenants on the same system and network.

In summary, the value proposition for PAAS is that it shows some benefits over traditional web platforms in terms of geographically distributed collaboration, facilitation of web service aggregation through centralization of code, reduced costs of infrastructure through pay-as-you-go model and cost reduction through higher-level programming abstractions. At the same time, PAAS is also simpler to manage than IAAS and represents a smaller platform to distribute and it can leverage more functionality and services from the provider.

5.3 **Cloud Software Systems**

Software as a Service (SAAS) which is a cloud software system differs from other delivery models in that it provides a service which is directly consumable by the end-user. IAAS and PAAS offer infrastructure and platforms where systems managers and developers can install their applications but they offer little intrinsic value to a non-technical user.

SAAS provides the full stack of cloud services, and ideally presents these to the end user in a fashion that is not radically different from how users expect to use their applications. There may be some user interface changes that ripple through to the users but the main difference is the deployment, licensing and billing model which should be transparent to corporate end-users.

Consistent with the basic motion of cloud computing, SAAS is a model whereby the customer licenses applications and provisions them to users on demand. The services run on the provider's infrastructure and are accessed through internet as browser applications or they must be downloaded and synchronized with user devices. Some of the characteristics of SAAS services are that they are centrally managed and updated. Typically they are highly standardized but may vary in their configurability as well as their efficiency and scalability. The most common pricing model based on the number of users but there may be additional fees based on bandwidth, storage and usage.

SAAS offers several compelling benefits. It simplifies licensing. In fact, the customer does not need to acquire or directly pay for a software license. There is also not required to calculate maximum

capacity. It outsources the tedious task of application maintenance and upgrades and ties customer costs to usage, which lowers fixed costs and capital investment.

However, SaaS does so at a price of restricting customer flexibility in terms of configuration options and update schedule. It also entails a significant commitment to the provider since it is not trivial to switch from one SaaS vendor to another. There may be APIs for extraction and loading but there are no standards on the semantics of these interfaces. So, it is requires significant effort to automate a migration process.

SAAS has both advantages and disadvantages compared to PAAS and IAAS. As long as the functionality offered by the vendor matches exactly to the requirements of the customer; SAAS is a standardized and optimized solution that generally will result in the lowest costs to the customer. However, the flexibility to customize, or indeed to build additional applications is much greater with PAAS or IAAS. Listed below in Fig. 6 are some applications obtainable from SaaS cloud offering.

Figure 6. Software as a Service End User Application

6. **Securing a Private Cloud**

Why would an enterprise invest in a private cloud when the field of public cloud offerings is expanding? In essence, a private cloud offers: - flexibility and security. Public clouds offer ease of access and financial incentives that are compelling. However, private clouds can address the combined desire for greater flexibility in defining cloud services along with a need to physically control information resources. However, the advantages of a private cloud would be limited to its scale and how it is managed.

A private cloud will likely serve many sets of internal users. These various user groups will often operate against sets of information that need to be isolated from each other. When there is no business need for making data from one group accessible to another, the private cloud must enforce separation. Likewise, the private cloud must maintain whatever sensitivity labels or levels. This can easily complicate the design and operation of storage, networking and other shared resources. A private cloud may also express some services to external users on the Internet, for instance, customers of an enterprise. Such connectivity for customers must be secured in and of itself, but overall the private cloud must also enforce various kinds of separation between sets of internal users and between internal users and any external internet users. In other words, there will be several different security boundaries even within a private cloud.

A private cloud can be networked to support delivery for internally and/or externally facing services. Supporting this is the overall network infrastructure that the private cloud is built upon. There are three choices here: network the cloud to present its services to the internet, to private networks or to both. Based on the mission of the cloud and data sensitivity, the security criteria for these cases may be different and they should be defined by the security policy.

So, it comes down to what should be different between securing a cloud from internet users versus from internal enterprise users? Probably not as much as one might think, the greatest differences will typically be the degree of robustness of the cloud-edge network and server gear and the nature of security strategies for responding to the frequent magnitude of attack attempts that originate on the internet.

Whether from internal users or from the internet, the cloud represents resources that must be protected. To do so, one must limit external interaction with these resources in terms of typical security questions: who (users and IP addresses) should have what form of access, and be capable of which actions and under which circumstances? Doing so; entails the use of several categories of counter measures. To begin, the ingress to the cloud is the best place to filter out

unwanted inbound traffic. Blacklisted IP addresses and IP ranges can be filtered or shunned by network devices, such as routers and firewalls. The flip side of this is identifying enterprise IP ranges, tenants with fixed IP addresses, or cloud operations and management.

In operating a private cloud, there are some security issues that also have to be considered a side the network security. These are practices ensuring a secure private cloud includes the following and these are ways to best protect a private cloud from intrusion and data loss.

Antimalware

The deployment and updating of antimalware software is important within a virtualized environment. Where virus-prone operating systems are used for virtual servers in a manner that makes them subject to viruses, an antivirus solution should be used. This should be made part of the template VM images before a VM is instantiated. The virus signature files will often need to be updated on at least a daily basis. Setting virus-prone servers to automatically update their signature files every several hours will not entail undue overhead, but it will ensure that the maximum protection against viruses is deployed. Using VM's, one achieves an advantage in terms of reducing cost-to-recover from infection. All that is really needed is to stand up a replacement uninfected VM.

A better antimalware approach for a cloud computing infrastructure is one where all input is filtered and examined before it gets to a server. Additionally, in the case of a mission critical application, one will need to maintain strict control over any changes to the system image/applications. For such applications, it is unwise to reach a point where a production environment is constantly being subject to per-host virus exposure and remediation.

Device Configuration

A cloud infrastructure is more than just a collection of servers in a hardware pool. The infrastructure to connect these together and to other networks is equally important, for instance, network switches and routers. If these are not configures correctly, then this would present a vulnerability that could be exploited. Additionally, how each server is configured can also play an important role in overall cloud security. In this regard, if a cloud server is wrongly configured, then this can be exploited by either a user who has access to the server or by a service end point outside the cloud. Consider the consequences that could result if such a server is trusted by other in-cloud servers. Therefore, it is a good practice in private cloud to verify security relevant server configurations.

Routers and switches are another category of devices that are often installed and then forgotten about until an additional route needs to be added. It is just as important to depreciate and remove

unused routers as it is to verify the correct configurations of permitted routes. A broader consideration needs to be given to the set-up of routers and other network devices. Weak password or authentication mechanisms could lead to these devices becoming a jumping point for an attacker. By its very nature, a router in the network will be a trusted device and an attacker on this box will be able to see the network and intercept any traffic going through it.

Intrusion and Anomaly Detection

Deploying network intrusion and anomaly detection systems for cloud should be seriously considered. Although these are generally not deployed throughout an enterprise, they are more common for data centre infrastructure. The down side of such detection system is that, like anti-virus systems, they require frequent care and feeding. Signatures will have to be updates on a regular basis, and labour intensive analysis of the alerts will be necessary. The amount of work and the skills involved in this should not be underestimated. Therefore, this is often outsourced to a third party security monitoring company.

The investment of such a capability is cost effective for a medium to large-sized private cloud. It scales regardless of the number of customers and those becomes cheaper (per customer) with each new internal customer. This scalability or otherwise expensive technologies is a key benefit with private clouds.

Data Backup and Storage

A separate network can also be used to backup servers and other cloud devices. Attaching backup and storage devices to this network (or to a separate storage area network) can reduce traffic on the main network and provide additional security. Users will rarely need to access the files storage directly as this will normally be undertaken via the application.

The internal cloud infrastructure should be designed to cater for the backup and storage requirements, which may well be larger than normal. Users may want to store different images or keep development snap shots of their servers and be able to enable and disable these as required.

While designing the storage requirements of the cloud computing environment, the following of storage will be considered:

Direct Attached Storage – DAS: This is a traditional method of grouping storage devices together for large SCSI disk arrays directly connected to one or more servers. This solution has ongoing utility for a private cloud, but the disks need to be physically collocated with the servers they are connected to.

Network Attached Storage – NAS: These devices are connected via an Ethernet network and can provide data storage services to a multitude of clients. Since NAS devices can be located further away than DAS devices, they can be grouped and located in a more secure part of the data centre.

Storage Area network – SAN: This provides a way to attach storage devices to servers such that they appear to be locally attached to the operation system. As with NAS, storage is typically located away from the client servers. The difference with SAN is that they utilize a Fibre Channel Topology, which provides fast access to the storage devices. Another SAN-style approach is ISCSI, which is important because it offers the control of SANs and the lower extent of IP network.

Internal Disk: The typical server configuration includes internal disks. Although internal disk is good for system performance, there are several draw backs to it in cloud computing. First, as VM's are provisioned to a server, the isolation between VMs may be compromised via disk path ways. The great risk here is that one VM may gain access to the hardware disk and thus be able to see files belonging to another VM.

There are security advantages to using a SAN, particularly in terms of disaster recovery. Servers can boot from a SAN, which can shorten the time from provisioning to booting a server. An additional advantage is that a SAN can be configured to span multiple locations, even remote locations. This means data can be replicated to remote locations and can be retrieved quickly for disaster recovery.

Regulation

The laws and regulation of a state or a country must be considered when considering a private cloud. Some of this regulation applies mainly to one country, such as the Sarbanes-Oxley Act (SOX) in the United States, whereas others will apply internationally. The Payment Card Industry Security Standard (PCI DSS) is a worldwide standard that has strict compliance rules.

Location of Data

Although the location of the data centre may not be considered while installing an internal cloud, a number of considerations need to be made. The location of the data centre and data will be

governed by laws just as if the data were handled by a third party, sometime during the transition to a private cloud.

There are many laws governing what personal data can be collected and where this can be transferred, even if this is for backup processes. Building two data centres, one in the United States and one in a country located within the European Union (EU) and backing up each data centre may seem as a secure and strategic option. However, there are strong privacy laws within EU and these may well prohibit the moving data to a country not within the EU, even if for backup purposes.

Data Retention

The retention time for certain data will also need to be considered. A consolidation of the data centre into cloud architecture may also have led to the modification of the data storage regime in place, perhaps SAN environment. The data must still be archived, probably an off-site location, to the requirements set out in law. Finance data in particular has strict rules and regulations governing it and failure in this area may be very damaging.

Cloud computing security is a critical element in establishing trust in the cloud. Confidence in using the cloud depends on trusted computing mechanisms, robust identity management and access control techniques, providing a secure execution environment, securing cloud communications. An additional effect will be to employ an autonomic computing system which enables self-management, self-healing, and self-protection features and makes the private cloud more reliable, secure, and safe choice for the growing requirements for processing and storing large amounts of information in a cost effective manner.

Because security is a major concern of cloud providers and users, incident management and response are critical factors that affect the viability of the cloud platforms. All stake holders in the cloud must have effective and practiced incident management and response capabilities in place to ensure safe and protected cloud processing and data storage functions. In addition, encryption is an important tool that should be applied to affect the security of VMs and cloud information. Key management and countermeasures against cryptographic key attacks must be incorporated into cloud policies.

Finally, retirement of VMs, software in general, and hardware must follow rigorous policies and procedures in order to protect against the compromise of sensitive information.

7. **Private Cloud Implementation**

To demonstrate the implementation of a private cloud, a Linux Ubuntu 9.04 LTS (Jaunty) server distribution was used. Ubuntu was used because UEC uses the same machine images as AMAZON EC2. It is lean, fast and powerful. While setting-up this cloud, XEN hypervisor was considered instead of KVM. This is because KVM is limited to running only virtualization extensions of Intel VT or AMD-V CPU processors, although Ubuntu prefers KVM for its virtualization support and it is added to the default packages but without these extensions, KVM will not work

XEN is the underlying technology used by EUCALYPTUS. XEN hypervisor allows the simultaneous execution of several guest operating systems on the same computer systems hardware. Several virtual machines can be obtained from a single physical machine using XEN, in order to provide utility computing and server consolidation. Binaries and existing applications run without being modified. The XEN hypervisor controls the CPU scheduling, MMU, and the interrupt controller, giving virtual machines to the guests. In XEN terminology, the first OS is called "domain 0" (dom0), which is booted automatically when the XEN hypervisor starts-up and this is given special administrative privileges and by default can access all physical hardware directly. The administrator can access the domain dom0 which can be used to manage any of the other guest operating systems installed in the system, called the domain U (Dom U) in XEN terminology.

Eucalyptus has three major packages as previously stated Cluster Controller which is responsible for administering virtual cloud network support; the Cloud Controller which composed of the storage system; the front-end and the Node Controller which interacts with XEN to manage individual virtual machines. In this practical setup, the front end server administers cloud controllers and cluster controllers and the back end machines act as node controllers and they run eucalyptus-NC. For this purpose, two dedicated systems are being setup and they have the following hardware configurations:

Front End: Intel Pentium 4 processor with 2.0 GB memory, 64-bit single core processor, 800MHz processor speed, 160 GB Hard disk space and 1000 Mbps internet network. Node Controller: Intel core 2 Duo Processor CPU, 64-bit VT extension multi-core processor, 2.3 GHz processor speed, 4.0 GB memory, 160 GB hard disk space and 100 Mbps internet network speed.

7.1 **Front End Setup**

Linux Ubuntu 9.04 (Jaunty) ISO was downloaded and burned to CD. This fresh Ubuntu CD was installed but could be done in two ways: it could be done either by installing an Ubuntu Cloud CD or an Ubuntu Server CD. For this purpose, we chose to install the Ubuntu Server CD and selected the open SSH server to be installed by default to enable remote connection to the front-end machine. Fig. 7 shows the Ubuntu cloud installation main menu page as seen in appendix 1.

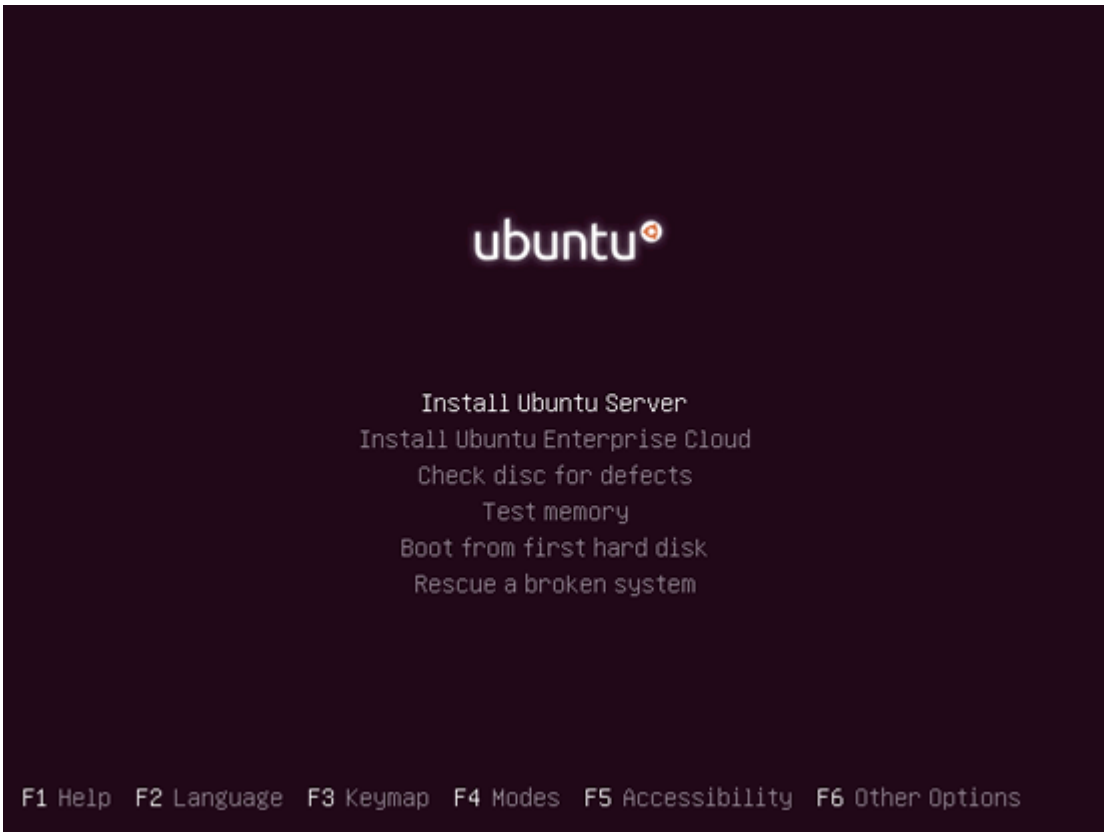


Figure 7. Ubuntu Cloud Installation

Step 1: The systems setup is updated with the following command to bring the systems up to date:

apt-get installs update

Step 2: Postfix Installation: Postfix is a mail server used by Eucalyptus to send notifications to users present on eucalyptus web interface with user privileges from the administrator.

Step 3: Installing the Cloud controller as well as the Cluster controller is the next step. This is done by installing the packages by Ubuntu; the package manager will install all the dependencies automatically.

Step 4: DHCP has to be installed because it is used by Eucalyptus to assign IP's to instances. DHCP server should not be in the running state. In addition, the DHCP server must be set not to run when the system is booted up again.

Step 5: Eucalyptus mode from the configuration files would have to be changed to static so that Eucalyptus can use Internet Protocol addresses from a dynamic pool of addresses that is in the configuration file.

Step 6: More configurations:

The path can be edited using NANO editor and the system has to be rebooted for the changes to take effect.

Step 8: After installing the eucalyptus front-end, the Eucalyptus web interface can be used to setup cluster and set other parameters. The Eucalyptus private cloud web address is URL: https://:8443/ which can be found under the configuration tab also, at this point, add cluster name and Set host to 'local host' Ram disk images and kernel can be created and set in the Eucalyptus official website. It is also important to remember to restart Eucalyptus for the system to effect changes.

7.2 **Back End Setup**

XEN is used as virtualization package; the node controller package eucalyptus-nc was manually installed as seen in Appendix 2. The Eucalyptus node controller was installed by manually downloading and installing by passing dependencies list. Moreover, by using the package manager apt-get dependencies were also installed.

Step 1: Hardware compatibility

Eucalyptus node controller requires Libvirt bin package for proper functioning. Libvirt is a C programming language toolkit with the ability to interact with the virtualization new Linux distributions (and probably other OS's). To install Libvirt-bin package, certain flags in CPU should be supported. Discontinue the app armor for Ubuntu System before continuing installation.

Step 2: Installing the XEN package

By using the XEN virtualization package instead of Ubuntu default KVM and installing the eucalyptus-manually the node controller dependencies, the XEN package is successfully installed on the backend

Step 3 Install Libvirt bin packages

This can be downloaded from the Ubuntu Jaunty web site.

Step 4 Installing Eucalyptus node package

The Eucalyptus-nc package was downloaded from the Ubuntu package repository.

Step 5: Configuring Grub loader for the XEN Kernel and load XEN images to boot into XEN kernel and Setup grub menus. Modules and images were downloaded from the Ubuntu Debian website.

Step 6: Configuring Eucalyptus

Set the configuration depending upon your installation and stop the Eucalyptus daemon.

Step 7: Cloud integration

The permission and certificate files will be replicated to the nodes in order to have password-less communication between the node and cloud. To verify the cloud installation, and check if the node

is communicating, run the **Eucalyptus-describe-availability-zones** verbose command and check the output listings for the presence of all the nodes.

7.3 **Eucalyptus Tools Setup**

Setting up Euca2ools: Dependencies as installed as seen in Appendix 3.

This is started by install curl lib open SSL-ruby ruby

The appropriate tar ball is downloaded from http://open.eucalyptus.com/downloads

Installing cloud dependencies and creating the guest machine:

Installing Java JDK, the current java version and guest image was created using VM builder.

Creating the image:

EMI is created using this image file. This image is in the Ubuntu-XEN folder which could be tested to verify compatibility with XEN using the XM create command. The EMI status can be checked using the XM list command.

Creating EMIs

Now, cloud is setup, images should be created that can be run using Eucalyptus. EMIs for kernel, ram disk and instance still need to be created.

7.4 **Troubleshooting**

In the process of installing Eucalyptus, a few problems were encountered and as such required joining forums as well as searching on Google for assistance.

Eucalyptus Logs

The first problem encountered with this installation was with the logs generated by Eucalyptus.

If there are any issues with starting eucalyptus-nc, on node controllers, checking the eucalyptus test-nc log file for errors will be the first step in solving this problem. The euca_test_nc.log files relates information about all the errors that eucalyptus-nc can possibly encounter while starting nc.log. This can be seen in Appendix 4.

Cloud resources

After installing Eucalyptus and the clusters, and the cloud and nodes are successfully running on the systems, instances may sometimes not run successfully. Most times, the problem is with the node resource registration with the cloud cluster, or the resources have been used up by the instances.

Figure 7.1. UEC Management Software

Searching through the available images and clicking on the install button for desired images. Once the images have been downloaded and installed, clicking on “How to run?” which is displayed below the image button can be used to execute/instantiate/start this image. The image tab will also contain a list of these images as shown in Fig. 7.1 and Fig. 7.2.

Figure 7.2. UEC List of Available Images

Instance issues:

At some point, instance related issues came up in the process of this experiment. In the process of running instances using ec2 tools, instances will stop running and the log files did not provide useful troubleshooting information. Running the XM -list command from XEN is best in troubleshooting this problem.

Some XEN commands used:

XM list

XM create xen.cfg

XM status

XM console

XM saves

XM pause

XM shutdown

XM destroy

XM memory-max

XM memory-set

8. **Conclusion**

Cloud computing is an emerging technology which is still in its infancy bringing about innovations in terms of applications and new business models. The economy of new business models is a major factor driving the adoption and implementation of cloud computing. However, cloud computing faces challenges related to security and privacy. The implementation as well as these security challenges is the motivation behind this project.

Basically, cloud computing allows the IT department of an enterprise to deliver infrastructure, platforms and software applications in an easily managed, easily scaled and easily accessible service architectural structure. More so, the cloud architecture will not be spread to different departments or divisions, but instead can be managed as a holistic resource across the entire enterprise as seen in this project.

Implementing a private cloud using Ubuntu (open source software) has shown in this project that it is possible to link regular desktop computers which lack virtualization capabilities to form clusters based on Ubuntu Enterprise Cloud, using Eucalyptus as the management software to build a private cloud. Installations were fairly easy and following the steps in this thesis would allow a beginner to install an open source cloud but there were problems with logging encountered with Eucalyptus.

The result of this thesis is a fully functional private cloud with possibilities of Infrastructure as a service which is the interactive Ubuntu Enterprise Cloud interface which is used to install images of operating systems and test the full functionality of the cloud. This has been a very challenging project because of limited educational resources. These problems require skills and experience to solve and in this case, was fortunate to find answers on internet blogs.

References

Antonpoulos, N. and Gilliam, L. (2010) Cloud Computing: Principles, Systems and Applications. London: Springer-Verlag London Limited.

Blog Akash (2010). Cloud Implementation:http://www.akashsharma.me/private-cloud-setup-using-eucalyptus-and-xen/#Step_4_Install_dhcp_server (Accessed October 2012)

Business case for cloud computing solutions (2008): Blue print of success http://www.ingrammicro.com/visitor/servicesdivision/businesscasecloudcomputing_practiceguide.pdf (Accessed October 2012)

Carugi, C.S., (2012) http://www.itu.int/ITU-D/tech/events/2011/Moscow_ZNIIS_July11/Presentations/07-Carugi_cloud.pdf (accessed November, 2012)

Cloud Advantage (2010) http://www.itworld.com/answers/topic/cloud-computing/question/what-are-main-advantagesdisadvantages-hybrid-cloud-model-vs-p (accessed November, 2012)

Cloud Architecture: http://www.khaleejtimes.com/Displayarticle09.asp?section=technology&xfile=data/technology/2011/March/technology_March18.xml (Accessed December 2012)

Cloud Background (2011): http://epic.org/privacy/cloudcomputing/#Background (Accessed October 2012)

Cloud based disaster recovery (2012) http://tek-tips.nethawk.net/defining-cloud-computings-key-characteristics-deployment-and-delivery-types/ (accessed November, 2012)

Cloud Business (2011): http://www.ingrammicro.com/visitor/servicesdivision/businesscasecloudcomputing_practiceguide.pdf (accessed October, 2012)

Cloud Computing Architecture: http://upload.wikimedia.org/wikipedia/commons/3/3c/Cloud_computing_layers.png (Accessed December 2012)

Community Cloud Architecture: http://www.itwissen.info/definition/lexikon/Community-Cloud-community-cloud.html (Accessed December 2012)

Community Cloud Computing (2009): http://arxiv.org/pdf/0907.2485.pdf (Accessed November 2012)

Cloud Computing Essentials (2010) http://www.isaca.org/Groups/Professional-English/cloudcomputing/GroupDocuments/Essential%20characteristics%20of%20Cloud%20Computing.pdf (accessed December, 2012)

Cloud computing (2011) http://en.wikipedia.org/wiki/Cloud_computing (accessed November, 2012)

Cloud computing (2012) http://epic.org/privacy/cloudcomputing/#Background (accessed November, 2012)

Cloud computing implementations (2012): http://www.networkcomputing.com/other/idc-cloud-computing-implementations-to-double-by-2012.php?type=article (Retrieved 12 December 2012)

Cloud Evolution Fig.2: http://jameskaskade.com/?p=344 (Accessed December 2012)

Cloud security (2012) http://my.safaribooksonline.com/book/-/9781597495929/chapter-7dot-security-criteria-building-an-internal-cloud/207 (accessed December, 2012)

Delivery Model Types SPI Fig. 5: http://upload.wikimedia.org/wikipedia/commons/b/b5/Cloud_computing.svg (Accessed December 2012)

Dell Systems (2009) http://www.dell.com/downloads/global/solutions/public/articles/flying-cloud-navigator-platform-infrastructure-private-community.pdf (accessed November, 2012)

Emeneker, W. et al, (2006) “Dynamic Virtual Clustering with Xen and Moab, ISPA”. Springer-Verlag LNCS 4331, pp. 440-451.

Forking Denmark (2009) http://www.it-c.dk/en/Forskning/Technical-Reports/2012/~/media/766D278B935547B18AD1D23A8D98A724.ashx (accessed December, 2012)

Good read (2006) G.K Chesterton. http://www.goodreads.com/quotes/55643-there-are-no-rules-of-architecture-for-a-castle-in (accessed December, 2012).

GSA systems (2010) http://info.apps.gov/content/what-cloud (accessed December, 2012)

Hwang, K.K., Fox, G.C., and Dongarra, J.J. (2012) Distributed and Cloud Computing: From Parallel Processing to the Internet of Things. Burlington: Morgan Kaufmann, Inc.

Hybrid cloud computing (2009): A New Era http://www.ijcst.com/vol22/2/sujayR.pdf (Accessed October 2012)

IBM Cloud Computing Infrastructure Architect V1 (2012) http://www-03.ibm.com/certify/tests/obj280.shtml (accessed December, 2012)

I-Net interactive (2012) http://www.webhostingtalk.com/showthread.php?t=1160495 (accessed December, 2012)

Installing Ubuntu (2012): http://www.canonical.com/projects/landscape/cloud-management (accessed December, 2012)

Install Ubuntu Server (2011): Ubuntu Architecture: http://testcases.qa.ubuntu.com/Install/ServerUECTopology1 (Retrieved December 2012)

Itworld.com (2010): Cloud advantages. http://www.itworld.com/answers/topic/cloud-computing/question/what-are-main-advantagesdisadvantages-hybrid-cloud-model-vs-p (Accessed December 2012)

J., Rhoton (2009) Classification of cloud systems: implementation handbook for enterprises. Massachusetts: Recursive Press Ltd. (accessed January, 2013).

Keller, S.S., Powell, B.A., Horstmann, B., Predmore, C. and M., Crawford (2006) Information Security threats and practices in Small businesses, 22, 7-19.

Kiran Murani (2011). UEC images: http://kiranmurari.wordpress.com/2010/03/29/uec-bundling-windows-image/ (Accessed October 2012)

Landscape Cloud Management Canonical (2012) https://help.ubuntu.com/community/UEC/CDinstall (Retrieved October 5, 2012)

Linux Ubuntu (2010). Ubuntu Versions Iso-Images: http://old-releases.ubuntu.com/releases/karmic/ (Accessed October 2012)

Michael Porterfield (2011) http://www.nasa.gov/offices/ocio/ittalk/06-2010_cloud_computing.html (accessed December, 2012)

NIST U.S (2011) http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (assessed November, 2012)

Open Source Cloud (2010). http://www.academia.edu/1910578/Tutorial_-_Building_Private_Cloud_with_Open_Source_Software_for_Scientific_Environments (Accessed December 2012)

Plexibus (2011): Configuring Amazon Images: http://blogs.plexibus.com/2010/05/26/eucalyptus-configuring-your-private-cloud-to-resemble-amazon-ec2/ (Accessed September 2012)

Private cloud Ubuntu (2012) http://www.ubuntu.com/cloud/private-steps

Private cloud setup (2011) http://www.scribd.com/doc/71599304/Private-Cloud-Setup-Using-Eucalyptus-and-Xen 2011-03-17_0248758.pdf (accessed November, 2012)

Rhoton, J. (2009) Cloud Computing Explained: Implementation Handbook for Enterprises. Massachusetts: Recursive Press Ltd.

Rittinghouse, J.W., and Ransome, J.F. (2010) Cloud Computing: Implementation Management and Security. USA: CRC Press.

Ronald, L., Krutz,A., and Russell, D.,(2010) Cloud Security: A comprehensive guide to secure cloud computing. London: Wiley publishing, Inc.

Run your cloud: part1 (2012). http://fnords.wordpress.com/2009/10/04/run-your-own-uec-part-1/ (Accessed 1 December 2012)

Run your cloud: part 2 (2012). http://fnords.wordpress.com/2009/10/07/run-your-own-uec-part-2/ (Accessed 1 December 2012)

Run your cloud: part 3 (2012). http://fnords.wordpress.com/2009/10/07/run-your-own-uec-part-3/ (Accessed 1 December 2012)

SAAS End User Applications Fig.6: http://www.bigfootretail.com/90-of-ecommerce-on-saas/ (Accessed December 2012)

Securing the cloud (2011) http://my.safaribooksonline.com/book/-/9781597495929/chapter-2dot-cloud-computing-architecture/40 (accessed December, 2012)

Sharma Akash (2010). Private cloud setup using eucalyptus and Xen http://www.akashsharma.me/private-cloud-setup-using-eucalyptus-and-xen/ (Accessed November 2012)

Sharma Akashi (2010) Private Cloud Installation http://www.akashsharma.me/private-cloud-setup-using-eucalyptus-and-xen/#Step_4_Install_dhcp_server (accessed November, 2012)

Smoot, S.R., and Tan, N.K. (2012) Private Cloud Computing: Consolidation, Virtualization, and Service-Oriented Infrastructure. Burlington: Morgan Kaufmann Press.

Sosinsky, B., (2011) Cloud Computing Bible. London: Wiley Publishing Inc.

Storage as a service (2009) Ruth et al., http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.134.2457 (accessed January, 2013)

TechNet UK (2011) http://technet.microsoft.com/en-us/magazine/hh509051.aspx (accessed September, 2012)

Tobias Bartholdi (2010) BachelorThesis_TobiasBartholdi_BWI-A09.pdf (accessed December, 2012)

Truth lies behind the code lines (2012) http://fnords.wordpress.com/2009/10/04/run-your-own-uec-part-1/ (accessed January, 2013)

Vaquero et al., (2009) Technology Policy Institute. http://www.techpolicyinstitute.org/files/yoo%20architectural_and_policy_implications.pdf (accessed December, 2012)

Velte, A.T., Velte, T.J., and Elsenpeter, R. (2012) Cloud Computing: A practical Approach. USA: McGraw-Hill Companies.

VMware Cloud Services (2010) http://www.vmware.com/files/pdf/vcat/Private-VMware-vCloud-Service-Definition.pdf (accessed November, 2012)

Winkler, V. (J.R) (2011) Securing the Cloud: Cloud Computer Security and Tactics. USA: Syngress.

Why Your Small Business Needs Cloud Computing (2012) http://www.cloudcomputingworld.org/Cloud-computing-for-businesses/why-your's-small-business-needs-cloud-computing.html (Retrieved December, 2012)

Zoran Pantic (2011) Private Cloud Building: http://www.academia.edu/1910578/Tutorial_ (Accessed December, 2012)

Appendices

Appendix 1: Front End Setup

Step 1: Update the systems setup by using the command:

apt-get installs update

Step 2: Install postfix.

Postfix is a mail server. Eucalyptus sends notifications to users present on eucalyptus web interface with user privileges from the administrator. Postfix is installed, which eucalyptus uses as mail-server. Procedures to installing postfix and setting up configurations as well as testing successful Postfix installation:

apt-get install postfix

Set configurations:

sudo dpkg-reconfigure postfix

Setting these specifications with given values.

General mail configuration type: Internet Site "NONE" doesn't appear to be requested in the configuration System mail name is server1.example.com Postmaster and Root mail recipient is: ('Optional')

Other destinations of mail: example.com, server1.example.com, localhost, localhost.example.com, ayoperu.dc.turkuamk.fi

Mail queue force synchronous updates? : No Local networks: 127.0.0.0/8

Current configuration doesn't request Yes Mailbox size limit (bytes): 0

Extension character of local address: + Internet protocols to use: all

Test postfix installation

EHLO localhost

Mail from: root@localhost

Rcpt to: ayoperu.dc.turkuamk.fi

Data

Subject: Type your message and finish with "." your sub (press enter twice) then Enter Finish with a "Quit" after this, Check the recipient mail (if mail is present in the mail box then it is functioning)

Step 3: Install Cloud controller and Cluster controller.

Install the packages by Ubuntu; the package manager will install all the dependencies automatically.

apt-get installs eucalyptus-cc eucalyptus-cloud

Step 4: Install dhcp server

DHCP has to be installed because it is used by Eucalyptus to assign IP's to instances. DHCP server should not be in the running state. Also, the DHCP server must be set not to run when system is booted up again.

apt-get installs dhcp3-server

/etc/init.d/dhcdbd stop

update-rc.d -f dhcdbd -removes

In the eucalyptus configuration file, Set the DHCP server path which is in the "/etc/eucalyptus/eucalyptus.conf" using nano text editor and change VNET_DHCPDAEMON to be /usr/sbin/dhcpd3

Step 5: Configuring Eucalyptus

Eucalyptus mode has to be set to static so that eucalyptus can use IP's from the pool of IP's that is in the configuration file and use the machine addresses provided along with IP addresses. Eucalyptus can be found in the configuration file /etc/eucalyptus/eucalyptus.conf. Open the: /etc/eucalyptus/eucalyptus.conf using nano text editor and set also the next constraints VNET_INTERFACE="peth0″

Locate 'kvm'and replace with XEN

VNET_BRIDGE="eth0″

locate VNET_MODE="STATIC" and enable it, locate VNET_MODE="SYSTEM" and that should be disabled. Add IP addresses and add machines for instances which are to be used and also change the network setting of the network with settings of VNET_MODE="STATIC"

vnet_subnet="192.168.145.0″

vnet_netmask="255.255.255.0″

vnet_broadcast="192.168.145.255″

vnet_router="192.168.145.1″

vnet_dns="192.168.150.42″

vnet_macmap="AA:DD:11:CE:FF:ED=192.168.145.73 AA:DD:11:CE:FF:EE=192.168.145.75 AA: DD: 11: CE: FF: EF=192.168.145.77″

Step 6 More configurations:

(Xend-http-server yes)

Step 7 Reboot the system to take effect

Step 8 Cloud configurations

After installing the eucalyptus front-end, the eucalyptus web interface can be used to setup cluster and set other parameters. The Eucalyptus private cloud web address is URL: https://:8443/ which can be found Under the configuration tab also, at this point, add cluster name and Set host to 'localhost'

Ramdisk images and kernel can be created and set in the eucalyptus official website. Restart the eucalyptus daemon to take effects

Appendix 2: Back End Setup

Step 1: Hardware compatibility

Eucalyptus-nc requires 'libvirt-bin'package. Libvirt is a toolkit in C which interact with the virtualization capabilities of recent Linux versions (and probably other OS's). To install 'libvirt-bin'package, certain flags in CPU should be supported. # egrep '(vmx|svm)' /proc/cpuinfo [If no print], libvirt-bin cannot be installed

Stop the apparmor for Ubuntu System

/etc/init.d/apparmor stop

update-rc.d -f apparmor remove

Step 2: Installing the XEN package

By using the XEN virtualization package instead of Ubuntu's default KVM and installing the eucalyptus-manually: node controller dependencies. # apt-get install iptables iproute module-init-tools python26 python2.5 # apt-get install xen-utils

apt-get install ubuntu-xen-server

Step 3 Install libvirt-bin package

Eucalyptus-nc requires 'libvirt-bin 'package. Libvirt is a toolkit written in C to interact with the virtualization capabilities of newest Linux versions (and probably other OS's).

apt-get install adduser bridge-utils iptables dnsmasq-base logrotate libsasl2-2 libxen3 # apt-get install netcat-openbsd libavahi-common3 libavahi-client3 libdbus-1-3 libc6 # apt-get install libgcrypt11 libhal1 libgnutls26 libpolkit-dbus2 libreadline5 libpolkit2 libselinux1 # apt-get install libtasn1-3 libxml2 zlib1g libvirt0 policykit

Libvirt-bin package can be downloaded from

http://packages.ubuntu.com/jaunty/libvirt-bin

dpkg -i libvirt-bin_0.6.1-0ubuntu5_i386.deb

sudo adduser $USER libvirtd

Step 4 Installing Eucalyptus node package

apt-get install apache2 eucalyptus-common libapache2-mod-axis2c eucalyptus-gl # apt-get install libaxis2c0 dhcp3-server librampart0 vlan aoetools # apt-get install libvirt0 zlib1g libc6 libcurl3-gnutls Eucalyptus-nc package was downloaded from Ubuntu package repository. eucalyptus-nc package: http://packages.ubuntu.com/jaunty/eucalyptus-nc

dpkg -i eucalyptus

Step 5: Configuring Grub loader for the Xen Kernel

Load xen images to boot into XEN kernel and Setup grub menus. Modules and image were downloaded from the ubuntu debian websites

sudo dpkg -i -xen-686_2.6.26-15_i386 linux-modules-2.6.26-2.deb

sudo dpkg -i -xen-686_2.6.26-15_i386 linux-image-2.6.26-2 .deb

Step 6: Configuring Eucalyptus

Set the configuration depending upon your installation and stop the eucalyptus daemon.

/etc/init.d/eucalyptus-nc stops edit: /etc/sysctl.conf

(Uncomment net.ipv4.ip_forward=1)

Edit: /etc/eucalyptus/eucalyptus.conf (Set)

vnet_bridge=”eth0″

vnet_interface=”peth0″

hypervisor=”XEN”

vnet_mode=”static”

Using nano: Edit: /usr/share/eucalyptus/gen_libvirt_xml. Search for ‘sda’ and replace with ‘xvda‘for static ip addresses

Edit: /etc/xen-tools/xen-tools.conf gateway 192.168.145.1

Broadcast addresst: 192.168.145.255

netmask: 255.255.255.0

(Xend-http-server yes)

(Xend-Unix-server yes)

/etc/init.d/eucalyptus-nc start

Cloud integration:

The front controller and node controllers are ready. Communication has to be initiated between cluster controller and node controllers. Nodes list will be added to the cloud, in order for the cloud to detect the available node.

sudo euca_conf -addnode

This command will add the node to the configuration file. The permission and certificate files will be replicated to the nodes in order to have password-less communication between the node and cloud. To verify the cloud installation, and check if the node is communicating, run Euca describe verbose command to verify the output listings for the presence of all the nodes.

Appendix 3: Setting up Euca2ools:

Install dependencies # apt-get install curl libopenssl-ruby ruby

The appropriate tarball is downloaded from **http://open.eucalyptus.com/downloads**

Steps:

Tar zxvf euca2ools-1.0-*.tar.gz

Cd euca2ools-1.0-* (this should replace the actual directory name)

sudo -s

Echo deb files ://${ PWD}> /etc/apt/sources. List

apt-get update

apt-get install euca2ools

a warning follows this...

WARNING: The following packages cannot be authenticated! …

You can install these packages: continue without verification [y/N]? Yes

Setting up ec2 tools:

Eucalyptus supports ec2-api-tools-1.3-30349 and ec2-ami-tools-1.3-26357 ec2 tools versions. Installing dependencies:

Install jdk the current type java version.

Steps:

Download ec2-api-tools-1.3-30349 and ec2-ami-tools-1.3-26357. Unzip these archives

Copy the ‘bin’ and ‘lib’ folders from the ~/.euca folder.

Export the following parameters

Export ec2_home= ~/.euca

Export java_home= /jdk1.6.0_12

Downloading the ‘euca2-admin-x509.zip‘file from cloud site at https://:8443/.

Unzip this archive which can be found at ~/.euca folder retrive and export it.

Source ~/.eucarc

Creating the guest machine

Guest image was created using vmbuilder.

Create the image.

Emi is created using this image file.This image is in the ubuntu-xen folderwhich could be tested to verify compartibility with xen using he xm create command.

xm create /usr/ayoperu/ubuntu-xen/ cloud/xen.cfg

The emi status can be checked using the xm list command.

xm list

If ayoperu is seen as the emi ID in the output, the image can be run with XEN.

Creating EMIs:

Now, cloud is setup, images should be created that can be run using eucalyptus. Also to be created is emi's for kernel, ramdisk and instance.

Kernel emi

Steps:

mkdir kernel

-bundle vm images, upload bundle and register ec2.

ec2-bundle-image

ec2-upload-bundle

EKI=`ec2-register

echo $EKI

Ramdisk emi

Steps:

mkdir ramdisk

-bundle vm images, upload bundle and register ec2.

ec2-bundle-image

ec2-upload-bundle

ERI=`ec2-register

echo $ERI

instance emi

Steps:

mkdir image

ec2-bundle -i root -image.img -d. /image

ec2 -b plabImage -uploads-bundle-m ./image/root.img.manifest.xml

Echo $EMI

Run instance

Add key pair

ec2-add-keypair mykey > ~/.euca/mykey.priv

chmod 0600 ~/.euca/mykey.priv

ec2-run-instances -k mykey emi-73FD0EAD –ramdisk eri-E47F14E8 –kernel eki-8D701398 –instance-type m1.xlarge

Appendix 4: Troubleshooting

Some Xen command

xm list

xm create xen.cfg

xm status

xm console

xm save

xm pause

xm shutdown

xm destroy

xm mem-max

xm mem-set

Screenshots and Images

Step-by-step installation screenshots of Ubuntu Private Cloud: Front-End and Back-End.

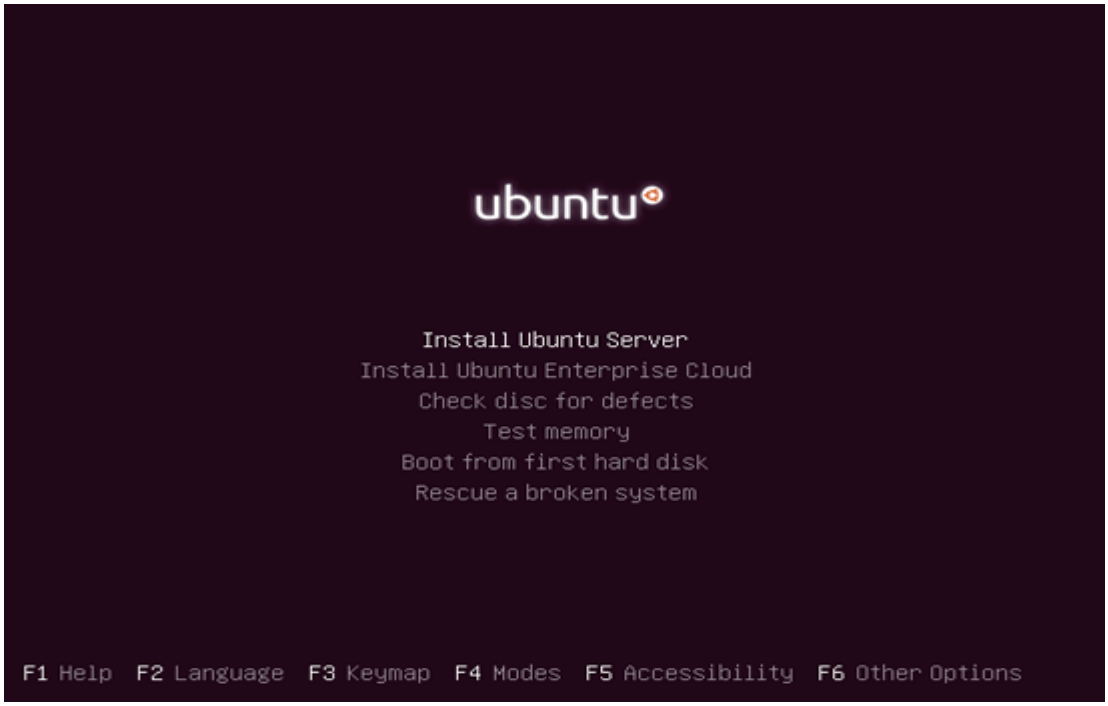
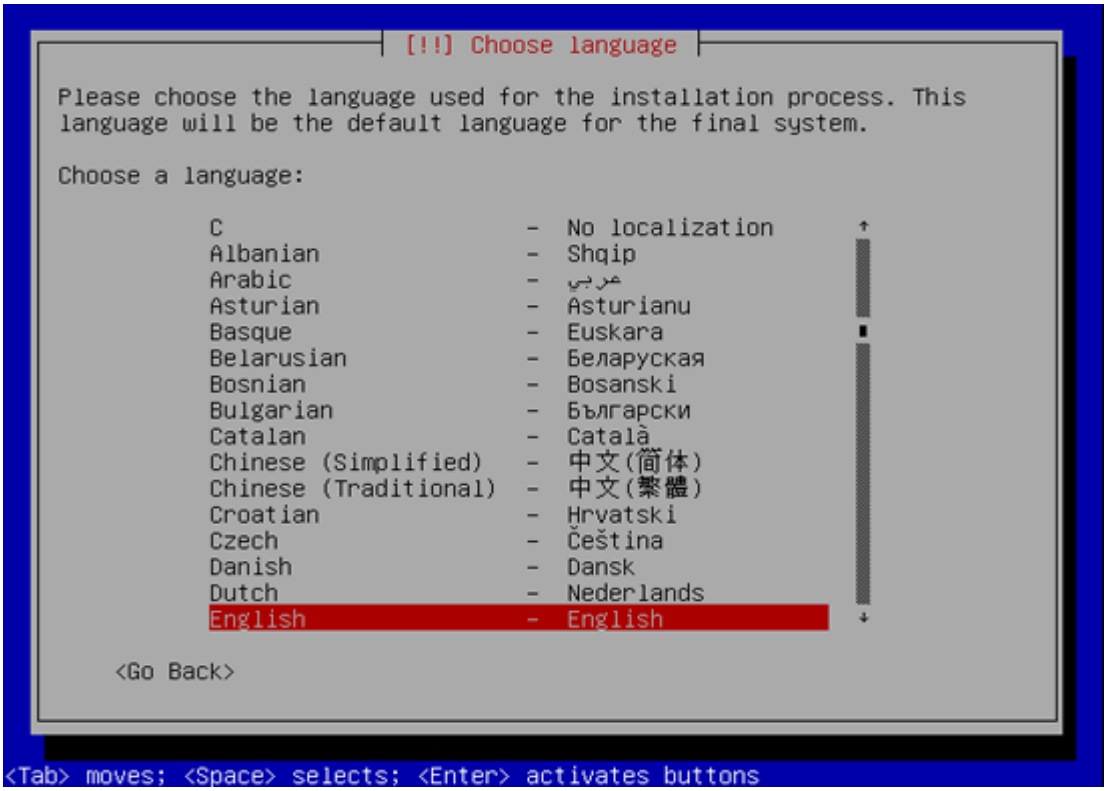


Image 1:- Ubuntu Installation Screen



-

Image 2:- Ubuntu language selection page

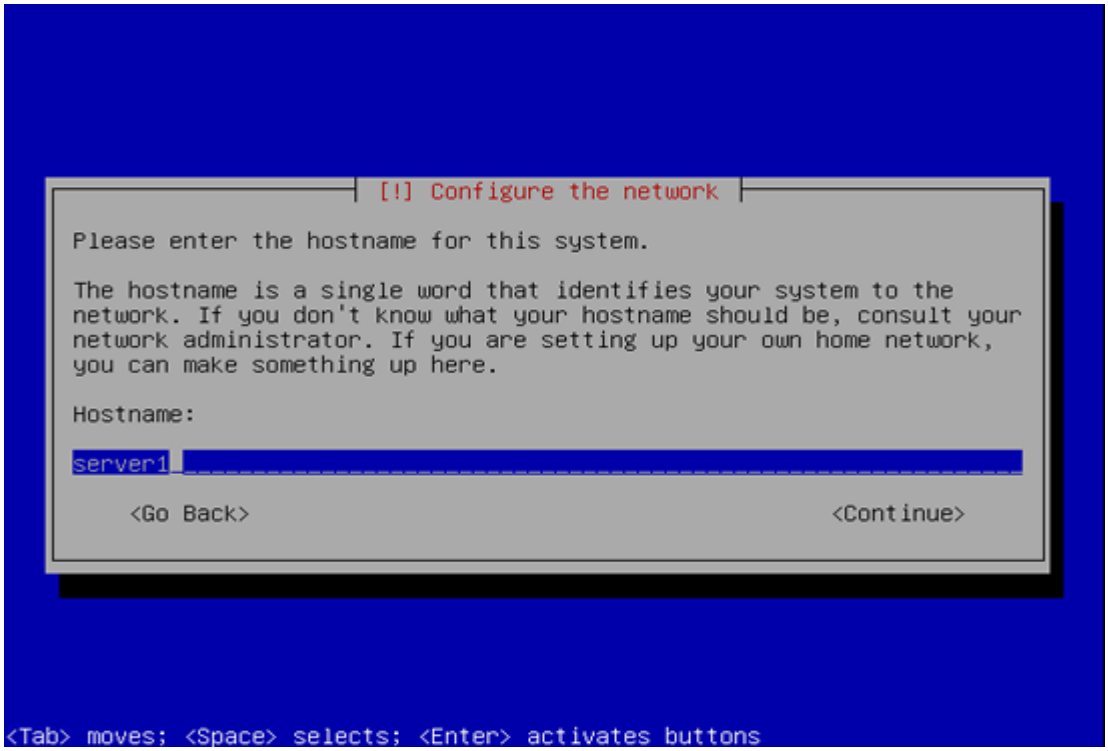


Image 3:- Ubuntu Network Configuration

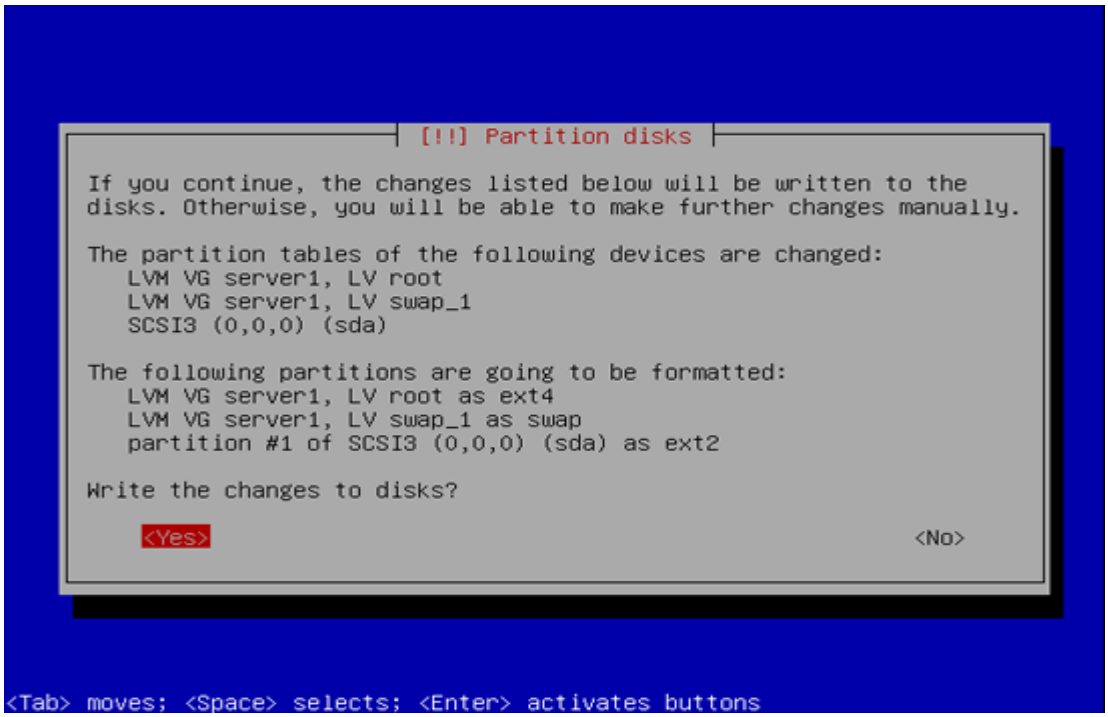


Image 4: Disk Partition Prompt

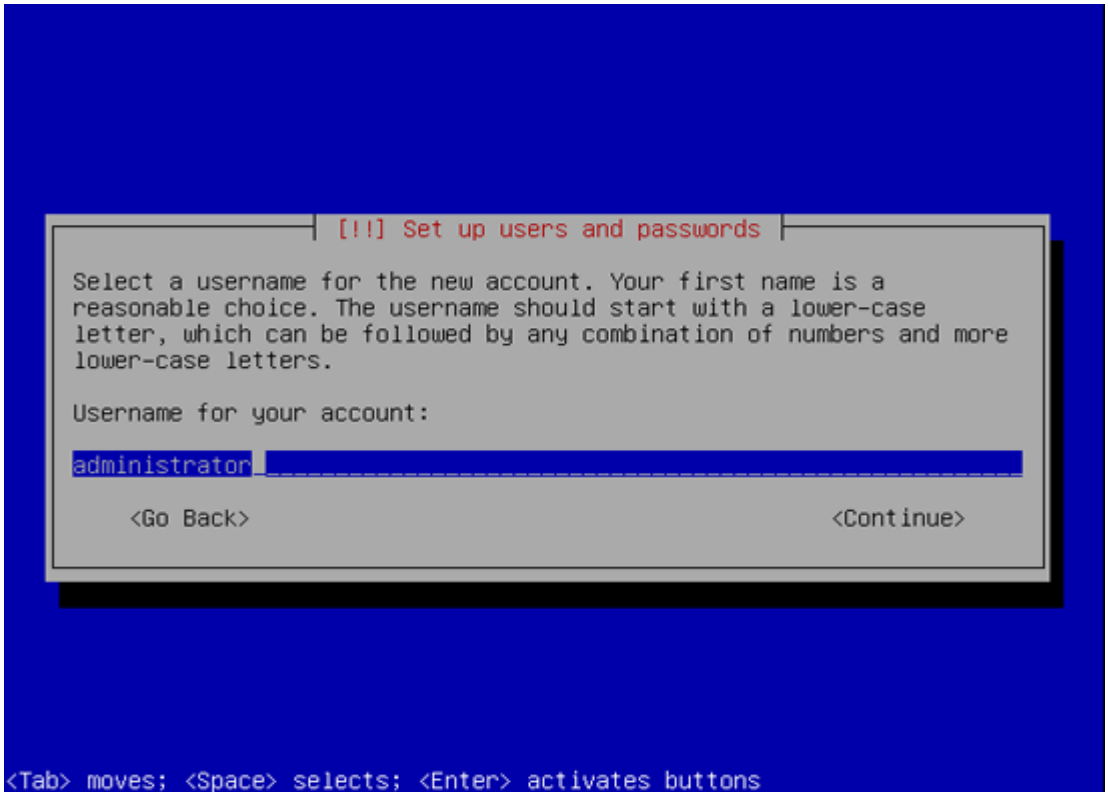


Image 5:- User Account Creation

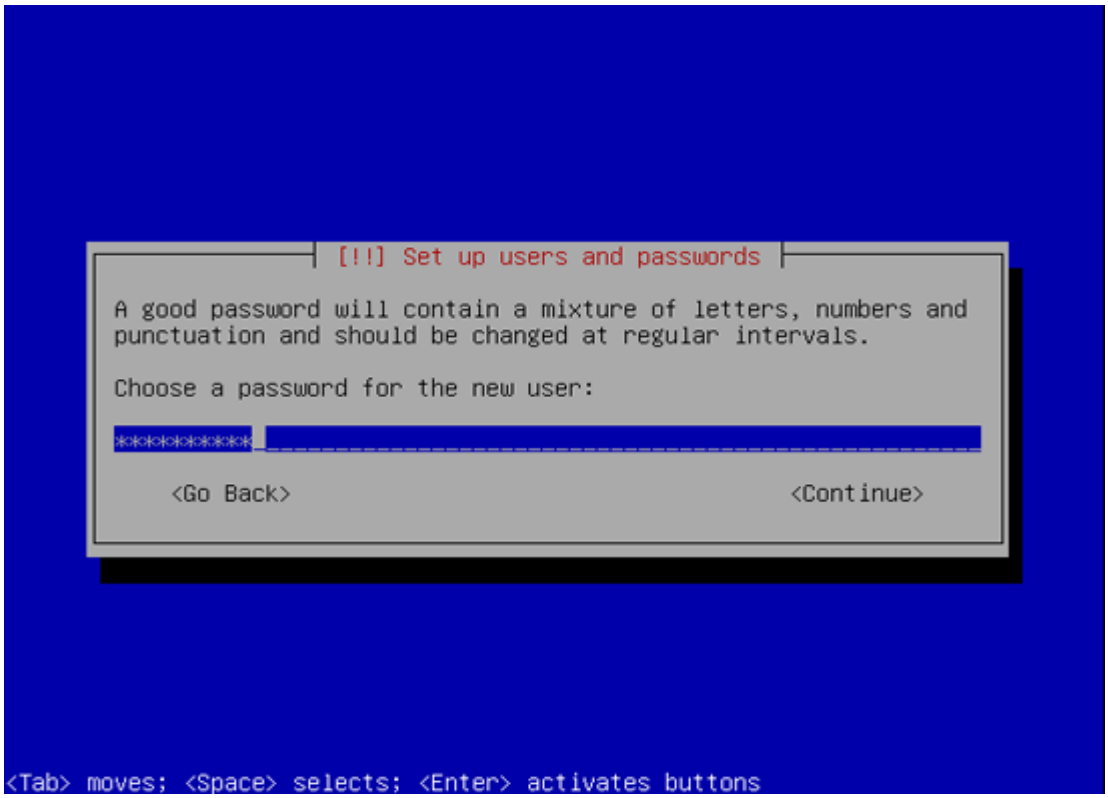


Image 6:- Password Creation

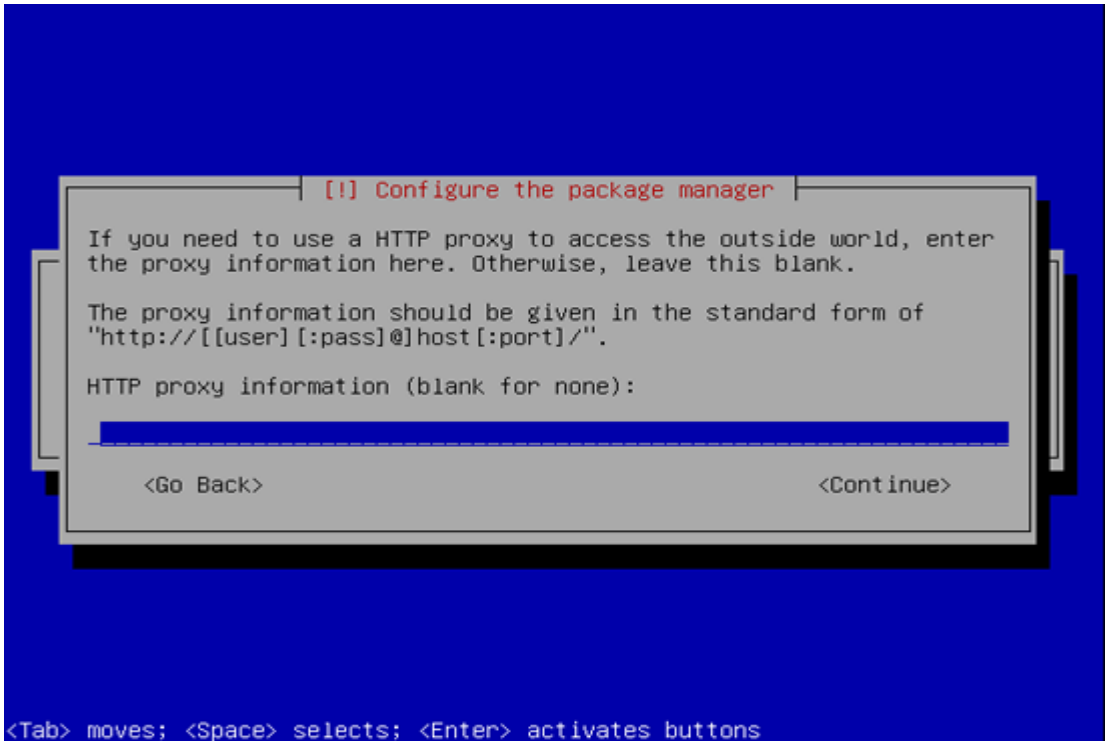


Image 7:- Proxy server Installation

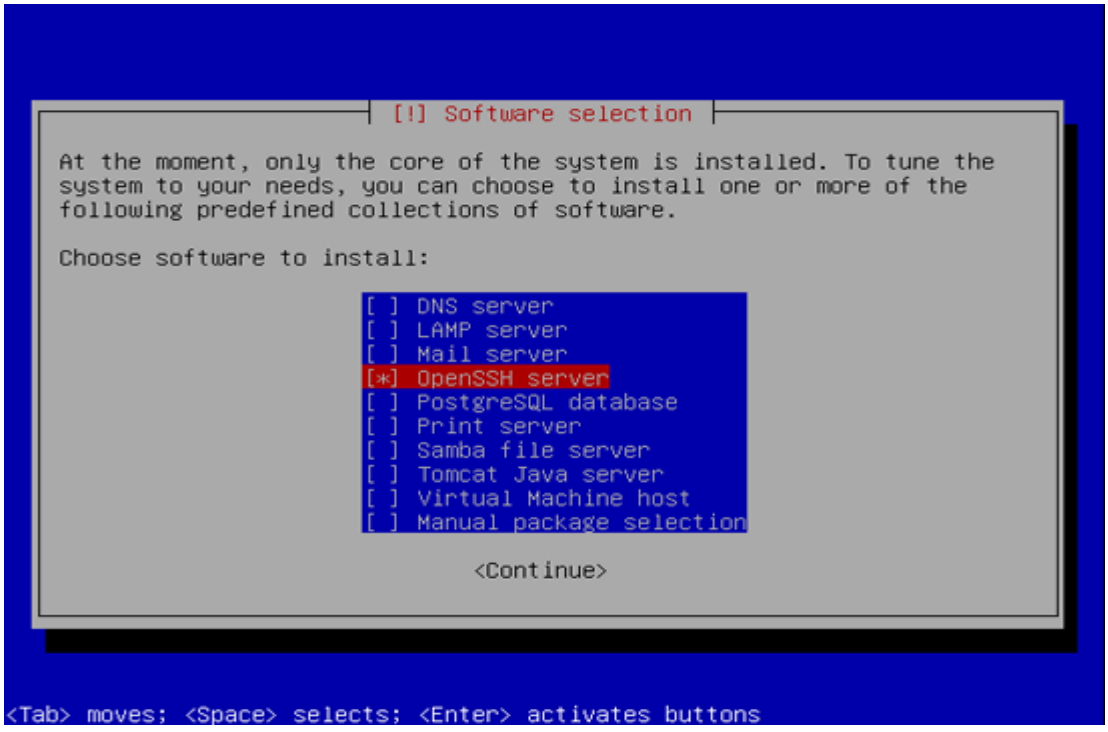


Image 8:- Software Installation Selection

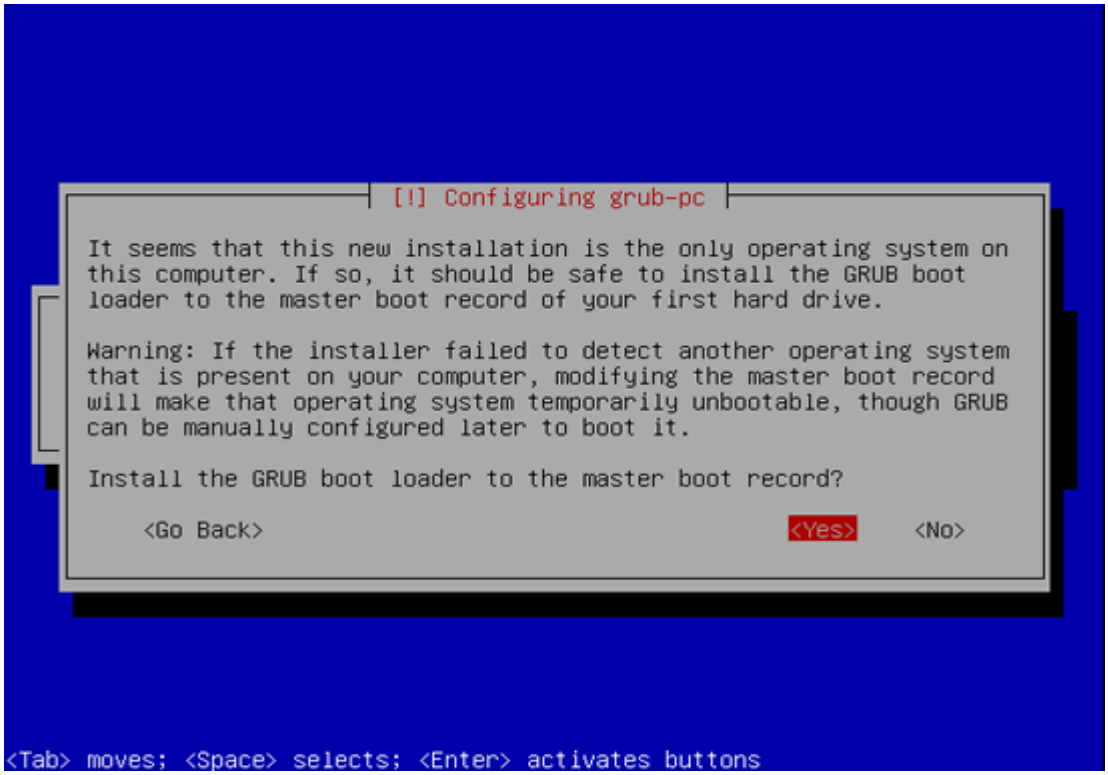


Image 9:- Grub Boot Loader Installation

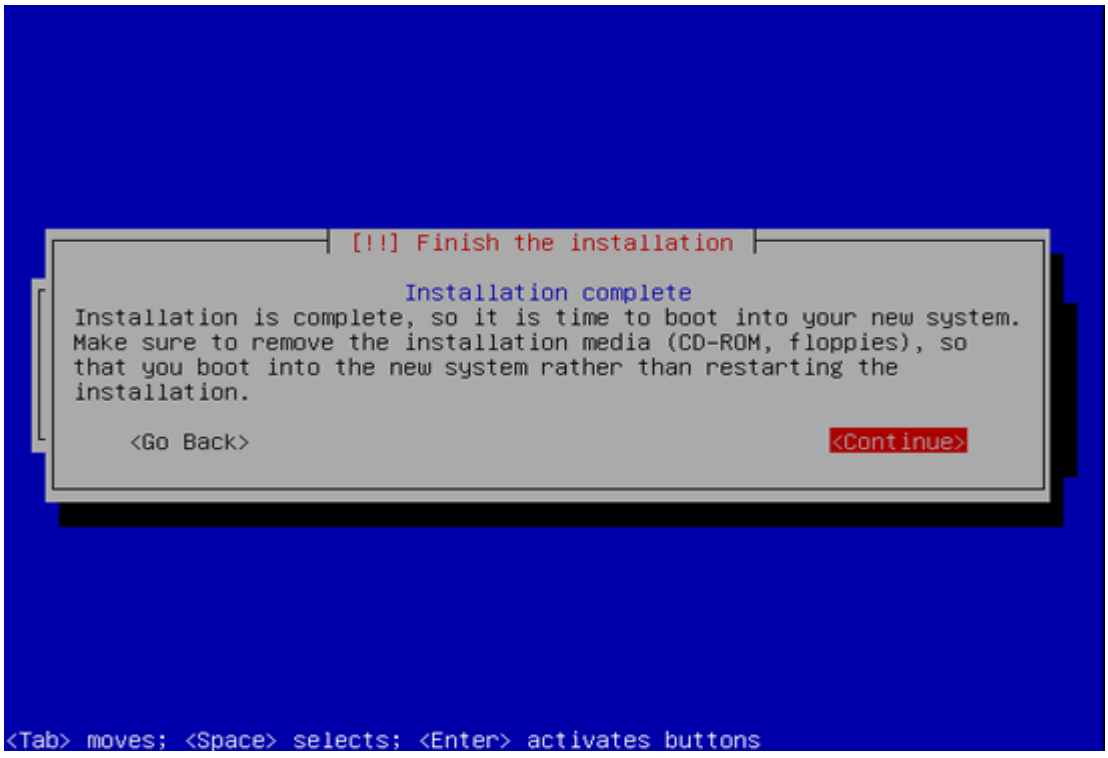


Image 10:- Installation Complete Page